



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

Facultad de Ingeniería Eléctrica

**"Diseño y Construcción del Analizador
Diferencial Digital"**

TESIS DE GRADO

**Previa a la obtención del Título de:
Ingeniero en Electricidad**

Especialización: ELECTRONICA

Presentada por:

JOHN RIGOBERTO PUNIN PALACIOS

Guayaquil - Ecuador

1990

A DEDICATORIA

Dedico esta tesis a mi madre ALEJANDRINA JURADO L. quien con mucho esfuerzo y cariño, me ha guiado en la vida impulsandome a superar y alcanzar mis mas anheladas metas, para su felicidad y la mia.

AGRADECIMIENTO

Al Ing. SERGIO FLORES M. director de esta tesis por su apoyo brindado, tanto en sus conocimientos, como en la confianza para que pueda culminar esta obra.

A la Sra. MARY RUTH DE FLORES quien con su constante apoyo moral espontaneo, facilitó en mí la entereza para superar todas las adversidades y poder así cumplir con mis objetivos.

A la Sra MERCEDES DE BEGUE y MOVIMIENTO JUVENIL ECUATORIANO por que me brindaron su generoso albergue en los momentos duros de mi vida.

A la familia BAYANCELA GUERRERO por su cariño familiar brindado todos estos años, haciendome sentir seguro y confiado.



ING. JORGE FLORES M.

Presidente del Tribunal



ING. SERGIO FLORES M.

Director de Tesis



ING. JUAN C. AVILES

Miembro Principal



ING. HUGO VILLAVICENCIO V.

Miembro Principal

DECLARACION EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, me corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL".

(Reglamento de Exámenes y Títulos profesionales de la ESPOL).

JOHN RIGOBERTO PUNIN PALACIOS

RESUMEN

El Analizador Diferencial Digital es una máquina que se utiliza en simulación de procesos. Su elemento principal es el Integrador Digital con facilidad de interconexión. El sistema de interconexión habilita incrementos a ser transferidos entre integradores. Con las ecuaciones diferenciales de las funciones de transferencia del Sistema, dibujamos el esquema de integradores interconectados que nos ayudan a resolver el problema de la misma manera como en la computadora Analógica.

El número de integradores que se utilizan, las características de cada integrador y las conexiones entre ellos es la información que se detalla a la máquina DDA por medio de un teclado con formato de entrada. La máquina DDA procederá a la conexión de los integradores digitales y a emitir los datos numéricos de las variables de salida que necesitamos conocer, dependiendo estas del tiempo o alguna otra variable independiente.

INDICE GENERAL

No	Pag.
RESUMEN	VI
INDICE GENERAL	VII
INDICE DE FIGURAS	XI
INTRODUCCION	XVI
 CAPITULO I	
DESCRIPCION TEORICA DEL ANALIZADOR DIFERENCIAL DIGITAL	
1.1 DIFERENCIA ENTRE COMPUTADOR ANALOGICA Y DIGITAL	19
1.2 EL INTEGRADOR DIGITAL (BASES MATEMATICAS Y ESTRUCTURA)	20
1.2.1 Integración numérica	20
1.2.2 Analizador diferencial digital (DDA)	22
1.3 LA ORGANIZACION GENERAL DEL DDA	26
1.3.1 El DDA Serie	27
1.3.2 El DDA Paralelo	28
1.4 PASOS GENERALES PARA LA PROGRAMACION	29
1.4.1 Estructura del Integrador	29
1.4.2 Diagrama para implementar una ecuación diferencial	32
1.4.3 Escalamiento de los Integradores Digitales	35
1.4.4 Sencillos subprogramas del DDA	43

Nº	Pag.
1.4.4.a El Lazo exponencial	44
1.4.4.b El lazo seno - coseno	45
1.4.4.c El Producto	46
1.4.4.d Generación de x^2	47
1.4.4.e La función recíproca	48
1.4.4.f La raíz cuadrada	50
1.5 ANALISIS DE ERROR	52
1.5.1 Error por Truncación	52
1.5.2 Error de Redondeo	53
1.5.3 Reducción de Error	54
1.5.4 Análisis Matemático del Error para la integración simple	56

CAPITULO II

DISEÑO DEL SISTEMA

2.1 DIAGRAMA DE BLOQUES GENERAL DEL SISTEMA DDA	63
2.2 DESCRIPCION Y FUNCIONAMIENTO DE CADA UNO DE LOS BLOQUES	64
2.3 SEUDOCODIGO DEL PROGRAMA EN EL CPU PARA EL MANEJO DE LOS DATOS DE ENTRADA, SALIDA Y COMUNICACION DE LA MAQUINA DDA	66
2.4 SEUDOCODIGO DEL PROGRAMA EN ASSEMBLER DEL TRS-80 COCO II PARA EL CONTROL DE LOS INTEGRADORES DIGITALES Y COMUNICACION DE LA MAQUINA DDA	81

CAPITULO III

CONEXION Y PRUEBAS

3.1 DIAGRAMA CIRCUITAL DE LA CONEXION PARA LA COMUNICACION DE DATOS ENTRE EL CPU Y EL TRS-80 .	93
3.1.1 Consideraciones en la comunicación serial en el PC	96
3.1.2 Consideraciones en la comunicación serial en el COCOII	101
3.2 MAPEO DE LA MEMORIA DEL TRS-80 PARA LA UBICACION DE LOS REGISTROS QUE SE COMPONE EL INTEGRADOR DIGITAL	104
3.3 PRUEBAS REALIZADAS	106
3.3.1 Pruebas para verificar el manejo de datos en el CPU	106
3.3.2 Pruebas para verificar la transmisión y recepción de datos	119
3.3.3 Pruebas para verificar el control de los integradores digitales en el TRS-80	121

CAPITULO IV

ANALISIS DE EJEMPLOS DE APLICACION DEL ANALIZADOR DIFERENCIAL DIGITAL

4.1 FILTROS DIGITALES	125
4.2 TECNICAS SOBRE EL DISEÑO DE FILTROS DIGITALES ..	125
4.2.1 Filtros	126

Nº	X Pag.
4.2.2 El filtro pasa bajo en general	128
4.2.3 Filtro pasabajo Butterworth	128
4.2.4 Función de transferencia del Butterworth	129
4.2.5 Filtro pasabajo Chebyshev	130
4.2.6 Función de transferencia del filtro Chebyshev	132
4.3 EJEMPLOS SOBRE EL DISEÑO DE FILTROS DIGITALES ..	134
4.3.1 Ejemplo 1	134
4.3.2 Ejemplo 2	148
4.4 SIMULACION DE RECURSOS DEL MEDIO AMBIENTE Y EL USO DEL LAZO ECONOMICO (ECONUSE)	166
4.4.1 Modelo ECONUSE	167
CONCLUSIONES Y RECOMENDACIONES	195
APENDICES	
APENDICE A	
DISEÑO DE PANTALLAS Y LISTADOS	198
APENDICE B	
MANUAL DEL USUARIO	211
APENDICE C	
LISTADO DE LOS PROGRAMAS	226
BIBLIOGRAFIA	364

INTRODUCCION

En las ciencias exactas, un gran número de problemas físicos son expresados en forma matemática como sistemas de ecuaciones diferenciales. Generalmente, estas ecuaciones se formulan debido a que las variables que se estudian son definidas en término de sus derivadas. Resolver tales ecuaciones no siempre es posible.

Existen soluciones analíticas, aplicando métodos matemáticos, solamente a ciertas clases de ecuaciones encuadradas en sucesos limitados.

La gran mayoría de ecuaciones diferenciales todavía son intratables y no tienen soluciones analíticas explícitas, por lo tanto, los ingenieros emplean evaluaciones numéricas para las funciones y sus integrales encontradas en tales ecuaciones. Los métodos numéricos son atractivos debido a que los computadores ejecutan dicha labor por medio de cálculos repetitivos. Los métodos numéricos para resolver ecuaciones diferenciales son conocidos desde hace algún tiempo y existen numerosos algoritmos que son utilizados para tal propósito. Usualmente, las soluciones son generadas a partir de una condición inicial y se van actualizando los valores, este procedimiento es repetido en intervalos hasta que la

función es construida. La precisión depende del algoritmo utilizado y del tiempo que emplee el computador en ejecutarlo.

La forma alternativa para ejecutar la integración numérica en un computador, es usar el Analizador Diferencial Digital (Digital Differential Analyzer, DDA). Este tipo de máquina es similar a un computador analógico en su configuración pero tiene representación digital de variables, permitiendo alta exactitud y precisión controlables.

Los bloques individuales de un Analizador Diferencial Digital pueden ser procesados secuencial o simultáneamente dando origen a dos modos de operación y consecuentemente dos velocidades de procesos. El proceso del DDA está fundamentado en hardware para resolver ecuaciones diferenciales, y es opuesto al método de software utilizados por un computador. Esta diferencia se refleja en los tiempos totales de ejecución de los dos tipos de máquinas. El DDA puede resolver ecuaciones mucho más rápido y su precisión es determinada por la longitud de sus registros.

El propósito de esta tesis es presentar el diseño y la organización de un sistema digital basado en el concepto de integración incremental DDA y un computador de propósito general que almacena las conecciones y

resultados. Se requiere de software para una óptima interface del DDA con el computador, aquí se describe detalladamente la forma de implementar el DDA y los algoritmos de los programas del computador.

Para ingresar los datos (interconexiones de integradores) y analizar a las funciones de resultado empleamos una microcomputadora IBM-XT. El Hardware para la implementación de los integradores digitales consiste de una tarjeta centrada en el microprocesador MC6809 que es de 16 bits, aquí se ejecuta el proceso del DDA. La comunicación entre el CPU y el microprocesador la hacemos mediante la interface RS232-C.

CAPITULO I

DESCRIPCION TEORICA DEL ANALIZADOR DIFERENCIAL DIGITAL

1.1 DIFERENCIA ENTRE COMPUTADORA ANALOGICA Y DIGITAL.

El computador digital trabaja con datos numéricos, que representa los valores de las variables en el tiempo y periodicamente produce salidas numéricas discretas; en cambio un computador analógico continuamente mide variables físicas como voltaje y corriente y es programado cableando en un panel grabado con varios símbolos (integradores, sumadores, etc), las salidas son instantáneas y continuas.

Aunque características del computador digital son precisión y flexibilidad, es imposible producir una salida continua la cual muchos sistemas de tiempo real demandan para su operación ideal.

La computadora analógica es capaz de producir salidas continuas pero tiene varias desventajas, probablemente la más seria es el hecho de que la variable independiente debe ser siempre el tiempo.

Comparadas con los computadores digitales, cuya precisión depende de la longitud de los registros, la

precisión de la computadora analógica es una función del número de amplificadores operacionales (directamente proporcional a la raíz cuadrada del número de amplificadores) e inherentemente una función de que los componentes sean precisos y estables, además depende de la calibración regular de las fuentes de voltaje o corriente que sirven de referencias.

1.2 EL INTEGRADOR DIGITAL (BASES MATEMATICAS Y ESTRUCTURA)

1.2.1.- INTEGRACION NUMERICA¹

La integración numérica de funciones juega un rol significativo en el desarrollo de algoritmos para la implementación en computadores. Muchas de las fórmulas han sido derivadas de aproximaciones e interpolaciones de la función original. El integral de una función continua puede ser representado como una suma de valores cuantizados tomados en intervalos espaciados, que son usualmente equidistantes. Considerando una función $y=f(x)$, su integral sobre el rango de X_0 a X_1 , puede ser denotado como

$$Z(x) = \int_{x_0}^{x_1} f(x) dx$$

$$= \lim_{|\Delta x_i| \rightarrow 0} \sum_{i=0}^n y_i \Delta x_i$$

o en forma diferencial :

$$dz = y dx$$

de esta relación :

$$Z = Z_0 + \int_{Z_0}^Z dz$$

Si todos los incrementos ΔX_i son iguales a ΔX , esta relación exacta, matemáticamente puede ser aproximada por la forma incremental:

$$\Delta Z_i = y_i \Delta x$$

y

$$Z_n = Z_0 + \sum_{i=0}^n y_i \Delta x$$

la cual es la versión discreta de la integración de EULER (rectangular). Los valores Y_i constituyen el integrando y Z_n es la integral.

Una mejor aproximación a las funciones continuas pueden ser obtenidas por integración trapezoidal en términos de cantidades previamente definidas. Las

fórmulas correspondientes son:

$$\Delta z_1 = \left[y_1 + \frac{\Delta y_1}{2} \right] \Delta x$$

y

$$z_n = z_0 + \Delta x \sum_{i=0}^n \left[y_i + \frac{\Delta y_i}{2} \right]$$

las cuales son ejemplos de integración interpolativa.

Basados en este principio, el concepto de computación incremental puede ser descrito como la técnica numérica por medio de la cual solamente cambios incrementales en las variables de una función son involucrados en el proceso de computación.

El Analizador Diferencial Digital es quizás el ejemplo mas conocido de una computador incremental.

1.2.2.- ANALIZADOR DIFERENCIAL DIGITAL (DDA)²

Un DDA consiste de un número de elementos digitales interconectados para modelar una ecuación diferencial particular. Su componente básico es un integrador, el cual, en su forma mas simple, está compuesto de dos registros y un sumador como se muestra en la figura:

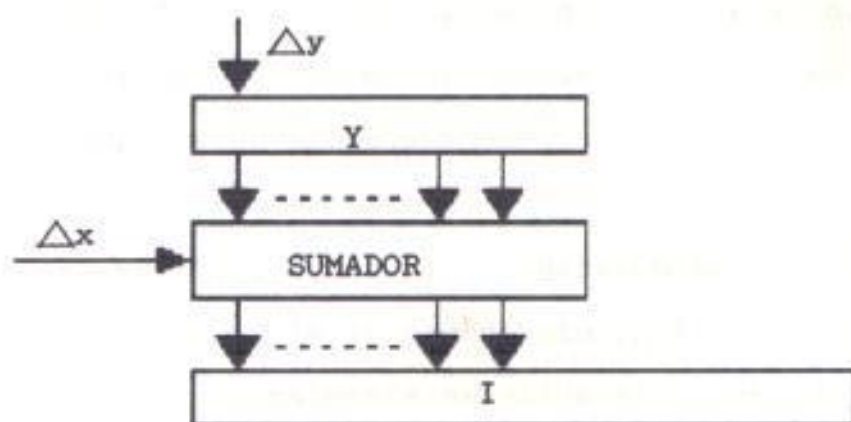


FIG. 1.1 INTEGRADOR DIGITAL

Referente a la ecuación (1) el integral Z_n está formado por repetidas sumas de $Y_i \Delta X$ al valor inicial Z_e . Como es imposible en cualquier máquina digital tener una cantidad ilimitada de posiciones digitales para representar números, Z_n puede ser acumulado solamente en un número finito de bits de un registro; esto se realiza en el Registro I. Similarmente un Registro Y mantiene el valor de la variable a ser integrada y tiene un puerto de entrada para actualizar su valor. El Registro I es lo suficientemente grande para mantener un resultado de doble longitud.

Para convertir este simple integrador en un dispositivo incremental el Registro I se corta en dos partes, los bits menos

significativos son retenidos en este nuevo Registro R, mientras la otra mitad es temporalmente descartada.

En esta operación de integración bajo control de la variable independiente ΔX (a la cual le es normalmente asignada el valor de la unidad), el corriente valor de Y_i es añadido algebraicamente (o sustraído si ΔX es negativo al contenido del Registro R). Debido a que R tiene una longitud finita igual al Registro Y ocasiona sobrecargas periódicamente. Estas sobrecargas son dirigidas a la mitad del Registro I que fue truncado que ahora son considerados como los incrementos ΔZ del integral y pueden ser enrutados a otros registros Y según sea la configuración de cada problema.



FIG. 1.2 REPRESENTACION DE LA SOBRECARGA EN EL INTEGRADOR DIGITAL.

En términos de este residuo, una ecuación diferencial puede ser escrita :

- Para el proceso de integración rectangular:

$$Y_n = Y_{n-1} + \Delta Y_n$$

$$R_n = R_{n-1} + \Delta X Y_n$$

- Para integración trapezoidal

$$R_n = R_{n-1} + \Delta X \left[Y_n + \frac{\Delta Y_n}{2} \right]$$

El resultado de esta operación ocasiona que el número R tenga sobrecargas periódicamente. Las sobrecargas representan los incrementos del integral como muestra la ecuación:

$$R_{n-1} + Y_n \cdot \Delta X = R_n + \Delta Z$$

donde ΔZ es una notación conveniente para la sobrecarga.

Las dos entradas al integrador son incrementos, del integrando y la variable independiente respectivamente, y estas pueden ser conectadas a la salida de otros integradores para formar diagramas de flujo que serán explicados luego.

El incremento ΔX tiene una limitación, que puede tomar el valor de 1, 0, -1; mientras que los incrementos ΔY no están limitados. En el integrador no está restringido el

número de entradas, las cuales, se conectan al registro Y; solamente una puede ser conectada a la entrada ΔX .

Ambos métodos de integración tienen retardos que se deben a su construcción, tanto en las salidas como en las entradas. Esto se debe a que R_n y Y_n no pueden ser computados simultáneamente.

Las combinaciones de estas unidades pueden ser usadas para simular un sinnúmero de ecuaciones diferenciales encontradas en la práctica.

1.3 LA ORGANIZACION GENERAL DEL DDA³

Al conjunto de integradores digitales con facilidades para su interconexión y la entrada/salida se la conoce como DDA (Analizador Diferencial Digital). La interconexión del sistema transfiere incrementos entre los integradores y el diagrama de los integradores resuelve ecuaciones diferenciales de la misma forma que la computadora analógica.

Existen dos maneras de organizar los DDA, una en forma simultánea o paralela y otra en forma secuencial o en serie.

1.3.1 EL DDA SERIE

En el DDA serie, el tiempo total de ejecución es dividido en periodos de interacción; en cada paso o periodo de interacción, los números de los registros Y y R de cada integrador son procesados en una Unidad Aritmética Central.

Si existen mas de una salida del integrador conectadas a una entrada Y, estos incrementos deben ser sumados juntos para formar el incremento total Y.

La organización del DDA secuencial es mostrada en esta figura:

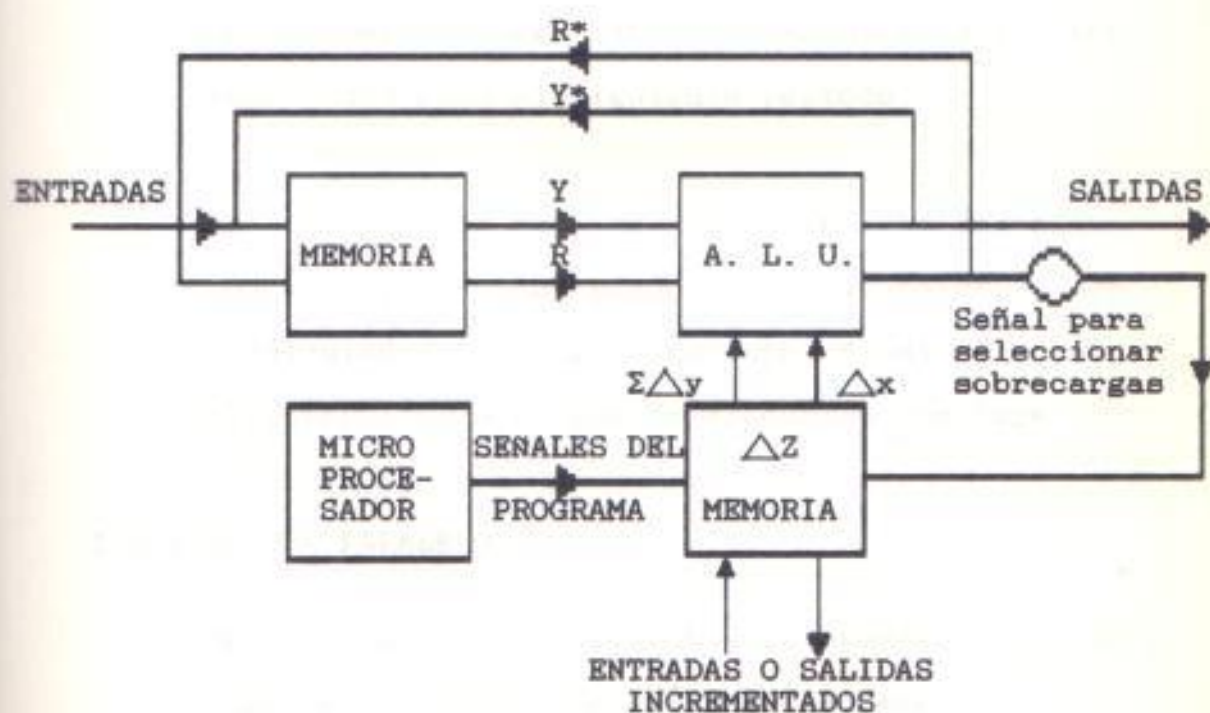


FIG. 1.3 DIAGRAMA DE BLOQUES DEL DDA SERIE

Como se observa el DDA serie consiste de cuatro unidades: el lugar de almacenamiento de los números de los registros R y Y, la unidad aritmética, el lugar de almacenamiento de los incrementos ΔX y la unidad de control.

Durante cada periodo de integración los números Y_n y R_n son leídos de la memoria y conducidos a la unidad aritmética donde se realiza el proceso. La sobrecarga resultante, que aparece al final del periodo, es seleccionado y transmitido al lugar de almacenamiento ΔZ , y los nuevos números Y_{n+1} y R_{n+1} son retornados a sus respectivos lugares de almacenamiento originales, luego los incrementos ΔY son acumulados para el siguiente período.

Este proceso se realiza una y otra vez hasta que todos los integradores digitales son recalculados. El proceso total finaliza cuando el intervalo de tiempo de ejecución concluye.

1.3.2 EL DDA PARALELO

Esta organización es mucho mas simple. En la siguiente figura se muestra su esquema:

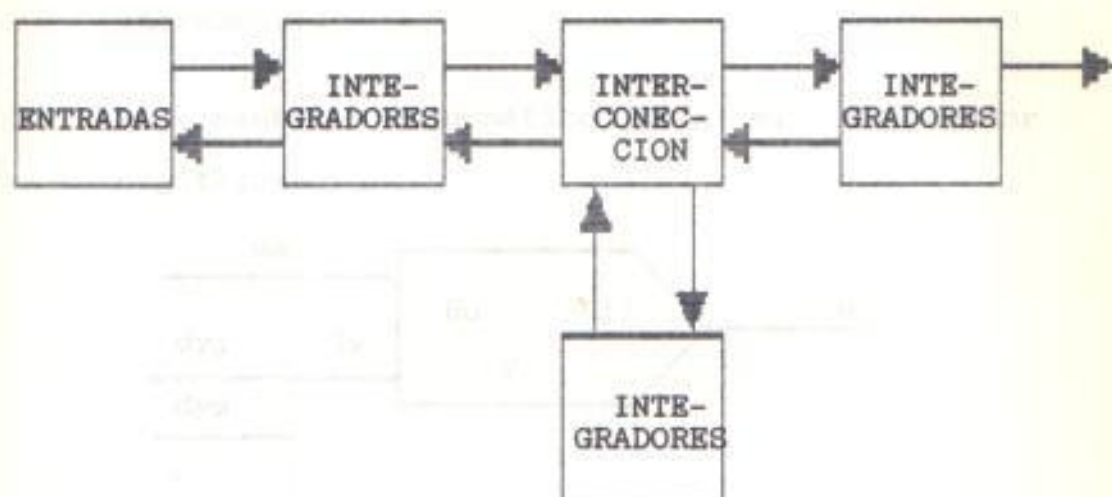


FIG. 1.4 DIAGRAMA DE BLOQUES DEL DDA PARALELO

En cada interacción todos los integradores son ejecutados al mismo tiempo. No hay necesidad de dividir en módulos. Las entradas y las salidas son conectadas directamente a los integradores y sus interconexiones se las hace cableando entre ellos por medio de un panel.

Ahora, el tiempo de ejecución total es igual al tiempo de ejecución de un integrador mas el tiempo que emplean las sobrecargas en transmitirse a los otros integradores.

1.4 PASOS GENERALES PARA LA PROGRAMACION

1.4.1 ESTRUCTURA DEL INTEGRADOR⁴

El Analizador Diferencial Digital es un dispositivo que implementa la integración

incremental.

El símbolo esquemático para el integrador digital es:

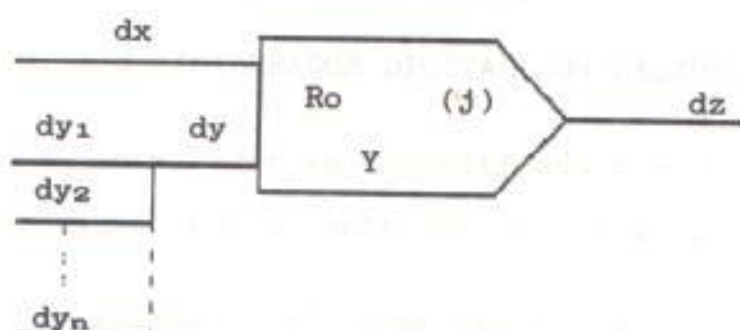


FIG. 1.5 SIMBOLO DEL INTEGRADOR DIGITAL

Las entradas al DDA son: la variable independiente y la variable dependiente, dx y dy respectivamente.

Los valores de dx son restringidos a 1, 0, -1. La suma de los incrementos dy es almacenada en el registro Y y Y es añadido al contenido del registro R si dx es positivo y substraído si es negativo.

En el momento que el registro R se excede de su máximo valor permitiendo una sobrecarga ocurre un incremento de salida. Un pequeño círculo a la salida significa que el signo de la salida es cambiado de $+dz$ a $-dz$ a todas las líneas que origina este elemento.

incremental.

El símbolo esquemático para el integrador digital es:

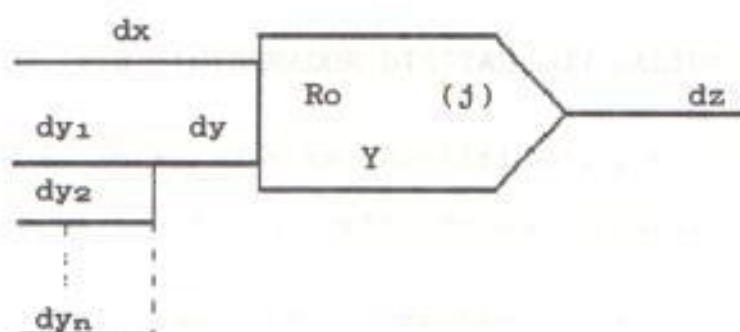


FIG. 1.5 SIMBOLO DEL INTEGRADOR DIGITAL

Las entradas al DDA son: la variable independiente y la variable dependiente, dx y dy respectivamente.

Los valores de dx son restringidos a 1, 0, -1. La suma de los incrementos dy es almacenada en el registro Y y Y es añadido al contenido del registro R si dx es positivo y substraído si es negativo.

En el momento que el registro R se excede de su máximo valor permitiendo una sobrecarga ocurre un incremento de salida. Un pequeño círculo a la salida significa que el signo de la salida es cambiado de $+dz$ a $-dz$ a todas las líneas que origina este elemento.



FIG. 1.6 INTEGRADOR DIGITAL CON SALIDA NEGATIVA

Cada integrador es identificado por un número que se encuentra dentro de un círculo.

Si utilizamos el complemento a 2 y si la longitud del registro R es L , el máximo número positivo permisible es $2^{L-1}-1$ y el mínimo número negativo es -2^{L-1} , de tal manera que la secuencia de números es circular, así:

$$0, 1, 2, \dots, 2^{L-1}-1, -2^{L-1}, -2^{L-1}+1, \dots, -2, -1,$$

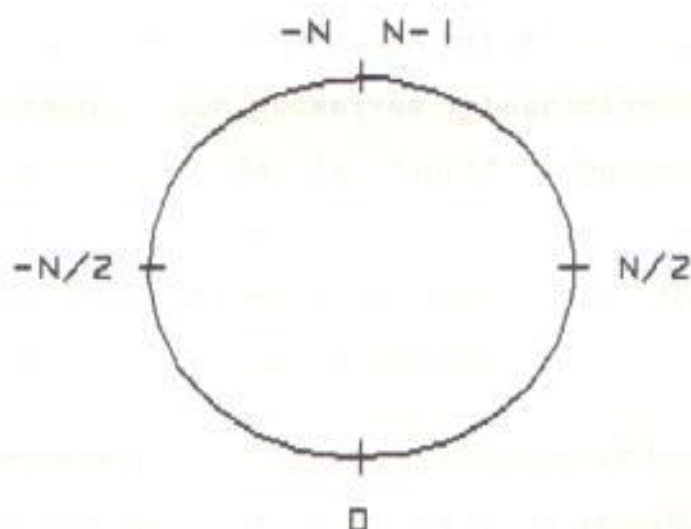


FIG. 1.7 REPRESENTACION CIRCULAR DE NUMEROS BINARIOS EN COMPLEMENTO A DOS.

Cuando el integrador contiene 2^{L-1} , la suma de un solo incremento al registro Y produce una

sobrecarga.

1.4.2 DIAGRAMAS PARA IMPLEMENTAR UNA ECUACION DIFERENCIAL.

Un solo integrador resuelve la ecuación diferencial $dz=y dx$

Para resolver ecuaciones mas complejas, varios integradores deben ser interconectados tal que el circuito resultante modela la ecuación .

Las técnicas utilizadas son similares a las utilizadas para la computadora analógica.

La ecuación debe ser reescrita en forma diferencial con la derivada de mas alto orden al lado izquierdo y todos los otros términos al lado derecho, con sucesivas integraciones todos las derivadas de la función pueden ser encontradas. La suma de estos términos es igual a la derivada de mas alto orden de la ecuación diferencial y el lazo es cerrado.

Consideremos una ecuación de segundo orden como la ecuación para los osciladores de VAN-DER-POL

$$\frac{d^2 v}{dt^2} - \epsilon(1-v^2) \frac{dv}{dt} + v = 0$$

Con las condiciones iniciales $v(0)=1.5$ y $dv/dt(0) = 0$ 33

El máximo valor de v y dv/dt será menor que 2 si ϵ está cercano a la unidad.

Reescribiendo esta ecuación con la derivada de mas alto orden en un lado y el resto de sus términos en el otro, el resultado es :

$$\frac{d^2 v}{dt^2} = \epsilon(1-v^2) \frac{dv}{dt} - v$$

o

$$\frac{d}{dt} \left[\frac{dv}{dt} \right] = \epsilon(1-v^2) \frac{dv}{dt} - v$$

Multiplicando esta ecuación por dt y desarrollando:

$$d \left[\frac{dv}{dt} \right] = \epsilon dv - \epsilon v^2 dv - v dt$$

Si existen mas variables y mas de una ecuación, este procedimiento se repite para cada ecuación y se realizan los enlaces respectivos entre ellas.

El diagrama que muestra los integradores usado para obtener una solución y sus interconexiones, está dibujado en la figura siguiente.

Las cantidades contenidas en los registros Y y las entradas y salidas de los respectivos integradores, son todas escritas en el diagrama. Las flechas indican la dirección del flujo de información.

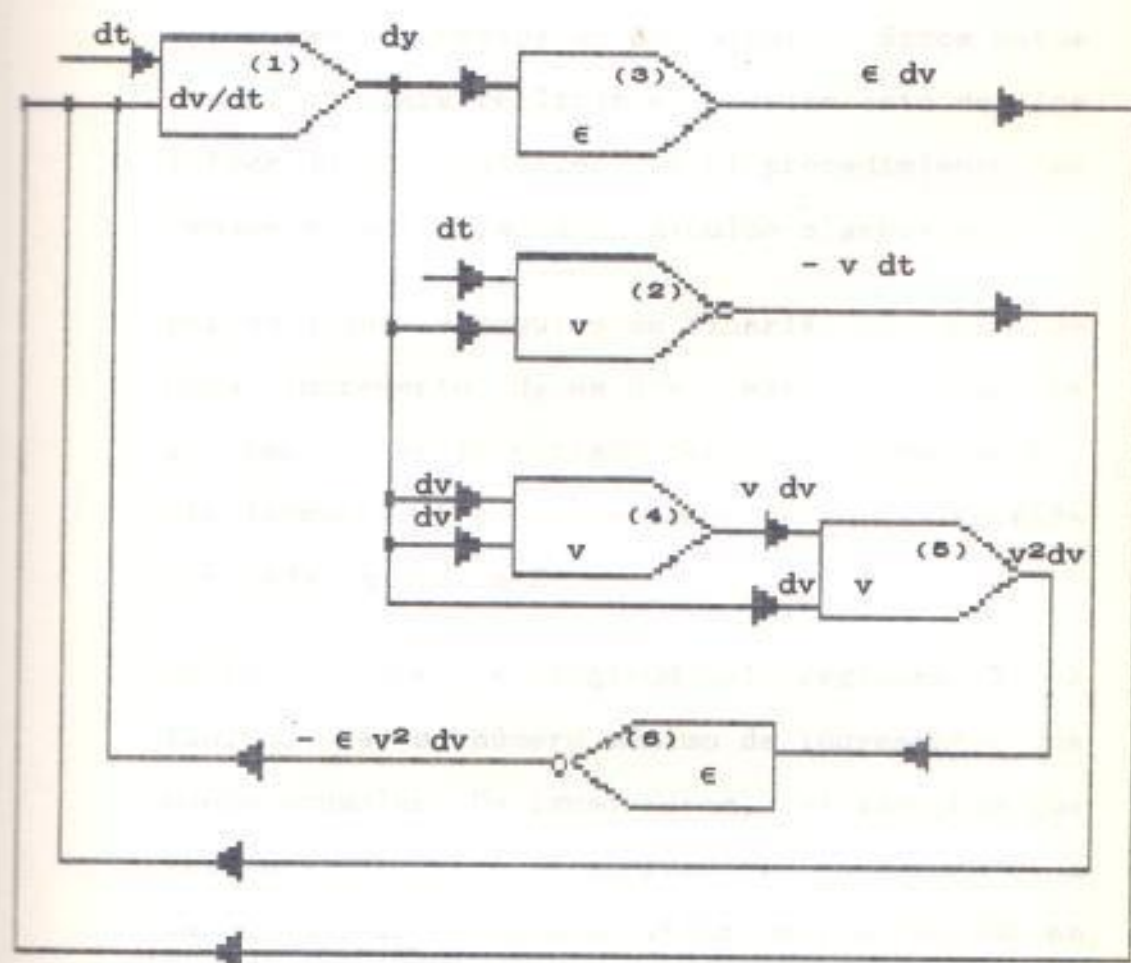


FIG. 1.8 DIAGRAMA DE LA SIMULACION DE LA ECUACION DE VAN DER POL

En los integradores 3 y 6 las constantes ϵ es mantenida, por lo cual no tiene entrada dy . El pequeño círculo a la salida dz indica que el signo se invierte.

1.4.3 ESCALAMIENTO DE LOS INTEGRADORES DIGITALES.⁵

En muchos problemas ya sean analógicos o digitales, al inicio el programador tiene idea del rango y precisión de las variables y la velocidad requerida en su cálculo. Estos datos nos sirven para realizar el escalamiento de los integradores digitales y el procedimiento se reduce a hacer pequeños cálculos algebraicos.

Debido a que la máquina es binaria, el tamaño de cada incremento Δy es 2^{-p} , esto es, hay 2^p incrementos en la entrada por cada unidad de Y . Similarmente hay 2^q incrementos en la entrada por cada unidad de X .

Debido a que la longitud del registro Y es finito, hay un número máximo de incrementos que puede acumular. De igual forma, si asumimos que este máximo número de incrementos corresponde a la unidad, el mas grande valor de Y permitido es 1. Nos indica esto que debemos normalizar a Y para que nunca exceda a 1. Analizamos Y en la ecuación original y determinamos su máximo valor, al menos aproximadamente. Se selecciona una potencia de 2, como n , tal que cuando $Y_{\max}$ es dividida para 2^n el cuociente es menor que uno; la precisión de Y será 2^{-n} , pues el rango

de Y tendrá 2^n partes.

Si la entrada al registro Y tiene 2^p incrementos por unidad de Y y Y ha sido normalizado por el factor 2^{-n} , puesto que dy debe ser añadido a Y, la longitud L para el integrador correcto será:

$$2L = 2^n \cdot 2^p$$

donde

$$L = n + p$$

Si $dz = y \cdot dx$, y el dx tiene una precisión de 2^{-s} y el Y tiene máximo 2^n incrementos por unidad, concluimos que la precisión del incremento dz es 2^{-s} o sea:

$$2^{-s} = 2^n \cdot 2^{-s}$$

por lo cual,

$$z = s - n$$

Si detallamos estas cantidades en el esquema de un integrador tendremos la notación de Monroe, la cual se indica en la siguiente figura:

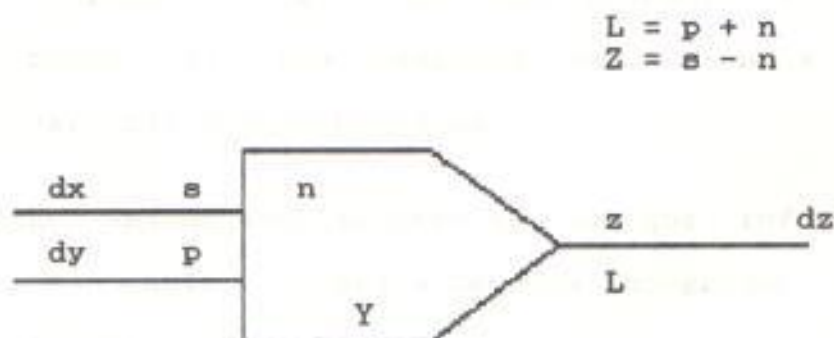


FIG. 1.9 SIMBOLO DEL INTEGRADOR DIGITAL CON PARAMETROS DE ESCALAMIENTOS.

El método de escalamiento consiste entonces:

- a) Escoger los números n tales que para cada integrador

$$2^n \geq Y_{\max}$$

- b) Por simple álgebra, calcular en función de la variable s la longitud de los registros de los integradores que es L .

El criterio será que el mas corto integrador debe ser mayor que 0 y el mas grande no puede sobrepasar la máxima longitud del registro Y . En nuestro caso que manejamos 16 bits, será 15, puesto que el bit mas significativo es el del signo.

- c) Una vez que calculamos el rango de s hacemos un chequeo para que la precisión de los incrementos dz no sea muy pequeña, y así poder dar una adecuada resolución a la variable que la contiene.

Hecho así el escalamiento tenemos que informar en una tabla el detalle de cada integrador con un formato ya establecido.

La tabla está dividida en columnas, las cuales

contienen leyendo de izquierda a derecha:

- El número del integrador (i).
- El contenido del registro Y, que es la variable a ser almacenada.
- El número del integrador cuya salida es conectada a la entrada dx, esta puede ser la variable independiente.
- Los números (hasta 5) de los integradores, cuyas salidas están conectados a la entrada dy.
- El máximo valor Y_m que puede adquirir la variable.
- El número n que representa la máxima capacidad de la variable.
- La precisión s de la entrada dx.
- La precisión z de la salida dz.
- La precisión p de la entrada dy.
- La longitud del registro del integrador L.

Número Integrador	Contenido Y	dx	dy	Y_m	n	s	z	p	L
-------------------	-------------	----	----	-------	---	---	---	---	---

Para el ejemplo de la simulación de la ecuación de VAN-DER-POL tenemos que :

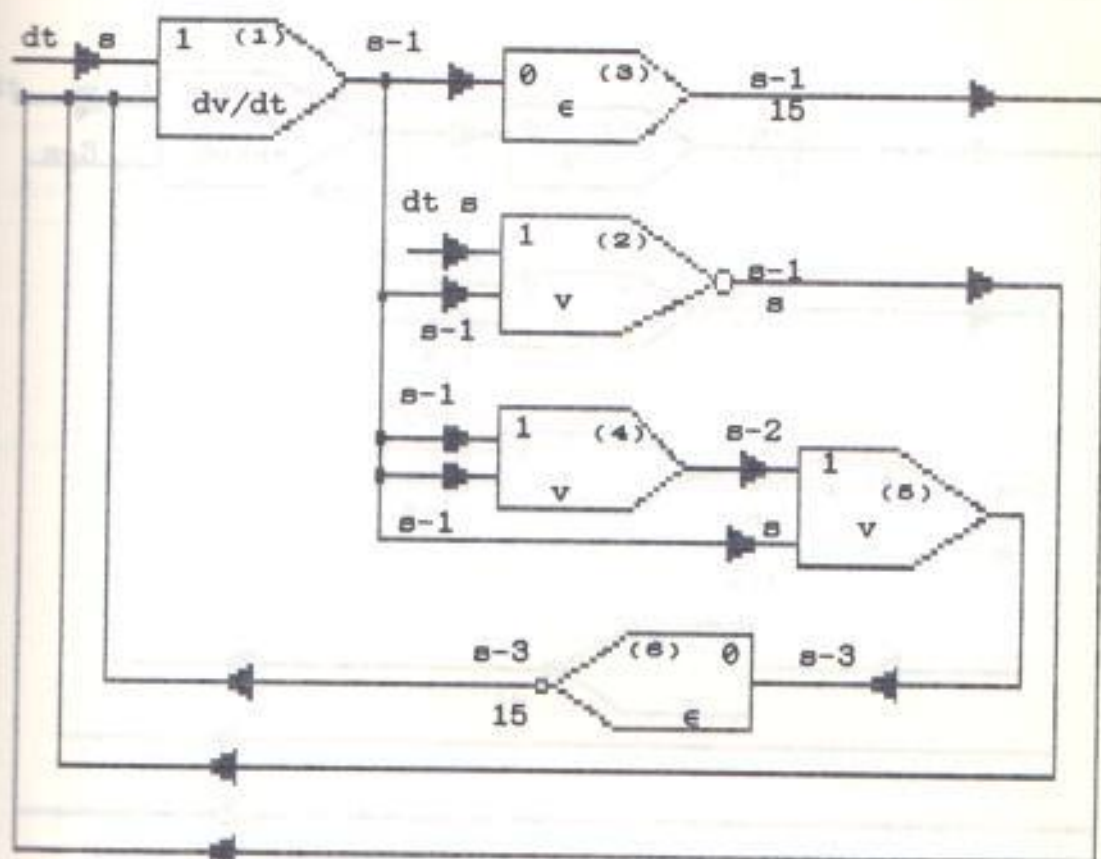


FIG. 1.10 DIAGRAMA CON ESCALAMIENTO PRIMITIVO.

El Y_m para la variable dv/dt es 2, para v es 1.5 y para ϵ es 1, con estos datos asignamos los valores de n para cada integrador.

Si observamos la entrada dy del integrador 1, vemos que difieren en valor cada uno de los ramales que llega. Escogemos entonces el mas pequeño de ellos que es $s-3$, que proviene del integrador 6, por lo que tenemos que incrementar el número n de los integradores 2 y 3, y así el nuevo diagrama es:

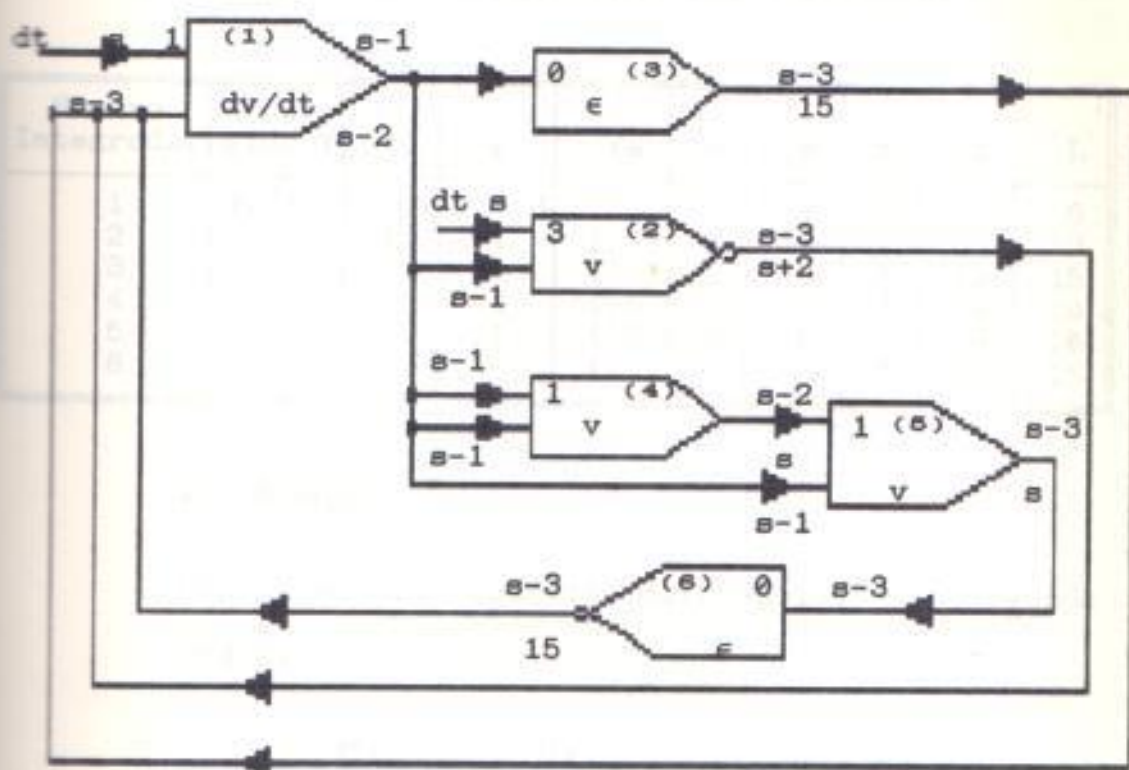


FIG. 1.11 DIAGRAMA DE LA SIMULACION DE LA ECUACION DE VAN DER POL CON ESCALAMIENTO COMPLETO.

Observamos que la longitud del integrador mas pequeño es $s-2$ y la del integrador mas grande es $s+2$, por lo que:

$$s - 2 > 0 \quad \text{y} \quad s + 2 \leq 15$$

Entonces, s está en un rango entre $2 < s \leq 13$.

Si escogemos un $s = 10$ tendremos precisiones para los dz bastantes altas.

Ahora hacemos una tabla que informe las

precisiones y longitud de los registros.

Número Integrador	Contenido Y	dx	dy	Y _m	n	s	z	p	L
1	dv/dt	dt	2,3,6	2	1	10	9	7	8
2	v	dt	1	1.5	3	10	7	9	12
3	ε	1	-	1	2	9	7	13	15
4	v	1	1	1.5	1	9	8	9	10
5	v	4	1	1.5	1	8	7	9	10
6	ε	5	-	1	0	7	7	15	15

2do Ejemplo:

Considere la ecuación diferencial de segundo orden,

$$\frac{d^2 y}{dx^2} + A \frac{dy}{dx} + B y = 0$$

Suponemos que $A = 0.271$, $B = 12.3$ y queremos la solución para $0 \leq x < 8$. El diagrama de interconexión de los integradores de:

$$\frac{d^2 y}{dx^2} = -A \frac{dy}{dx} - B y$$

$$\frac{d}{dx} \left[\frac{dy}{dx} \right] = -A \frac{dy}{dx} - B y$$

multiplicado por dx,

$$d \left[\frac{dy}{dx} \right] = -A dy - B y dx$$

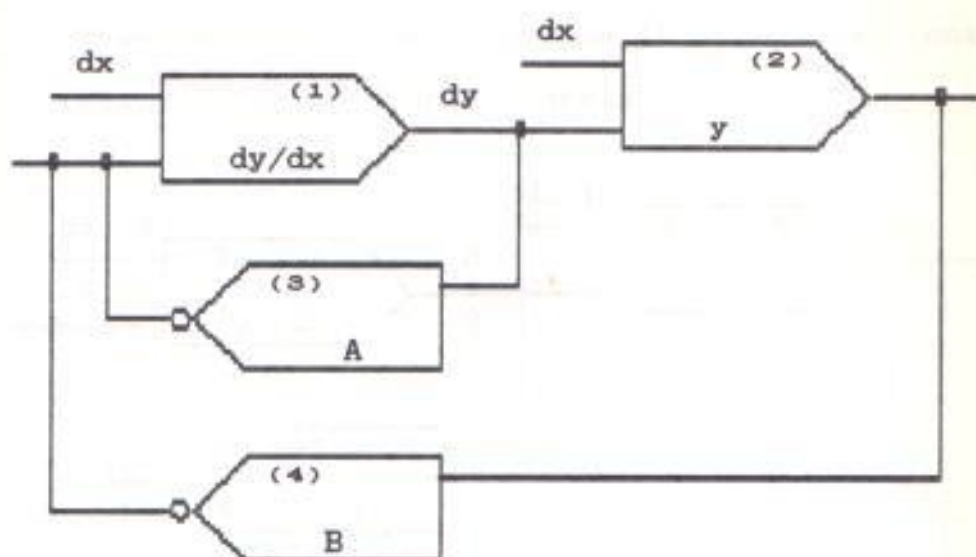


FIG. 1.12 DIAGRAMA DE LA SIMULACION DE LA ECUACION DIFERENCIAL DE SEGUNDO ORDEN.

Con los valores de $Y_m = 1$ y $dy/dx_{max} = 4$ tenemos que $n_1=0$, $n_2=2$. Debido a que $A = 0.271$, $n_3 = 0$ y $B = 12.3$, $n_4 = 4$.

Para escalar utilizamos la variable s en la entrada independiente, así tenemos que:

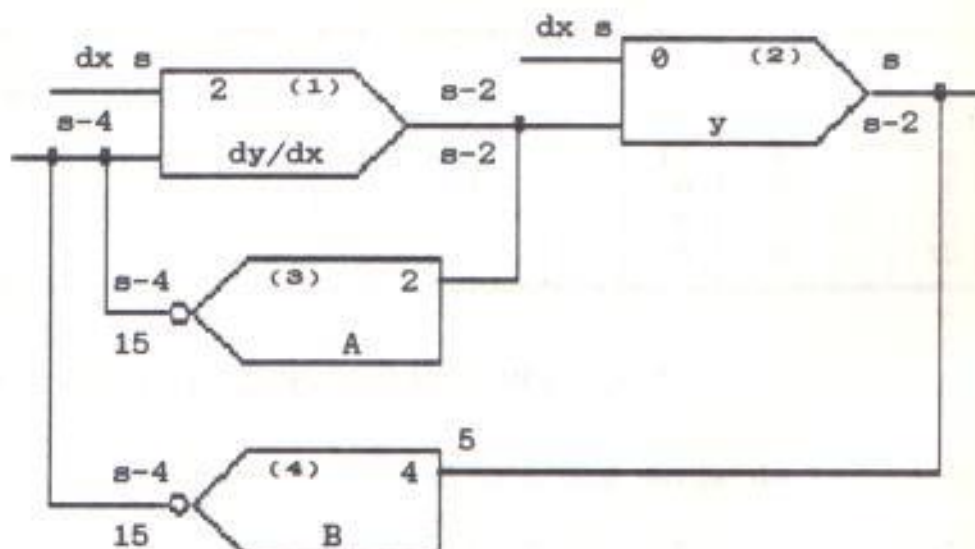


FIG. 1.13 ESCALAMIENTO DE LA ECUACION DIFERENCIAL DE SEGUNDO ORDEN.

El integrador mas pequeño será s-4 por lo que escogemos un s conveniente de 10 para tener una precisión de 2^{-10} en la escala x.

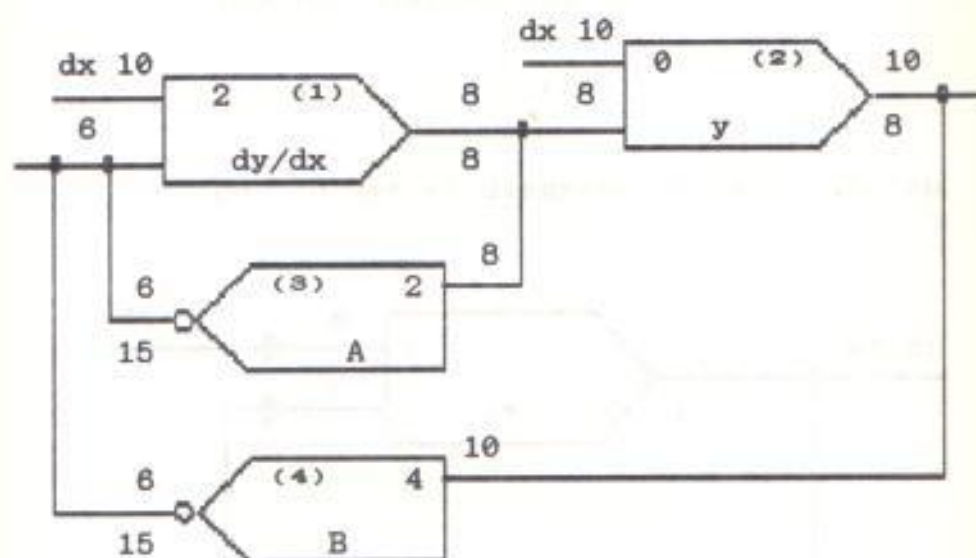


FIG. 1.14 DIAGRAMA COMPLETO DE LA SIMULACION DE LA ECUACION DIFERENCIAL DE SEGUNDO ORDEN.

La tabla del informe de las conecciones es:

Número Integrador	Contenido Y	dx	dy	Y _m	n	s	z	p	L
1	dy/dx	dx	3,4	1	2	10	8	6	8
2	y	dx	1	4	0	10	10	8	8
3	A	1	-	0.27	2	8	6	13	15
4	B	2	-	12.3	4	10	8	11	15

1.4.4 SENCILLOS SUBPROGRAMAS DEL DDA.^e

A continuación detallamos una serie de funciones que pueden utilizarse en ecuaciones diferenciales más complejas.

1.4.4.a El Lazo Exponencial.

La ecuación diferencial que define la función exponencial es :

$$d(e^t) = e^t dt$$

por lo que el diagrama de conexión es :

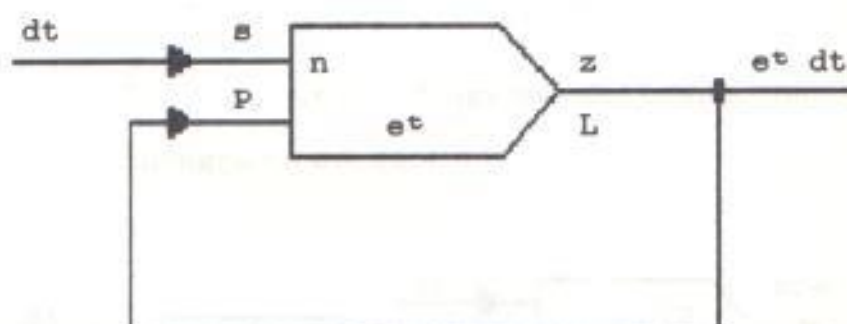


FIG. 1.15 LAZO EXPONENCIAL

Observamos que $z = p$, debido a que se encuentra conectadas entre sí, como

$L = n + p$ y $z = s - n$ reemplazando tendremos que:

$$L = n + (s - n) = s$$

para un buen escalamiento $L = s$

En el caso especial que $Y_{\max} = 1$, $n = 0$; si fijamos un valor de L cualquiera,

$$p = L$$

1.4.4.b El lazo Seno - Coseno.

Las funciones seno y coseno son generadas la una con respecto a la otra, pues las ecuaciones diferenciales que las definen son:

$$d(\text{sen } t) = \text{cos } t \, dt$$

$$d(\text{cos } t) = -\text{sen } t \, dt$$

Por lo que el diagrama de conexión de integradores es:

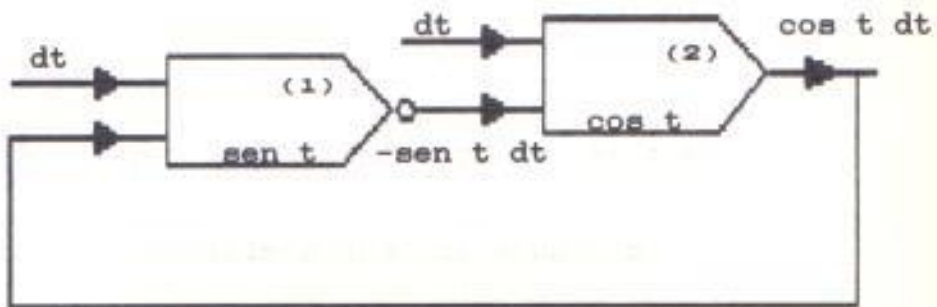


FIG. 1.16 LAZO SENO - COSENO

En el escalamiento, por observación tenemos que :

$$z_1 = p_2 \quad \text{y} \quad z_2 = p_1$$

Como es la misma escala de la variable independiente, entonces:

$$s_1 = s_2 = s$$

Si tomamos a los dos integradores con igual capacidad.

$$L_1 = L_2 = L$$

entonces planteamos las ecuaciones para cada integrador

$$z_1 = s - n_1 \quad (1) \quad z_2 = s - n_2 \quad (3)$$

$$L = p_1 + n_1 \quad (2) \quad L = p_2 + n_2 \quad (4)$$

igualamos (1) con (3) y (2) con (4).

$$z_1 + n_1 = z_2 + n_2$$

$$p_1 + n_1 = p_2 + n_2$$

$$\text{y como } z_1 = p_2 \text{ y } z_2 = p_1$$

$$\text{concluimos que } n_1 = n_2 = n$$

$$\text{por lo que si } z_1 = s - n$$

$$\text{y } p_2 = L - n$$

igualamos y observamos que:

$$s - n = L - n$$

$$s = L$$

1.4.4.c El Producto.

El producto está definido por la ecuación diferencial:

$$d(uv) = u \, dv + v \, du$$

El diagrama es :

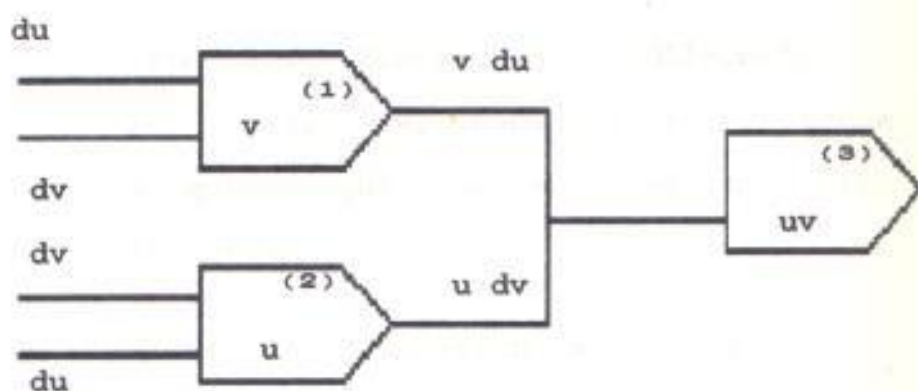


FIG. 1.17 EL PRODUCTO.

Observamos que :

$$p_3 = z_1 = z_2$$

$$\text{Si } z_1 = s_1 - n_1 \text{ y } L_1 = p_1 + n_1$$

$$z_1 = s_1 - L_1 + p_1$$

$$\text{y } z_2 = s_2 - L_2 + p_2$$

$$\text{Si escogemos que } L_1 = L_2 = L$$

$$\text{entonces , } s_1 + p_1 = s_2 + p_2$$

1.4.4.d Generación de x^2

La ecuación diferencial para este diagrama es:

$$d(x^2) = 2x \, dx$$

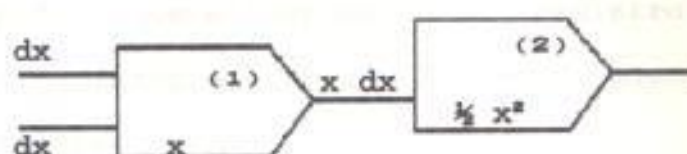


FIG. 1.18 GENERACION DE POLINOMIOS.

Por medio del escalamiento podemos asegurar que no se necesita otro integrador.

Si $2 \cdot 2 = 4 = 2 - p_2$ entonces $p_2 = 4 - 2 = 2$

De esta forma no tenemos necesidad de multiplicar por 2 para generar la función x^2 .

1.4.4.e La función recíproca.

La siguiente ecuación diferencial define a la función recíproca así:

$$\begin{aligned} d(1/x) &= - 1/x^2 \, dx \\ &= - 1/x \cdot 1/x \, dx \end{aligned}$$

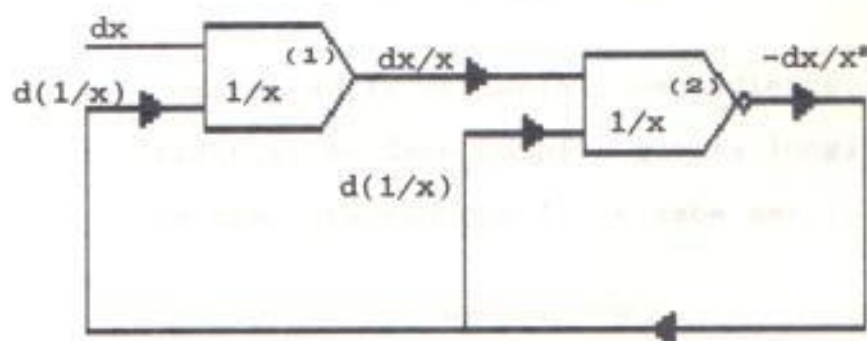


FIG. 1.19 LA FUNCION RECIPROCA.

Debido a que el contenido del registro Y de los integradores 1 y 2, es el mismo, entonces:

$$n_1 = n_2 = n \quad \text{y} \quad L_1 = L_2 = L$$

además notamos por el diagrama que:

$$z_1 = s_2$$

y,

$$z_2 = p_1 = p_2$$

entonces,

$$z_1 = s_1 - n$$

reemplazando,

$$z_2 = s_2 - n = (s_1 - n) - n = s_1 - 2n$$

Si,

$$L = p_1 + n \quad \text{y} \quad p_1 = z_4$$

entonces,

$$L - n = s_1 - 2n$$

como s_1 es la variable independiente, al fijar s_1 se debe cumplir que la longitud de los integradores 1 y 4 debe ser :

$$L = s_1 - n$$

1.4.4.f La raíz cuadrada.

El diagrama mostrado en la siguiente figura está conformado por los siguientes ecuaciones diferenciales.

$$d(x^{1/2}) = 1/2 x^{-1/2} dx$$

$$d(x^{-1/2}) = - 1/2 x^{-3/2} dx$$

$$d(x^{-1}) = - x^{-2} dx$$

Son generados a la vez $x^{-1/2}$, x^{-1} y $x^{1/2}$

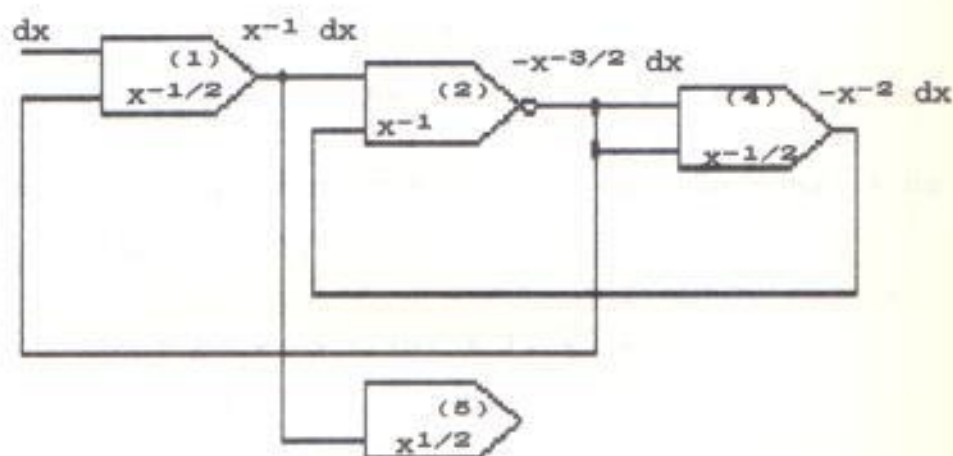


FIG. 1.20 LA RAIZ CUADRADA.

Observamos que :

$$z_1 = s_2 = p_3 = 1$$

$$z_2 = s_4 = p_4 = 1 = p_1 = 1$$

$$z_4 = p_2$$

Si consideramos que $n_1 = n_4 = n$ y si fijamos $s_1 = s$

Las ecuaciones que fijan la precisión de cada salida será:

$$z_1 = s - n = s_4$$

$$z_2 = s_2 - n_2 = (s - n) - n_2 = s_4$$

$$z_4 = s_4 - n = (s - n - n_2) - n = s - 2n - n_2$$

La longitud de cada registro se la calcula así:

$$L_1 = p_1 + n = (z_2 + 1) + n$$

$$L_1 = (s - n - n_2 + 1) + n = s - n_2 + 1 = L_4$$

$$L_2 = p_2 + n_2 = z_4 + n_2 = (s - 2n - n_2) + n_2$$

$$L_2 = s - 2n$$

$$L_3 = p_3 + n_3 = (z_1 + 1) + n_3$$

$$L_3 = (s - n + n_3 + 1)$$

1.5 ANALISIS DE ERROR⁷

Todas la cantidades digitales son discretas y nunca pueden ser representadas en variables continuas exactamente. El proceso se lo puede hacer tan preciso como se requiera, haciendo el tamaño del paso suficientemente pequeño, pero nunca será completamente preciso. Así, dos tipos de errores existen en el proceso digital de integración, por conveniencia será referido como "redondeo" y "truncación", también existen efectos de segundo orden que son causados por desfaseamiento al procesar los integradores, pero son insignificantes. El orden de procesamiento de los integradores contribuye al error porque algunos integradores reciben en la entrada Δx a la variable independiente y varios reciben las conecciones Δz de otros integradores, esto produce una diferencia en la transmisión de error de un integrador a los otros. La naturaleza de la ecuación a ser resuelta, determina el orden de procesamiento puesto que es lógico primero procesar a los integradores cuyas derivadas proveen a los otros.

1.5.1 ERROR POR TRUNCACION

Esta fuente de error (también conocida como error de discretización), es la diferencia entre el valor verdadero de una función y su

aproximación digital. Es causada por los infinitésimos de la aproximación numérica a una función y es dependiente del algoritmo utilizado. Por ejemplo una función conteniendo exponenciales y cantidades hiperbólicas no puede ser computada exactamente, porque su expansión de serie de Taylor tiene un número infinito de términos. Para integraciones rectangulares el error por paso de truncación es sobre el orden de $(\Delta x)^2/2$. Usando altas fórmulas de cuadraturas el error por truncación puede ser disminuido, pero la ventaja ganada es pérdida porque la conceptual simplicidad del procesador DDA es sacrificada por el mucho Hardware y pasos adicionales en el proceso que son requeridos para implementar los algoritmos de alto orden de integración.

1.5.2 ERROR DE REDONDEO

En la máquina DDA las magnitudes se representan en palabras de longitud fija, lo que nos lleva a cortar los números representados introduciendo un error. En un integrador individual, el error de redondeo es pequeño; pero como las ecuaciones son resueltas por sistemas de lazos cerrados,

estos errores se propagan rapidamente.

Error por truncación puede estar definido matematicamente, despues que el algoritmo de integración es especificado y en muchos casos un límite puede ser establecido. Error por redondeo es un efecto no lineal y no se lo puede especificar en muchos casos, por lo que recurrimos a muchas teorías probabilísticas. Para la solución de ecuaciones diferenciales lineales, un estimado basado en distribuciones aleatorias, calcula al error acumulativo de redondeo en aproximadamente $0.58 \times N$, donde N es el número de iteraciones.

1.5.3 REDUCCION DE ERROR

Como lo notamos anteriormente, error por truncación puede ser reducido utilizando algoritmos de integración de alto orden, pero esto es realizable solamente hasta cierto punto, ya que manejar un solo bit de incremento hace que los cálculos de diferencias de altos ordenes sea una tarea sin significado. La experiencia nos muestra que la regla de integración trapezoidal es uno de los mejores compromisos entre precisión y facilidad de implementación; teniendo un error

de truncación del orden de $\Delta x^3/4$. Otra forma de minimizar el error es aumentar la precisión usando palabras mas largas, pero la desventaja adicional es que se incrementa el tiempo de procesamiento.

En resumen, la precisión total del proceso incremental involucrado en DDA es dependiente del algoritmo seleccionado, la longitud de la palabra usada y la naturaleza de la ecuación a ser resuelta. En muchas situaciones prácticas una palabra de 16 bit de longitud causa un error que raramente excede 0.01%.

Si el error de redondeo es ignorado, el error puede ser limitado a tres principales fuentes: truncación y errores en Δx y Δy . Cada uno de ellos puede ser tratados independientemente y sus efectos superpuestos para producir el estimado total.

Desviaciones en Δx y Δy pueden ser asumidos seguramente con un límite de un incremento; como ellos pueden ser positivos o negativos se toma como promedio el valor de la mitad de un incremento. Tomando la desviación estandar de una distribución uniforme para el error en un rango de ± 1 , un esperado valor de un tercio

es obtenido y este es un número mas realístico. De esta manera el error total puede ser expresado así:

$$\text{total error} \approx 1/3(\Delta x + \Delta y + N \Delta x^{3/4})$$

El último término es el error de truncación para el algoritmo de integración trapezoidal, su contribución es pequeña y casi no existe. Para un procesador que tiene palabra con una longitud de 16 bits (con un bit de signo), un incremento tiene una magnitud de 2^{-15} unidades; un límite aproximado puede ser establecido para el error:

$$\begin{aligned} \text{total de error} &\approx 2/3 \text{ de } 1 \text{ incremento} \\ &\approx 2/3 * 2^{-15} * 100\% \\ &\approx 0.002\% \end{aligned}$$

El error de redondeo puede ser ignorado hasta aquí. En el peor de los casos, con cada contribución de redondeo el total de error llega al 0.004%, lo cual es mas bajo que el 0.01% estimado anteriormente.

1.5.4 ANALISIS MATEMATICO DEL ERROR PARA LA INTEGRACION SIMPLE

Para la integración simple la variable independiente se incrementa en iguales pasos y

usualmente esto significa que el dispositivo es integrado con respecto al tiempo. El primer error se considera, en el efecto de la diferencia entre la señal analógica a la representación digital de una función. Si llamamos y_1 al valor de la función analógica y Y_1 el valor digital cuantizado, difiere entonces en α_1 llamado el error de redondeo. Esto es representado por la ecuación

$$Y_1 = y_1 - \alpha_1.$$

Supongamos que i pasos de integración han ocurrido, entonces al inicio de $(i+1)$ ésimo paso, el dato Y_1 es insertado en el registro Y del primer integrador, Y_1 es entonces añadido al número R_1 en el registro R y lo cambia a R_{i+1} , una sobrecarga dz_1 es entonces generado el cual es añadido al integrando del segundo integrador y cambia su contenido de z_1 a z_{i+1} , así:

$$z_{i+1} = z_1 + dz_1$$

Para obtener la relación entre la sobrecarga dz y los números Y y R del primer integrador, es necesario definir la capacidad del registro R , 2^n unidades; cuando estas 2^n unidades son añadidas al registro R una sobrecarga de una

unidad es producido.

Cuando Y_1 es añadido al registro R existen 3 posibilidades:

1) sobrecarga cero, en este caso:

$$dz_1 = 0 \quad R_{i+1} = R_i + Y_i$$

2) sobrecarga positiva, en este caso:

$$dz_1 = 1 \quad R_{i+1} = R_i + Y_i - 2^n$$

3) sobrecarga negativa, en este caso:

$$dz_1 = -1 \quad R_{i+1} = R_i + Y_i + 2^n$$

Estos tres resultados pueden ser combinados en una fórmula, como:

$$dz_1 = 1/2^n * (Y_i + R_i - R_{i+1})$$

Sumando esta ecuación desde $i=0$ hasta $i=2^{L-1}-1$, donde $2^{L-1} = N$ es el número total de pasos dados y L el número de bits del registro R , tendremos:

$$z_N - z_0 = 1/2^n * (Y_0 + Y_1 + \dots + Y_{N-1}) + 1/2^n * (R_0 - R_N)$$

donde por lo general $R_0 = 0$

Esta ecuación puede ser escrita en la forma:

$$z_N - z_0 + R_N/2^n = 1/2^n * (Y_0 + Y_1 + \dots + Y_{N-1})$$

considerando a z_N y $R_N/2^n$ como la mayor y la menor cantidad significativas de las mitades del número digital. En la integración se suman sucesivamente los números $Y_0, Y_1, \dots, Y_{N-1}$ en la mitad menos significativa del número en doble longitud.

Introduciendo el efecto de redondeo en la ecuación tenemos:

$$z_N - z_0 = \frac{1}{2^n} (Y_0 + Y_1 + \dots + Y_{N-1}) + \frac{R_N}{2^n} - \frac{1}{2^n} (a_0 + a_1 + \dots + a_{N-1})$$

Si utilizamos la integración trapezoidal, una aproximación podríamos obtener. Supongamos que los pasos de integración ocurren en tiempos de $t_0, t_1, \dots, t_{N-1}$ en el cual el intervalo de tiempo entre pasos sucesivos es igual a Δt , entonces el área bajo la curva entre t_i y t_{i+1} es igual a $\frac{1}{2} \Delta t * (y_i + y_{i+1})$, así:

$$\int_{t_0}^{t_N} y \cdot dt = \frac{1}{2} \Delta t (y_0 + y_1) + \frac{1}{2} \Delta t (y_1 + y_2) + \dots + \frac{1}{2} \Delta t (y_{N-1} + y_N)$$

$$\int_{t_0}^{t_N} y \cdot dt = \frac{1}{2} \Delta t (y_N - y_0) + \Delta t (y_0 + y_1 + \dots + y_{N-1})$$

Sustituyendo en la ecuación :

$$Z_N - Z_0 = \frac{1}{2^n \Delta t} \int_{t_0}^{t_N} y \cdot \Delta t - \frac{1}{2 \cdot 2^n} (y_N - y_0) + \frac{R_N}{2^n} + \frac{1}{2^n} (a_0 + a_1 + \dots + a_N)$$

La ecuación da el valor del número en el registro Z en términos del área bajo la curva y muestra los varios errores involucrados. El primer punto es el factor de escala $(2^n \Delta t)^{-1}$, tiene el efecto de convertir a Z_N en un número entero no dimensional.

El término $(y_N - y_0)/(2 \cdot 2^n)$ se debe al muestreo de la curva en discretos instantes de tiempo y es el error de truncación mencionado anteriormente; este término no representa el error completo de truncación pero es el elemento mas significativo, es de hecho la contribución que nos lleva al $(\Delta t)^2$ término de la expansión de la serie de Taylor en la integral; su valor no excede una unidad o quantum en el registro Y .

El término $R_N/2^n$ es el primer error de redondeo y proviene del registro R del primer integrador, recordando que este efecto se produce porque uno está redondeando la mitad menos significativa del número de doble

longitud. Aunque el número R puede variar entre 0 y 2^n , el valor mas grande que puede tener es de una unidad.

El término $(a_0 + a_1 + \dots + a_{n-1})/2^n$ es debido al efecto de redondeo entre la señal analógica que se estudia y la digital que tratamos y depende del número de pasos de la integración. Este error es mas conveniente representarlo en una proporción; aunque el valor promedio de a es claramente la mitad de una unidad, este término contribuye al error total a la proporción de la mitad de una unidad en 2^n pasos de integración.

Ignorando el último efecto de redondeo, podemos ver que los efectos de error de redondeo y truncación en el proceso puramente digital de integración tienen el mismo orden de magnitud, una unidad en la mitad menos significativa de la integral. En este sentido es verdadero decir que en el proceso de integración simple en el DDA, los efectos de redondeo y truncación son balanceados entre ellos.

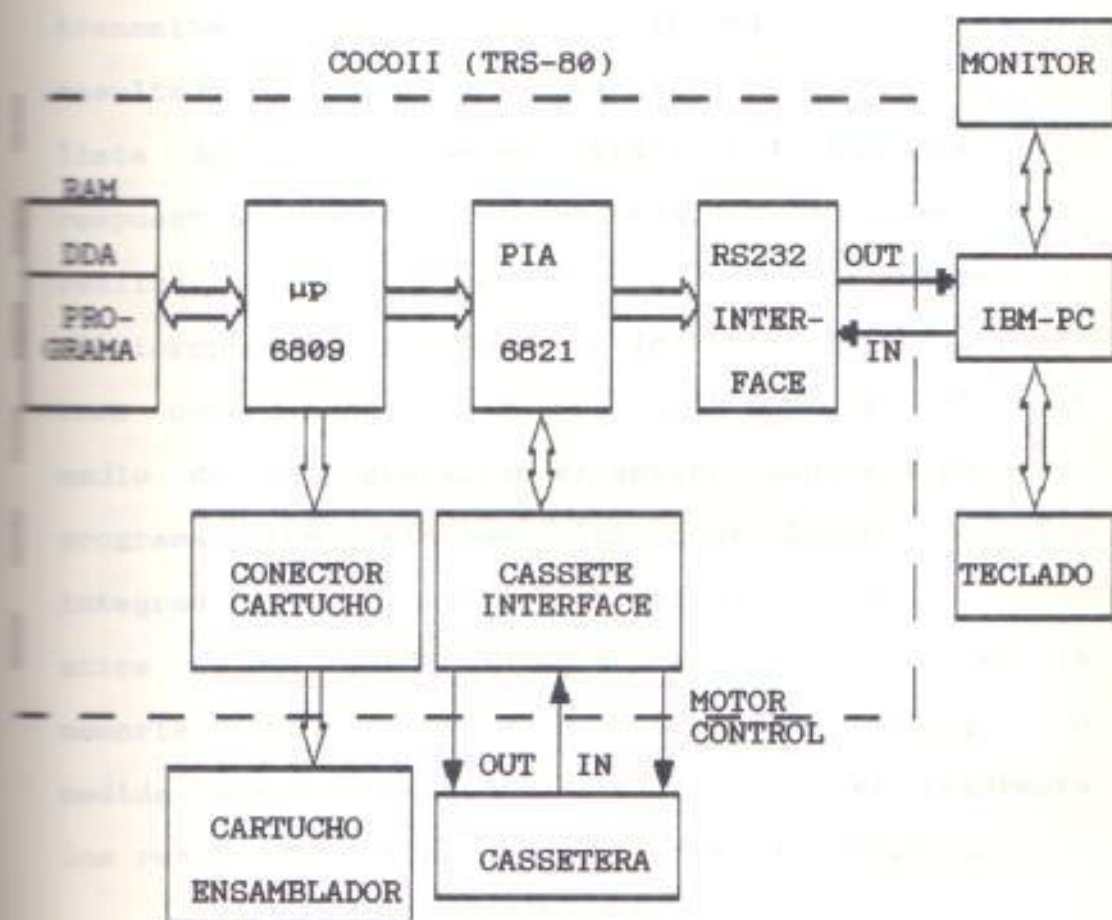
REFERENCIAS

1. PARASURAMAN B. Digital Incremental Computation using Automatic Programming Thecniques. Scientific Report No 40 National Aeronautics and Space Administration. 1972. pp 7-11.
2. SIZER, T.R.H. C.ENG. M.I.E.E. The Digital Differential Analyser. Chapman and Hall Ltd. 1968. pp 17-19.
3. SIZER, Op. Cit., pp 23-27.
4. SIZER, Op. Cit., pp 28-30.
5. MONROE, A.J. Digital Processes for Sampled Data Systems. 1971 . pp 40-43
6. SIZER, Op. Cit., pp 35-40.
7. PARASURAMAN, Op. Cit., pp 12-15.
8. SIZER, Op. Cit., pp 59-63.

CAPITULO II

DISEÑO DEL SISTEMA

2.1 DIAGRAMA DE BLOQUES GENERAL DEL SISTEMA DDA.



2.2 DESCRIPCION Y FUNCIONAMIENTO DE CADA UNO DE LOS BLOQUES.

La máquina DDA se la implementó utilizando el microcomputador IBM-PC y el microprocesador 6809 comunicados con la interface RS232-C. En el primero de ellos se ejecuta el programa en lenguaje de alto nivel que maneja los datos de entrada: interconexión de integradores, escalamiento de cada uno; luego, transmite estos datos, recibiendo los datos de resultado que pueden ser analizados ya sea en una lista de datos puntuales o gráficos de las diferentes respuestas de los integradores, también se puede realizar un reporte impreso de cada uno de las partes. El microprocesador 6809 es de 16 bits y lo encontramos como controlador del TRS-80 COLOR COMPUTER II, por medio de la programación assembler desarrollamos un programa que efectúa las conexiones de los integradores digitales e interactúa el flujo de datos entre ellos; estos integradores se los ubica en la memoria RAM asignándoles una parte de la misma. A medida que se va ejecutando el proceso se transmite los resultados por medio de la interface RS232-C.

El procedimiento a seguir para que la máquina DDA comience a ejecutar el proceso es:

- Conectamos el cartucho ensamblador para que se ejecute el sistema operativo ensamblador y el

microprocesador 6809 está listo para almacenar el programa del DDA.

- Cargamos el programa que se encuentra almacenado en un cassette por medio de una cassetera.
- Ensamblamos y ejecutamos el programa en el ZBUG del ensamblador.
- En el microcomputador IBM-PC ejecutamos el programa DDA para introducir los valores que adquieren los integradores bajo un formato preestablecido.
- Una vez que los datos de entrada se graban en un archivo ejecutamos la comunicación bajo interface RS232-C con el microprocesador 6809, transmitiendo el archivo de entrada y abrimos un archivo para recibir los datos de salida que es la respuesta de los integradores digitales que queremos analizar.
- Efectuada la comunicación de datos, pasamos a analizar los datos recibidos, ya sea observando en la pantalla los valores puntuales en función de la variable independiente u observamos los gráficos de cada una de las respuestas de los integradores, podriamos además emitir un reporte impreso de cualquiera de las dos formas de presentación de resultados.

Nombre: DGA
 Clasificación: 1000000
 Fecha: 10/10/77
 Autor: 1000000

Nombre: Tablas de datos
 Nombre: Datos
 Descripción:
 Datos de entrada y salida

2.3 SEUDOCODIGO DEL PROGRAMA EN EL CPU PARA EL MANEJO DE LOS DATOS DE ENTRADA, SALIDA Y COMUNICACION DE LA MAQUINA DDA

1. Inicio
 2. Se lee el dato de entrada
 3. Se lee el dato de salida
 4. Se lee el dato de comunicación
 5. Se lee el dato de entrada
 6. Se lee el dato de salida
 7. Se lee el dato de comunicación
 8. Se lee el dato de entrada
 9. Se lee el dato de salida
 10. Se lee el dato de comunicación
 11. Se lee el dato de entrada
 12. Se lee el dato de salida
 13. Se lee el dato de comunicación
 14. Se lee el dato de entrada
 15. Se lee el dato de salida
 16. Se lee el dato de comunicación
 17. Se lee el dato de entrada
 18. Se lee el dato de salida
 19. Se lee el dato de comunicación
 20. Se lee el dato de entrada
 21. Se lee el dato de salida
 22. Se lee el dato de comunicación
 23. Se lee el dato de entrada
 24. Se lee el dato de salida
 25. Se lee el dato de comunicación
 26. Se lee el dato de entrada
 27. Se lee el dato de salida
 28. Se lee el dato de comunicación
 29. Se lee el dato de entrada
 30. Se lee el dato de salida
 31. Se lee el dato de comunicación
 32. Se lee el dato de entrada
 33. Se lee el dato de salida
 34. Se lee el dato de comunicación
 35. Se lee el dato de entrada
 36. Se lee el dato de salida
 37. Se lee el dato de comunicación
 38. Se lee el dato de entrada
 39. Se lee el dato de salida
 40. Se lee el dato de comunicación
 41. Se lee el dato de entrada
 42. Se lee el dato de salida
 43. Se lee el dato de comunicación
 44. Se lee el dato de entrada
 45. Se lee el dato de salida
 46. Se lee el dato de comunicación
 47. Se lee el dato de entrada
 48. Se lee el dato de salida
 49. Se lee el dato de comunicación
 50. Se lee el dato de entrada
 51. Se lee el dato de salida
 52. Se lee el dato de comunicación
 53. Se lee el dato de entrada
 54. Se lee el dato de salida
 55. Se lee el dato de comunicación
 56. Se lee el dato de entrada
 57. Se lee el dato de salida
 58. Se lee el dato de comunicación
 59. Se lee el dato de entrada
 60. Se lee el dato de salida
 61. Se lee el dato de comunicación
 62. Se lee el dato de entrada
 63. Se lee el dato de salida
 64. Se lee el dato de comunicación
 65. Se lee el dato de entrada
 66. Se lee el dato de salida
 67. Se lee el dato de comunicación
 68. Se lee el dato de entrada
 69. Se lee el dato de salida
 70. Se lee el dato de comunicación
 71. Se lee el dato de entrada
 72. Se lee el dato de salida
 73. Se lee el dato de comunicación
 74. Se lee el dato de entrada
 75. Se lee el dato de salida
 76. Se lee el dato de comunicación
 77. Se lee el dato de entrada
 78. Se lee el dato de salida
 79. Se lee el dato de comunicación
 80. Se lee el dato de entrada
 81. Se lee el dato de salida
 82. Se lee el dato de comunicación
 83. Se lee el dato de entrada
 84. Se lee el dato de salida
 85. Se lee el dato de comunicación
 86. Se lee el dato de entrada
 87. Se lee el dato de salida
 88. Se lee el dato de comunicación
 89. Se lee el dato de entrada
 90. Se lee el dato de salida
 91. Se lee el dato de comunicación
 92. Se lee el dato de entrada
 93. Se lee el dato de salida
 94. Se lee el dato de comunicación
 95. Se lee el dato de entrada
 96. Se lee el dato de salida
 97. Se lee el dato de comunicación
 98. Se lee el dato de entrada
 99. Se lee el dato de salida
 100. Se lee el dato de comunicación

SEUDOCODIGO

PDD-002A

Asunto: DDA

Autor: RPP

Página: 1

Clasificación: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

Iniciar Tablas y Contadores;

Mostrar Menu Principal;

REPEAT

Seleccionar Opcion (Opcion);

CASE Opcion OF

1: BEGIN

Write ('Nombre del archivo e datos a crear:');

Leer Nombre de Archivo (NombArch,Ok);

IF Ok THEN

BEGIN

Colocar extension '.DDA' a NombArch;

LeeArchivo (NombArch);

Mostrar Menu Principal;

Mostrar 'Grabar S/N';

Leer Decision (Decision);

IF Decision = 'S' THEN

Graba(NombArchivo);

END;

END;

2: BEGIN

Write ('Nombre del archivo de datos:');

Leer nombre de archivo (NombArch,Ok);

IF Ok THEN

BEGIN

IF NombArch = '' THEN

BEGIN

Cargar archivos (*.DDA',ListaArch);

IF Lista no Vacía THEN

Escoger un archivo (ListaArch,NombArch,Ok);

END;

END;

IF Ok AND NombArch <> '' THEN

BEGIN

Colocar extension '.DDA' a NombArch;

LeeArchivo (NombArch);

Mostrar Menu Principal;

Mostrar ('Grabar Si');

Mostrar ('Grabar No');

Mostrar ('Nuevo Archivo');

Leer Decision (Decision);

IF Decision = 'Si' THEN

PDD-002A

SEUDOCODIGO

Sujeto: DDA

Autor: RPP

Página: 2

Clasificación: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

```

    Graba (NombArch)
ELSE Decision = 'Nuevo Archivo' THEN
    BEGIN
        Leer Nombre de Archivo (NombArch,Ok);
        IF Ok THEN
            BEGIN
                Colocar extension '.DDA' a NombArch;
                Graba (NombArch);
            END;
        END;
    END;
END;
3: BEGIN
    Write ('Verifique si el diskette tiene')+
        'suficiente capacidad');
    Write ('Est. listo para realizar la comunicaci"n');
    Leer Decision (Decision);
    IF Decision = 'S' THEN
        BEGIN
            Colocar extension '.DDE' a NombArch;
            WriteLn ('Comunicaci"n realiz ndose ...');
            Comunica (NombArch);
        END;
    END;
4: BEGIN
    Mostrar ('1: Datos de Entrada');
    Mostrar ('2: Datos de Salida');
    Mostrar ('3: Graficos');
    Seleccionar Salida;
    CASE Salida OF
        1: BEGIN
            Mostrar ('1: Por Pantalla');
            Mostrar ('2: Por Impresora');
            CASE DriveSalida OF
                1: DatosEnt (NombArch);
                2: DatosEntImp (NombArch);
            END;
        END;
        2: BEGIN
            Colocar extension '.DDS' a NombArch
            Mostrar ('1: Por Pantalla');
            Mostrar ('2: Por Impresora');

```

SEUDOCODIGO

PDD-002A

Sujeto: DDA

Autor: RPP

Página: 3

Clasificación: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

```

CASE DriveSalida OF
  1: Listar (NombArch);
  2: ListarImp (NombArch);
  END;
3: BEGIN
  Marcar maximo 3 salidas para graficar;
  Cargar datos de graficos marcados;
  Graficar (NombArch,nint,xst,xfn,cn[i],cy[i]);
  END;
END;
END;
5: BEGIN
  Mostrar menu de ayuda;
  Seleccionar Ayuda;
  CASE Ayuda OF
    1: Listar archivos del disco actual;
    2: BEGIN
      Mostrar discos disponibles;
      Seleccionar Disco (ValDisco);
      FijarDrive (ValDiso);
      END;
    3: BEGIN
      Mostrar base de impresion por defecto;
      IF base impresion = 'TIEMPO' THEN
        Fijar base de impresion a 'TIEMPO'
      ELSE
        BEGIN
          Leer Nombre de nueva base de impresion;
          Fijar base de impresion al nombre leído;
        END;
      END;
    4: BEGIN
      Mostrar archivos del disco actual;
      REPEAT
        Marcar archivos para borrar;
        Esperar Tecla;
        IF Tecla = 'Aceptar borrar' THEN
          Borrar archivos marcados;
        UNTIL Tecla = 'ESC';
      END;
    5: Retornar al menu principal;
  END;
END;

```

SEUDOCODIGO

PDD-002A

Proyecto: DDA

Autor: RPP

Página: 4

Identificación: LeeArchivo

Versión: 1.0

Fecha: 10-OCT-90

REPEAT

Lect = 1;

CASE Lect OF

1: BEGIN

Mostrar Pantalla DDASCR02;

Leer(Nint,S_bit,ValIni,ValFin,PasoImp);

END;

2: BEGIN

IF Nint <= 6 THEN

Mostrar parte baja de DDASCR02 para Nint líneas

ELSE

Mostrar parte baja de DDASCR02 para 6 líneas;

Leer (NombVar,dy1,dy2,dy3,dy4,dy5,dx);

END;

3: BEGIN

Mostrar Pantalla DDASCR03;

IF Nint <= 6 THEN

Mostrar parte baja DDASCR03 para Nint líneas

ELSE

Mostrar parte baja de DDASCR03 para 6 líneas;

Mostrar (Nint,NombVar);

Leer(Sale,FactEscala,P_bit,CondIni);

END;

END;

Leer Tecla de control de lectura (Tecla);

CASE Lect OF

1: IF Tecla = 'ENTER' THEN

Lect = 2;

2: IF Tecla = 'ENTER' THEN

Lect = 3

ELSE IF Tecla = 'ESC' THEN

Lect = 1;

3: IF Tecla = 'ENTER' THEN

Lect = 4

ELSE IF Tecla = 'ESC' THEN

Lect = 2;

END;

UNTIL Tecla = 4;

PDD-002A

SEUDOCODIGO

Proyecto: DDA

Autor: RPP

Página: 5

Identificación: Graba

Versión: 1.0

Fecha: 10-OCT-90

```

BEGIN
  Abrir disco;
  Abrir archivo (NombArch);
  EscribirEnArchivo (Nint,S_Bit,ValIni,ValFin,PasoImp);
  FOR i := 1 TO Nint DO
    EscribirEnArchivo (NombVar,dy1[i],dy2[i],dy3[i],dy4[i],
                      dy5[i],dx[i]);
  FOR i := 1 TO Nint DO
    EscribirEnArchivo (Sale[i],FactEscala[i],P_bit[i],CondIni[i]);
  Cerrar Archivo (NombArch);
  Colocar Extension ".DDE" a NombArch;
  Abrir archivo (NombArch);
  TotalIter = 2^Nite * (ValFin-ValIni) + PasoImp;
  Transformar a cadenas (Nite,TotalIter,PasoImp);
  EscribirEnArchivo (NiteStr,TotalIterStr,PasoImpStr);
  FOR i := 1 TO Nint DO
    BEGIN
      Transformar a cadenas (dy1[i],dy2[i],dy3[i],dy4[i],
                          dy5[i]);
      EscribirEnArchivo (NombVar,dylstr,dy2str,dy3str,dy4str,
                          dy5Str);
    END;
  InicialCadenas (S1,S2,S3,S4,S5);
  FOR i := 1 TO Nint DO
    BEGIN
      Yi = 2^Nite * CondIni[i];
      Transformar a cadenas (dx[i],FactEscala[i],P_bit[i],Yi);
      S1 = S1 + ' ' + dxstr;
      S2 = S2 + ' ' + FactEscalastr;
      S3 = S3 + ' ' + P_bitstr;
      S4 = S4 + ' ' + Yistr);
      S5 = S5 + ' ' + Sale [i]);
    END;
  EscribirEnArchivo (S1,S2,S3,S4,S5);
  Cerrar Archivo (NombArch);
END;

```


PDD-002A

SEUDOCODIGO

Respecto: DDA

Autor: RPP

Página: 6

Identificación: DatosEnt

Versión: 1.0

Fecha: 10-OCT-90

BEGIN

EscribirCentrado ('DATOS DE ENTRADA');

Writeln;

EscribirCentrado ('Archivo: ', NombArch);

Writeln;

Writeln ('Numero de integradores : ', Nite);

Writeln ('S_bit de precision : ', S_bit);

Writeln ('Paso de impresion : ', PasoImp);

Writeln ('Valor Inicial : ', ValIni);

Writeln ('Valor Final : ', ValFin);

Writeln;

Writeln ('Nombre del Integrador dy1 dy2 dy3 dy4 dy5 dx',
' n p CondIni');

NumLinea = 15;

FOR i = 1 TO Nint DO

BEGIN

Writeln (NombVar, dy1[i], dy2[i], dy3[i], dy4[i], dy5[i], dx[i],
FactEscala [i], P_bit[i], CodIni[i]);

NumLinea = NumLinea + 1;

IF NumLinea >= 24 THEN

BEGIN

Hacer Pausa;

NumLinea = 15;

END;

END;

END;

PDD-002A

SEUDOCODIGO

Aspecto: DDA

Autor: RPP

Página: 7

Identificación: DatosEntImp

Versión: 1.0

Fecha: 10-OCT-90

```

BEGIN
  EscribirCentrado (LST,'DATOS DE ENTRADA');
  Writeln(LST);
  EscribirCentrado (LST,'Archivo: ',NombArch);
  Writeln(LST);
  Writeln (LST,'Numero de integradores :',Nite);
  Writeln (LST,'S_bit de precision :',S_bit);
  Writeln (LST,'Paso de impresion :',PasoImp);
  Writeln (LST,'Valor Inicial :',ValIni);
  Writeln (LST,'Valor Final :',ValFin);
  Writeln(LST);
  Writeln (LST,'Nombre del Integrador dy1 dy2 dy3 dy4 dy5 dx',
    ' n p CondIni');
  NumLinea = 14;
  FOR i = 1 TO Nint DO
    BEGIN
      Writeln (LST,NombVar,dy1[i],dy2[i],dy3[i],dy4[i],dy5[i],
        dx[i],FactEscala [i],P_bit[i],CodIni[i]);
      NumLinea = NumLinea + 1;
      IF NumLinea >= 55 THEN
        BEGIN
          Saltar Pagina;
          NumLinea = 14;
        END;
    END;
  END;

```

PDD-002A

SEUDOCODIGO

Respecto: DDA

Autor: RPP

Página: 8

Identificación: Graficar

Versión: 1.0

Fecha: 10-OCT-90

```

BEGIN
  Ejecutar disco;
  IF Arch no existe THEN EXIT;
  Habilitar pantalla para alta resolucion;
  Obtener mayor valor en X y Y de las salidas a graficar;
  Calcular escala en eje X;
  Calcular escala en eje Y;
  Dibujar Ejes X y Y;
  Particionar los Ejes y escribir valores;
  IF Sale [1] es SI THEN
    BEGIN
      FOR i := 1 TO TotalPuntos DO
        BEGIN
          Calcular (j,di,dj);
          Graficar recta del punto (i,j,i+di,j+dj);
        END;
      IF Numero de graficas > 1 THEN
        Graficar marca de distincion de sale [1];
      END;
    IF Sale [2] es SI THEN
      BEGIN
        FOR i := 1 TO TotalPuntos DO
          BEGIN
            Calcular (j,di,dj);
            Graficar recta del punto (i,j,i+di,j+dj);
          END;
          Graficar marca de distincion de sale [2];
        END;
      IF Sale [3] es SI THEN
        BEGIN
          FOR i := 1 TO TotalPuntos DO
            BEGIN
              Calcular (j,di,dj);
              Graficar recta del punto (i,j,i+di,j+dj);
            END;
            Graficar marca de distincion de sale [3];
          END;
        Hacer Pausa;
      END;
    END;
  END;

```

PDD-002A

SEUDOCODIGO

Proyecto: DDA

Autor: RPP

Página: 9

Identificación: Listar

Versión: 1.0

Fecha: 10-OCT-90

```

BEGIN
  Fijar disco;
  IF NombArch no existe THEN EXIT;
  Abrir archivo (NombArch);
  Habilitar formato pantalla DDDLST02;
  Numr := ROUND (2^Nite*(ValFin-Vallni)+PasoImp);
  NumInt := Nint Seleccionados;
  WHILE NumInt > 0 DO
    BEGIN
      IF NumInt > 6 THEN
        BEGIN
          SalidasAMostrar := 6;
          NumInt := NumInt - 6;
        END
      ELSE
        BEGIN
          SalidasAMostrar := NumInt;
          NumInt := 0;
        END;
      NumLinea := 1;
      FOR i := 1 TO (Numr DIV (PasoImp+1)) DO
        BEGIN
          Leer del archivo cadena de datos (sttr);
          Temp = (((i-1)*PasoImp+1)/Nite) + Vallni;
          Write ((i-1)*(PasoImp+1):6, ' ', Temp:10:3);
          NumInt := 1;
          REPEAT
            IF Salida Habilitada THEN
              BEGIN
                Extraer valor a mostrar de sttr;
                Convertir a numero entero (Temp);
                Validar rango (Temp);
                IF P_bit [Salida] > 0 THEN
                  BEGIN
                    TempAux = 2^P_bit[Salida];
                    Write (Temp/TempAux:10:3);
                  END;
                END;
              END;
            Pasas a la siguiente Salida;
          UNTIL Salidas mostradas <= SalidasAMostrar;
          WriteLn;
          NumLinea = NumLinea + 1;
        END;
      END;
    END;
  END;

```

SEUDOCODIGO

PDD-002A

Aspecto: DDA

Autor: RPP

Página: 10

Identificación: Listar

Versión: 1.0

Fecha: 10-OCT-90

```
IF NumLinea > 24 THEN
```

```
  BEGIN
```

```
    Hacer Pausa;
```

```
    NumLinea = 1;
```

```
  END;
```

```
END;
```

```
END;
```

```
Cerrar Archivo;
```


SKUDOCODIGO

PDD-002A

Proyecto: DDA

Autor: RPP

Página: 11

Identificación: ListarImp

Versión: 1.0

Fecha: 10-OCT-90

```

    Leer disco;
    IF NombArch no existe THEN EXIT;
    Abrir archivo (NombArch);
    Habilitar formato pantalla DDDLST02;
    Num := ROUND (2^Nite*(ValFin-ValIni)+PasoImp);
    NumInt := Nint Seleccionados;
    WHILE NumInt > 0 DO
        BEGIN
            IF NumInt > 6 THEN
                BEGIN
                    SalidasAMostrar := 6;
                    NumInt := NumInt - 6;
                END
            ELSE
                BEGIN
                    SalidasAMostrar := NumInt;
                    NumInt := 0;
                END;
            NumLinea := 1;
            FOR i := 1 TO (Num DIV (PasoImp+1)) DO
                BEGIN
                    Leer del archivo cadena de datos (str);
                    Temp = (((i-1)*PasoImp+1)/Nite) + ValIni;
                    Write (LST, (i-1)*(PasoImp+1):6, ' ', Temp:10:3);
                    NumInt := 1;
                    REPEAT
                        IF Salida Habilitada THEN
                            BEGIN
                                Extraer valor a mostrar de str;
                                Convertir a numero entero (Temp);
                                Validar rango (Temp);
                                IF P_bit [Salida] > 0 THEN
                                    BEGIN
                                        TempAux = 2^P_bit[Salida];
                                        Write (LST, Temp/TempAux:10:3);
                                    END;
                                END;
                                Pasarse a la siguiente Salida;
                            UNTIL Salidas mostradas <= SalidasAMostrar;
                            WriteLn (LST);
                            NumLinea = NumLinea + 1;

```

SEUDOCODIGO

PDD-002A

Asunto: DDA

Autor: RPP

Página: 12

Especificación: ListarImp

Versión: 1.0

Fecha: 10-OCT-90

```
IF NumLinea > 55 THEN
```

```
  BEGIN
```

```
    Hacer salto de pagina;
```

```
    NumLinea = 1;
```

```
  END;
```

```
END;
```

```
END;
```

```
Cerrar Archivo;
```

SEUDOCODIGO

PDD-002A

Proyecto: DDA

Autor: RPP

Página: 14

Especificación: Comunica

Versión: 1.0

Fecha: 10-OCT-90

```
Recibir(ch);  
EscribirEnArchivo(ch);  
PalabParidad := PalabParidad XOR ord(ch);  
UNTIL ch = EOL OR ch = EOF;
```

```
END;
```

```
error := PalabParidad <> 0;
```

```
UNTIL ch = EOF OR k = 2;
```

```
Cerrar Archivo (NombArch);
```

```
END;
```

```
IF r = chr($FF) OR k = 2 THEN
```

```
  WriteLn ('No existe cable');
```

```
IF error THEN WriteLn ('Error en comunicacion');
```

CHECOQUEO

Nombre: DDA

Apellido: García

Matrícula: 17111111

Fecha: 11/11/11

1. INTRODUCCION 11/11/11 11/11/11 11/11/11
 2. OBJETOS 11/11/11 11/11/11 11/11/11
 3. RESULTADOS 11/11/11 11/11/11 11/11/11

Conclusiones:

El programa se ejecutó correctamente.
 Se obtuvieron los resultados esperados.
 Se verificó la correcta ejecución de los algoritmos.
 Se comprobó la correcta comunicación de la máquina DDA.
 Se concluye que el programa funciona correctamente.

2.4 SEUDOCODIGO DEL PROGRAMA EN ASSEMBLER DEL TRS-80 COCO II PARA EL CONTROL DE LOS INTEGRADORES DIGITALES Y COMUNICACION DE LA MAQUINA DDA

El programa se ejecutó correctamente.
 Se obtuvieron los resultados esperados.
 Se verificó la correcta ejecución de los algoritmos.
 Se comprobó la correcta comunicación de la máquina DDA.
 Se concluye que el programa funciona correctamente.

PDD-002A

SEUDOCODIGO

Proyecto: DDA

Autor: RPP Página: 1

Identificación: PRINCIPAL

Versión: 1.0 Fecha: 10-OCT-90

RECEPCION DE CARACTERES ENVIADOS POR LA IBM - PC QUE
INDICAN LAS CONECCIONES DE LOS INTEGRADORES , SU
ESCALAMIENTO Y SUS VALORES INICIALES *)

inicializo puertos;

REPITO {handshake}

recibo2 (ch) {recibo caracter}

HASTA QUE ch = 'T';

transm ('R'); {transmito caracter}

REPITO

recibo (ch); {recibo caracteres}

almaceno ch en TABLA;

lejo palabra de paridad del grupo de datos;

HASTA QUE ch = 'Y'; {fin de recepcion de caracteres}

lea (palabra de paridad);

seguo palabra de paridad;

TRANSFORMACION DE CARACTERES ASCII A BINARIO *)

SIEMPREAS no sea EOF

REPITO

lea caracteres de TABLA;

almaceno en DATO;

HASTA QUE EOLN o ' ';

SI no es 'S' o 'N' ENTONCES

convierto DATO a binario en TEMPO;

SI es negativo obtengo complemento-2;

almaceno TEMPO en VALOR;

FIN-SI;

FIN-MIENTRAS;

ALMACENO NUMEROS BINARIOS EN SUS RESPECTIVAS VARIABLES *)

asigno a VALOR;

leer dato almaceno en M; {numero de integradores}

leer dato almaceno en FIN; {numero de iteraciones}

leer dato almaceno en PRNT; {numero de paso de salida}

inicializo i:=0; {contador de filas}

REPITO

inicializo j:=0; {contador de columnas}

REPITO

almaceno en SS[i,j]; {conecciones dy de integradores}

PDD-002A

SEUDOCODIGO

Aspecto: DDA

Autor: RPP

Página: 2

Identificación: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

```

    incremento j;
    HASTA QUE j = 5;
    almaceno en XS[i];
    incremento i;
    HASTA QUE i = M;

inicializo i:=0; {contador de filas}
REPITO
    almaceno en SN[i]; {numero de bits n}
    almaceno en SY[i]; {numero de bits p}
    almaceno en YY[i]; {condicion inicial de y}
    almaceno en CY[i]; {condicion de salida, 'S' o 'N'}
    SI CY[i] = 'S' incremento NDASAL {numero de datos de salida}
    incremento i;
    HASTA QUE i = M;

inicializo i:=0; {contador de filas}
REPITO
    inicializo R[i]:=0; {valor inicial del integrando}
    inicializo Z[i]:=0; {valor inicial del dz}
    incremento i;
    HASTA QUE i = M; {ultimo integrador}

inicializo ictr:=0; {contador de iteraciones}
inicializo ndbuf:=0; {contador de datos en el Buffer}
inicializo dtem:=0; {contador de datos emitidos}

* DEFINICION DE CUANTAS CONECCIONES TIENE CADA INTEGRADOR *

inicializo i:=0; {contador de filas}
REPITO
    inicializo DA[i]:=0; {contador de conecciones dy en cada
                          integrador}
    inicializo j:=0; {contador de columnas}
    REPITO
        SI SS[i,j] <> 0 ENTONCES incremento DA[i];
        incremento j;
        HASTA QUE j = 5 o SS[i,j] = 0;
    incremento i;
    HASTA QUE i = M; {ultimo integrador}

```

SEUDOCODIGO

PDD-002A

Respecto: DDA

Autor: RPP

Página: 3

Clasificación: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

* ENVIAR AL PC LOS DATOS DE CADA UNA DE LAS VARIABLES DE LOS INTEGRADORES PROGRAMADOS COMO SALIDA *

inicializo print:=PRNT; {contador de pasos para emitir un resultado}

REPITO

 incremento print;

 SI print = 0 ENTONCES

 conver; {procedimiento de conversión de binario a carácter}

 salida; {procedimiento que transmite los resultados a la IBM-PC}

 FIN-SI;

* ACTUALIZAR LOS VALORES DE DX Y DY DE CADA INTEGRADOR *

 incremento ictr;

 inicializo i:=0; {contador de filas}

 REPITO

 inicializo j:=0; {contador de columnas}

 inicializo dy[i]:=0 {acumulador de las cinco entradas dy}

 REPITO

 SI SS[i,j] = 101 ENTONCES

 incremento dy[i] {variable independiente}

 SINO

 acumulo a dy[i] -1,0,1 del Z[SS[i,j]] {variables dependientes}

 FIN-SI;

 HASTA QUE j = DA[i]; {numero de conexiones al dy}

 SI XS[i] = 101 ENTONCES

 DX[i] := 1 {variable independiente}

 SINO

 asigno a DX[i] -1,0,1 del Z[XS[i]] {variables dependiente}

 FIN-SI;

 HASTA QUE i = M; {ultimo integrador}

* PROCESO DE LOS INTEGRADORES DIGITALES EN LA MAQUINA DDA *

 inicializo i:=0; {contador de integradores}

 REPITO

 actualizamos los parametros de la subrutina que procesa cada integrador)

PDD-002A

SEUDOCODIGO

Aspecto: DDA

Autor: RPP

Página: 4

Identificación: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

```

aa:= DX[i]; {diferencial de entrada dx}
bb:= DY[i]; {diferencial de entrada dy}
ccy:= YY[i]; {valor binario del registro Y}
dd:= R[i]; {valor binario del registro R}
e:= Z[i]; {diferencial de salida dz}
f:= SN[i]; {factor de escala n}
g:= SY[i]; {p - bit de precision del dy}

modtrz(aa,bb,ccy,dd,e,f,g); {proceso del integrador}

Actualizamos los valores de los registros Y, R y el dz}
YY[i]:= ccy;
R[i]:= dd;
Z[i]:= e;
Incremento i; {pasamos al siguiente integrador}
Mientras QUE i = M; {ultimo integrador}

Mientras QUE ictr = FIN; {fin del proceso de simulacion}

```

PDD-002A

SEUDOCODIGO

Sujeto: DDA

Autor: RPP

Página: 5

Identificación: RCBCA

Versión: 1.0

Fecha: 10-OCT-90

* PROCEDIMIENTO RCBCA PARA RECIBIR CARACTERES *

PROCEDIMIENTO rcbca(RSIN); {caracter recibido en RSIN}

{contacto bit de inicio}

B=0;

{ chequeo bit 0 de la palabra FF22;

Mientras QUE bit 0 = 0;

{muestreo la señal de entrada luego de 833.3µs*B}

B:=4; {mitad del retardo de muestreo}

esper(B); {subrutina de retardo}

inicializo nbit:=8; {contador de bits a receptor}

inicializo part:=0; {indicador del bit de paridad}

B=0;

{muestreo la señal de entrada luego de 833.3µs*B}

B:=9;

esper(B);

Si bit recibido = 1 ENTONCES incremento part;

decremento nbit;

Mientras QUE nbit = 0; {fin de recepcion de caracter de 8 bits}

{muestreo la señal de entrada luego de 833.3µs*B}

B:=4;

esper(B);

recibo bit de paridad;

Si bit de paridad recibido <> bit de paridad en part ENTONCES

error en recepcion;

Fin (procedimiento rcba)

PDD-002A

SEUDOCODIGO

Proyecto: DDA

Autor: RPP

Página: 7

Identificación: ESRP

Versión: 1.0

Fecha: 10-OCT-90

* PROCEDIMIENTO ESRP PARA RETARDO DE TIEMPO EN EL MUESTREO DE LA
SEÑAL DE ENTRADA, DEPENDE DEL VALOR DEL REGISTRO B *)

PROCEDIMIENTO espr(B);

Depende del dato en B para retardo de 833.3µs

INICIO

 Habilito interrupcion IRQ

 Mascaro la direccion FF01H con 00110111B ;

 Espero por llegada de interrupcion

 CALL EPH; {llamada de interrupcion}

 Incremento B;

 Mientras QUE B = 0; {fin de retardo}

FIN {procedimiento espr}

* SUBROUTINA LLAMA DE INTERRUPCION IRQ *)

Cuando IRQ es habilitado salta a esta subrutina

SUBROUTINA llama;

 Habilito interrupcion IRQ

 Mascaro la direccion FF01H con 00110110B;

 Seteo bit; {bit 0 de FF22H}

 Mascaro bit en RSIN;

FIN {subrutina llama}

PDD-002A

SEUDOCODIGO

Proyecto: DDA

Autor: RPP

Página: 8

Especificación: TRANSM

Versión: 1.0

Fecha: 10-OCT-90

* PROCEDIMIENTO TRANSM PARA TRANSMITIR CARACTERES *

PROCEDIMIENTO transm(A); {dato a transmitir en el registro A}

carga A en RSOUT;

inicializo B:= 9;

 esper(B); {retardo de $833.3\mu s * B$ }

carga bit 1 de FF20H; {bit de inicio}

inicializo nbit:= 8; {numero de bit a transmitir}

B:= 9;

transmite bit; {rota RSOUT para transmitir cada bit a través del bit 1 de FF20H}

esper(B); {espero fin de retardo para transmitir el siguiente bit}

decremento nbit;

Mientras QUE nbit = 0; {fin de transmisión de carácter}

transmite bit de paridad;

 esper(B); {retardo de $833.3\mu s * B$ }

carga bit 1 de FF20H; {bit de parada}

esper(B);

incremento ndbuf; {contador de datos en buffer}

FIN {procedimiento transm}

SEUDOCODIGO

PDD-002A

Sujeto: DDA

Autor: RPP

Página: 9

Clasificación: SALIDA

Versión: 1.0

Fecha: 10-OCT-90

* PROCEDIMIENTO SALIDA PARA TRANSMITIR LOS DATOS DE LOS
 INTEGRADORES ALMACENADOS EN LA DIRECCION ASCI *)

PROCEDIMIENTO salida;

 inicializo i:=5; {contador de retardo}

 trans ('T'); {handshake}

 SI

 decremento i;

 recibo2(ch); {recibo caracter}

 Mientras QUE i = 0 o ch = 'R';

 SI i=0 ENTONCES retorno inicio del programa; {no existe cable}

 inicializo j:=NDASAL; {contador de datos a transmitir}

 SI

 inicializo i:=5; {contador de bytes por dato}

 SI

 trans(ch); {caracter en la direccion ASCI}

 decremento i;

 Mientras QUE i = 0; {fin de transmision de un dato}

 decremento j;

 Mientras QUE j = 0; {fin de transmision de todos los datos}

 SI (ctr + PRNT) > FIN ENTONCES

 trans(EOF); {fin de transmision del proceso}

 SI

 trans(EOL); {fin de transmision de un grupo de datos}

 acumulo dtem en ndbuf; {contador de datos en el buffer del PC}

 SI ndbuf > 5000H ENTONCES {maximo numero de datos en el buffer}

 trans('Y'); {codigo de espera para IBM-PC}

 delay; {subrutina de retardo}

 encero ndbuf; {buffer vacio}

 FIN-SI;

 FIN-SI;

FIN {procedimiento salida}

SEUDOCODIGO

PDD-002A

Proyecto: DDA

Autor: RPP

Página: 10

Especificación: MODTRZ

Versión: 1.0

Fecha: 10-OCT-90

PROCEDIMIENTO MODTRZ PROCESO DE INTEGRACION DE UN INTEGRADOR
 CENTRAL *)

```

PROCEDIMIENTO modtrz(aa,bb,ccy,dd,e,f,g);
  n ← f y g en l; {longitud del registro R}
  cc ← bb en ccy; {actualizo valor del registro Y con el dy}
  resto ← 0 resto en el registro R el valor del registro Y
  dependiendo del dx}
  cc ← dd + sgn(aa)* (ccy + bb/2); {integracion trapezoidal}
  max ← 21; {maximo valor del registro R}
  SI cc > max ENTONCES
    cc ← -1; {seteo dz}
    resto en dd valor max; {actualizo en R el residuo}
  SI SI cc = max ENTONCES
    cc ← 0; {encero dz}
  SI NO
    cc ← 1; {seteo dz}
  FIN SI;
FIN {procedimiento modtrz}

```

SEUDOCODIGO

PDD-002A

Proyecto: DDA

Autor: RPP

Página: 11

Especificación: CONVER

Versión: 1.0

Fecha: 10-OCT-90

PROCEDIMIENTO CONVER PARA CONVERTIR DATOS BINARIOS DE LOS
 "BITSETOS Y" DE SALIDA EN CARACTERES (5 bytes por dato),
 PARA SER TRANSMITIDOS A PARTIR DE LA DIRECCION ASCII *}

PROCEDIMIENTO conver;

Inicializo i:=0; {contador de integradores}

REPTO

SI YY[i] = 'S' ENTONCES {condicion de salida}

convierto valor YY[i] en dato a transmitir; {5 bytes por dato}

almaceno dato en direccion ASCII; {buffer de transmision}

FIN-SI;

Incremento i;

HASTA QUE i = M; {ultimo integrador}

FIN {procedimiento conver}



FIN FIN

FIN FIN

FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN

FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN
 FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN

FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN

FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN

FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN FIN

CAPITULO III

CONECCION Y PRUEBAS

EL DIAGRAMA CIRCUITAL DE LA CONECCION PARA LA COMUNICACION DE DATOS ENTRE EL CPU Y EL TRS-80.

La conexión entre el CPU y TRS-80 lo hacemos por medio de un cable por el cual se transmiten los datos serialmente bajo interface RS-232C y software adicional en ambas computadoras personales.

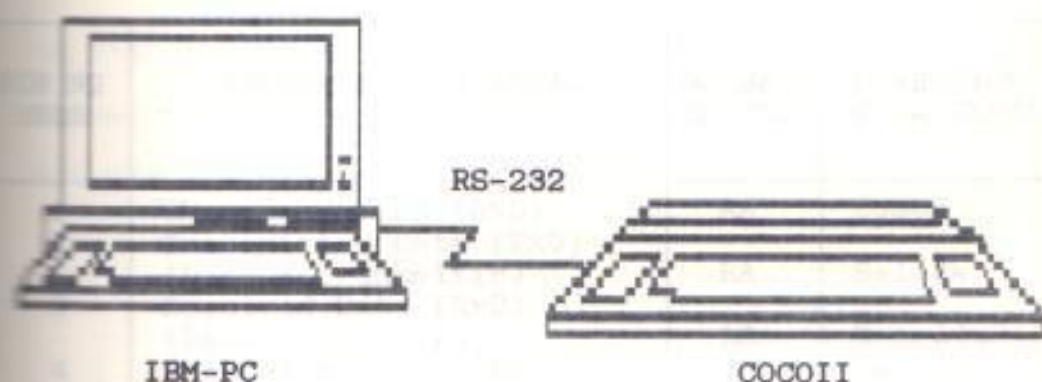


FIGURA 3.1 CONECCION ENTRE CPU Y TRS-80

En IBM-PC se utiliza el puerto asíncrono COM1 que se encuentra disponible.

En COCOII la interface RS-232C.

La interface de comunicación para la IBM-PC es la RS-232C con conector DB25.

- La siguiente figura ilustra la localización de pines del conector DB25 ¹.

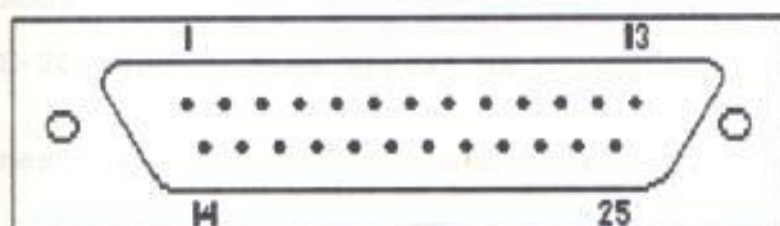


FIG. 3.2 CONECTOR DB25

La siguiente tabla es una lista de las señales de asignación del conector DB25 para poder utilizar el puerto serial del IBM-PC.

PIN N° DB25	NOMBRE DE LA SENAL	NOMBRE RS-232C	DIRECCION DE LA SENAL
1	Chassis Ground (GND)	AA	Común
2	Transmitted Data (TxD) (Dato a Transmitir)	BA	Salida
3	Received Data (RxD) (Dato a Recibir)	BB	Entrada
4	Request to Send (RTS)	CA	Salida
5	Clear to Send (CTS)	CB	Entrada
6	Data Set Ready (DSR)	CC	Entrada
7	Signal Ground (GND) (Señal de Tierra)	AB	Común
8	Data Carrier Detect (DCD)	CF	Entrada
20	Data Terminal Ready (DTR)	CD	Salida

TABLA ASIGNACION DE SENALES DEL CONECTOR DB25.

La interface RS-232C para COCOII utiliza un conector DIN de 4 pines². Las cuatro señales usadas para esta interface son:

- 1.- CD .- Una línea de entrada de Estado
- 2.- RS-232IN - Entrada serial de Datos
- 3.- GROUND - Voltaje de referencia cero
- 4.- RS-232OUT - Salida serial de datos

Los pines del conector DIN son mostrados en la figura:

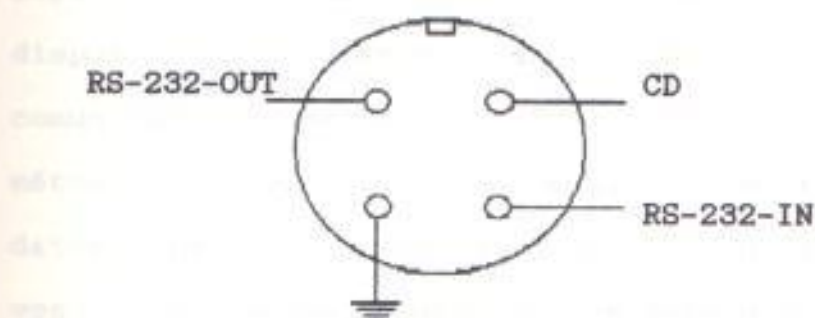


FIG. 3.3 CONECTOR DIN.

La fig. 3.4 muestra el diagrama de conecciones para la comunicación serie entre la COCOII y la IBM-PC.

L/S SERIE DE LA TRS-80
COCOII

CONECTOR DE 25 LINEAS
(RS-232) IBM-PC

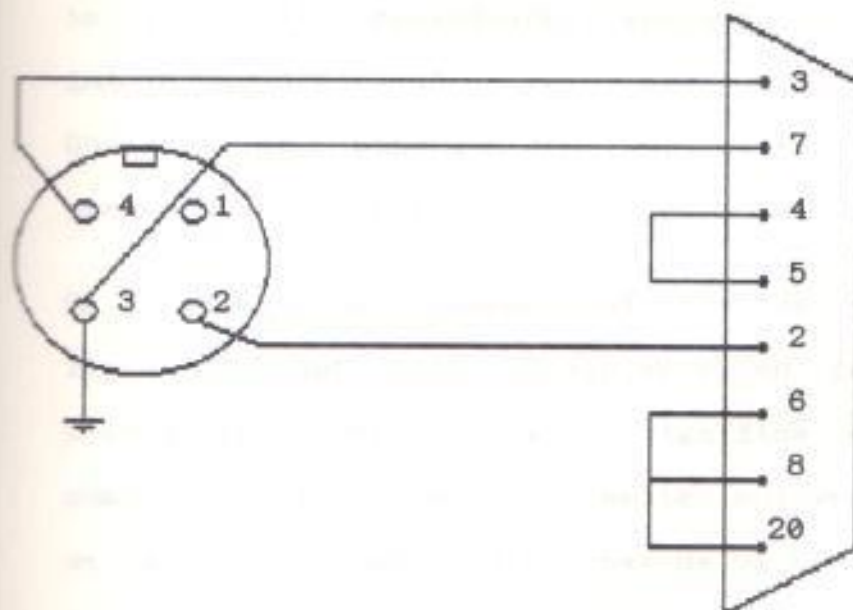


FIG. 3.4 CABLEADO

3.1.1 CONSIDERACIONES DE LA COMUNICACION SERIAL EN EL PC.³

La tarjeta adaptadora de comunicación, con plugs en una ranura de expansión, da al PC la capacidad de comunicar con otros computadores o dispositivos usando una interface de comunicación serial RS-232C standard. El método de comunicación serial, transmite los datos y la información del control, un bit cada vez. El ancho de cada bit es determinado por la velocidad de transmisión de datos, la cual se mide en bits por segundo (baudios), transmitiremos a 1200 baudios. Cuando el dato no es transmitido, la línea esta en voltaje alto (1 lógico) o estado de marcar. Cuando iniciamos la transmisión, enviamos el bit de inicio que será un voltaje bajo (0 lógico). Continúa los 8 bits que son los del caracter que queremos transmitir.

Se transmite seguidamente el bit de paridad impar que debe ser consistente en toda la transmisión, paridad impar significa que el número de bit 1 que se transmite incluyendo el de paridad sera un número impar de veces.

Luego existen un bit de parada para poder

reiniciar el envío de datos.

Debido a que los caracteres pueden ser enviados en cualquier momento, la comunicación serial se dice que es asincrónica.

El IBM-PC utiliza un UART (Universal Asynchronous Receiver Transmitter) 8250, que es el que realiza la transmisión y recepción de los datos serialmente, evitándonos realizar la conversión paralelo-serial, serial-paralelo.

Antes tenemos que programarlo con las características deseadas del protocolo serial. El UART añade los bits de inicio, parada y paridad a el caracter que queremos transmitir. El dato a transmitir lo almacenamos en un registro del UART (Transmitter Shift Register) y el dato que se recepta lo leemos de otro registro (Receiver Shift Register).

Periodicamente se chequea a estos registros para ver si el registro esta en capacidad de aceptar otro dato.

Cuando el dato es receptado es porque ha chequeado previamente el UART que los bits de parada, inicio y paridad han sido los apropiados. El computador sabe que otro byte de datos puede ser leído cuando el bit de status

Receiver Data Ready es el correcto; el UART provee una señal de interrupción, para no tener que sensar el bit de status, sino que el UART interrumpe el programa para que éste almacene el dato que se receiptó.

Para programar el 8250 UART nosotros tenemos acceso a 10 registros cuyas direcciones aparecen en la siguiente tabla:

DIRECCION PUERTO I/O	ENTRADA O SALIDA	REGISTRO SELECCIONADO
3F8H (a)	Salida	Registro de Transmisión (Transmitter Holding Register)
3F8H (a)	Entrada	Registro de Recepción (Receiver Data Register)
3F8H (b)	Salida	Divisor velocidad en baudios (LSB)
3F9H (b)	Salida	Divisor velocidad en baudios (MSB)
3F9H (a)	Salida	Registro habilita interrupción
3FAH	Entrada	Registro Identificación Interrupción.
3FBH	Salida	Registro Control de Línea
3FCH	Salida	Registro Control de Modem
3FDH	Entrada	Registro Status de Línea
3FEH	Entrada	Registro Status de Modem

(a) Bit 7 de registro control de línea = 0

(b) Bit 7 de registro control de línea = 1

Cinco de estos registros se los programan con una instrucción OUT, una vez que son inicializados se los ignora durante el tiempo que se realiza la transmisión y recepción de datos.

Estos registros son:

Divisor de velocidad en baudios (LSB)

Divisor de velocidad en baudios (MSB)

Registro de control de linea

Registro de control de modem

Registro habilita interrupción

El primero de estos, el divisor de velocidad en baudios nos sirve como divisor de frecuencia del reloj del PC que es de una alta frecuencia.

Primero seteamos el bit 7 del registro de control de linea a 1 con una instrucción OUT a la dirección 3FBH. Luego, como queremos transmitir a 1200 baudios enviamos un 00H al LSB del divisor y un 60H al MSB (esta información nos proporciona en cualquier manual del IBM-PC). Después inicializamos el registro de control de linea; éste determina la longitud del caracter, número de bits de parada, y el tipo de paridad. El bit 7 lo ponemos a 0 para que multiplexe el registro 3F8H como registro de datos para entrada/salida.

En el registro de control de Modem solamente nos aseguraremos que el bit 4 este en 0 pues si está en 1 la transmisión y recepción se la prueba sin conexión física, para verificar que el

protocolo está bien establecido.

El último registro que necesitamos inicializar es el registro de habilitación de interrupción. Si la interrupción no está siendo usada, entonces este registro deberá ser seteado a 0.

Una vez que el 8250 ha sido inicializado, podemos efectuar la comunicación serie.

El registro de status de línea se usa para decirnos cuando en la entrada o salida existe un caracter. Si nosotros queremos transmitir un caracter, debemos leer el registro y chequear el bit 5, hasta que este sea 1.

Una vez que transmitimos el caracter del "Transmitter Holding Register", el bit 5 del registro de status de línea se vuelve 0.

El bit 0 del registro de status de línea es 0 y se vuelve 1 cuando un caracter de datos ha sido recibido por el 8250 y se lo coloca en el "Receiver Data Register". El programa detecta esta condición y almacena el caracter recibido, y así el bit 0 del registro de status de línea se vuelve 0 hasta que el siguiente caracter es recibido por el 8250.

recepción en COCOII nosotros debemos acceder a los siguientes registros⁴:

FF20	bit 1	Dato de salida para RS-232
FF22	bit 0	Dato de entrada para RS-232
FF01	bit 0	0 Deshabilita IRQ 1 Habilita IRQ

Control del reloj de sincronismo horizontal (63.5 μ seg)

RSIN Nombre de la dirección de memoria donde se almacena el dato de entrada.

RSOUT Nombre de la dirección de memoria donde se encuentra el dato a transmitir.

Si queremos transmitir un dato en COCOII seguimos los siguientes pasos:

- a) Guardamos el caracter en la dirección de memoria RSOUT.
- b) A través del bit 1 del registro FF20 enviamos bit de inicio.
- c) Habilitamos la interrupción IRQ por el bit 0 del registro FF01.
- d) Contamos nueve veces la llamada a la subrutina de interrupción antes de enviar otro bit (deshabilitamos IRQ en la subrutina de interrupción).

- e) Transmitimos los 8 bits del caracter de RSOUT, repitiendo los pasos c) y d).

Si queremos receptar un dato en COCOII seguimos los siguientes pasos:

- a) Chequeo bit 0 del registro FF22 hasta que este sea 0 indicando inicio de recepción.

- b) Habilito interrupción IRQ por el bit 0 del registro FF01.

- c) Contamos nueve veces la llamada a la subrutina de interrupción antes de receptar otro bit (deshabilitamos IRQ en la subrutina de interrupción)

- d) Receptamos los 8 bits a través del bit 0 del registro FF22 repitiendo los pasos b) y c), y continuamente los vamos almacenando en la dirección de memoria de RSIN.

- e) Receptamos el bit de paridad impar y comprobamos si la recepción se ha efectuado correctamente, sino generamos un estado de error.

- f) Receptamos el bit de parada para disponer del caracter que se encuentra almacenado en RSOUT.

EL MAPEO DE LA MEMORIA DEL TRS-80 PARA LA UBICACION DE LOS REGISTROS QUE SE COMPONE EL INTEGRADOR DIGITAL.

Buffer de recepción, transmisión y cálculos

2E32	DATO	5
2E37	TABLA	300
2F63	TEMP	1
2F64	ASCI	80
2FB4	VALOR	600
3217	TEMPO	2

Variables actualizadas en transmisión y recepción

320C	NBIT	1
320D	ERROR	1
320E	RSIN	1
320F	PART	1
3210	PPAR	1
3211	RSOUT	1
3212	NDBUF	2
3214	DTEM	2
3216	NDASAL	1

Variables utilizadas en los cálculos

3336	DA	20
334A	ICTR	2
334C	I	1
334D	J	1
3362	POS	1

3363	SIGN	1
3364	J1	1
3386	RESTA	2
3388	DIVD	3
338B	RS	2
338D	K	10
3374	Q	1

Variables utilizadas para definición y conexión de los
integradores:

3219	M	1
321A	FIN	2
321C	PRINT	1
321D	PRNT	1
321E	SS	100
3382	XS	20
3296	CY	20
32AA	SN	20
32BE	SY	20

Variables utilizadas como registros del integrador
digital:

3202	R	40
32FA	Z	20
330E	YY	40
334E	DY	20
3365	XI	20

variables requeridas.

f) Opciones que ayudan al usuario al manejo de los archivos.

Se comenzó por la pantalla de manejo de los datos de entrada.

Se dividió la pantalla en tres partes, en la primera tercera parte se ingresan los datos generales como son: número de integradores, la variable s que es el factor de escalamiento de la variable independiente, valor inicial y final de la simulación y valor de impresión o número de iteraciones que efectúa el integrador para emitir un resultado.

Fuimos verificando que el valor que se ingrese en cada una de las celdas sea acorde a como se lo declaró en el programa, si no lo era impide el ingreso del dato.

Al presionar la tecla de ingreso, dependiendo del número de integradores, se publican en las otras dos terceras partes las celdas para ingresar las conexiones de los integradores. Cada fila representa un integrador, la primera celda corresponde al nombre de la variable que maneja dicho integrador; las cinco celdas siguientes se ingresa el número, del integrador

que va a una de las cinco entradas para los dy.

Si el número es negativo es que proviene de un integrador cuya salida esta invertida, se verifica entonces que los números que se ingresen sean enteros positivos o negativos y que no sean mayores al número de integradores, si el número es cero no existe conexión y si es 101 la señal de entrada es el diferencial de la variable independiente.

La última celda corresponde a la entrada independiente que solamente puede ser números positivos que provienen de la señal de salida de otros integradores o si es 101 será la señal de entrada o sea la variable independiente.

Máximo en cada pantalla se publican 6 filas y presionando la tecla de ingreso pasamos a la siguiente pantalla, con la tecla escape regresamos a la pantalla anterior. Estas especificaciones fueron hechas para máximo 80 integradores.

La siguiente pantalla es el escalamiento de los integradores y de la misma manera cada fila representa un integrador y se encuentra subdividida en celdas. La primera ya es fija y es el nombre de la variable ingresada

almacenados en su disco. Estos archivos se los distingue por la extensión ".DDA". En esta parte se verifica que el nombre del archivo este acorde a las reglas impuestas por su sistema operativo DOS y que si el número de archivos creados excede a 10 con presionar la tecla de la letra "C" (continua) se accedan a otros archivos.

Al finalizar el ingreso de datos aparecen 3 opciones que son:

Grabar los datos, así se guardan los datos en el archivo que se quiere crear o cambiar los datos de un archivo ya creado.

Cancelar, esto es no grabar los datos si se llegara a arrepentir de algun valor que se modifíco o sencillamente solo accedamos a ellos para consulta.

Grabar en nuevo archivo, esta opción se creo para que al cambiar un detalle en los datos de ingreso poder comparar los resultados con el original sin cambiarlo a éste.

Se verificó el manejo de estas pantallas en los diversos ejemplos que se realizaron y se reportan en la tesis.

Por medio del programa una vez grabado este archivo de datos se crea un archivo que esta en ASCII y son los datos resumidos que se van a enviar al COCOII y esta compuesto por lineas de la siguiente manera:

- Número de integradores
- Número de iteraciones
- Número de iteraciones ejecutadas antes de recibir un dato.
- Arreglo de valores en que cada fila corresponde un integrador y cinco columnas en las cuales se registra el número del integrador que se conecta el dz al dy del integrador presente. El número de líneas depende del número de integradores.
- Arreglo en una línea de las conecciones dx, los números representan de que integrador proviene la conección o si es "101" la señal es de la variable independiente.
- Arreglo en una línea de los valores "s" o "n" que indica si queremos o no los resultados de la señal del integrador.
- Arreglo en una línea de los valores n que es el factor de escalamiento de cada uno de los integradores.

- Arreglo en una línea de los valores p que es la posición en número de bit del valor dy.
- Arreglo de una línea de los valores digitales de las condiciones iniciales de cada uno de los integradores.

Para señalar fin de archivo se inserta una letra "F" que controla el fin de comunicación. Verificamos que para cada uno de los ejemplos que generamos el archivo de comunicación se graba con la extensión ".DDE" y que su formato este de acuerdo a lo expuesto anteriormente, esto lo observamos con cualquier editor de texto.

Los datos de salida son los recibidos en un archivo con extensión ".DDS" y está de acuerdo al siguiente formato:

- Los números están en código ASCII
- Cada fila representa los datos receptados correspondientes a la iteración en la que se desea el valor de los registros y de los integradores.
- En la línea cada cinco caracteres es el valor digital del registro y del integrador y el número depende de cuales de los integradores hemos programado para que transmitan sus resultados a la IBM-PC.

Con un editor de texto podemos verificar si realmente se ha receptado el archivo de acuerdo al formato explicado.

Bajo el menú principal tenemos la opción de "Salida de Datos" y comprenden los reportes tanto de los datos de entrada como los de salida.

El reporte de los datos de entrada puede hacerlo en pantalla o impresora, y con la tecla de ingreso y las teclas de movimientos se escogen cualquiera de las opciones.

En el reporte por pantalla se coloca como título el nombre del archivo de entrada a continuación los valores del número de integrador, valor de s , número de iteraciones para la impresión, valor inicial y final de la variable independiente. En columnas van los nombres de la variable del integrado, cinco conecciones dy , una conección dx , valores de n y p la condición de salida y los valores iniciales que adquieren.

El reporte por impresora se lo hace en 66 columnas, y se escogió como referencia una hoja tamaño standard de 55 líneas.

El reporte de los datos de salida implica la publicación de los valores numéricos que

adquieren las diferentes variables que manipulan los integradores y la graficación de dichos valores.

Para el primero de ellos podemos seleccionar las opciones: por pantalla o por impresora.

Luego procedemos a elegir cuales de las variables de los integradores digitales queremos observar, para esto hemos habilitado una pequeña ventana con los números de los integradores que previamente escogimos en los datos de entrada como de salida, con el simple presionar de la tecla de ingreso se escogen dichos integradores y al salir de la ventana se observan los valores numéricos de las variables de los integradores bajo el siguiente formato:

- El título es el nombre del archivo de salida.
- La primera columna lleva el título de "PASO" podrá indicar que número de iteración corresponden los valores.
- La segunda columna lleva el título del nombre de la variable independiente y en esta columna van los valores que toma la variable independiente para la iteración correspondiente.
- De la tercera a la octava columna cada una de

ellas lleva el título del número de integrador y del nombre de la variable que procesa dicho integrador digital. Bajo estas columnas se presentan los valores reales que toman las variables en la iteración y el valor de la variable independiente correspondientes en la fila.

Con la tecla de ingreso pasamos a la pantalla siguiente y así sucesivamente hasta el valor final que tenga la variable independiente.

Si hemos seleccionado mas de seis integradores para ser publicados, los primeros seis se observan y luego que se termine su presentación; se observan el resto.

En la opción "POR IMPRESORA" se imprime en hoja continua de tamaño standard de 55 líneas; antes de imprimir tenemos que tener habilitada la impresora para lo cual ponemos mensajes de advertencia antes de ejecutar dicha opción.

Verificamos estas opciones con los ejemplos que detallamos en esta tesis, además simulamos funciones sencillas como exponenciales y sinusoidales y comprobamos que los valores reales de estas variables sean iguales a los calculados, así mismo sacamos un impreso sin

tener problemas. Advertimos que el programa se interrumpe si la impresora no se habilita previamente.

Al escoger la opción de graficación al igual que la anterior opción se ha habilitado una pequeña ventana en la cual se seleccionan los números de los integradores de salida, es decir escogemos las variables que queremos graficar, podemos graficar en una pantalla de uno a tres variables. Si tenemos 2 o 3 gráficos sus escalas aparecen en el lado izquierdo y uno en el lado derecho, y en el gráfico para distinguir las diversas curvas, marcamos el inicio y el fin con una señal que puede ser una equis, un rombo y un cuadrado.

Para el gráfico cambiamos el video por medio del programa a alta resolución, por lo que solo podemos escoger un solo color de pantalla. Una vez terminado el gráfico al presionar la tecla de ingreso regresamos al menú principal. Las escalas son automáticas y el programa las escoge según sean los máximos y mínimos de las variables.

La impresión de los gráficos de estas variables se las realiza manejando la INT 05H.

Previamente antes de ejecutar el programa tenemos que ejecutar el programa del DOS llamado GRAPHICS.COM. Al escoger la opción de gráfico por impresora se observa el gráfico por pantalla y si la impresora se encuentra habilitada se imprime el gráfico que se observa, si la impresora no estuviera habilitada el programa no se interrumpe y al presionar la tecla de ingreso pasamos al menú principal. Las verificaciones correspondientes las realizamos con las simulaciones que hemos efectuado, observamos los gráficos de cada una de las variables de los integradores y superponemos 2 o 3 gráficos, luego efectuamos la impresión de ellos.

En el capítulo 4 reportamos los gráficos de las variables que hemos analizado en cada uno de los ejemplos.

Como última opción del menú principal tenemos las ayudas, bajo ésta tenemos:

- Listado de archivos
- Disco
- Cambio de base de impresión
- Borrar archivo

Adjunto a esta ventana tenemos otra que informa:

- Memoria libre, el cálculo se lo hace en Kbytes

- con precisión de 2 decimales.
- Disco, me informa de cual de los discos A, B o C recupero o grabo los datos.
- Var. Base, me indica el nombre de la variable independiente que utiliza, por default tenemos a Tiempo.

Este status lo variamos con las diferentes opciones expuestas, así, si queremos cambiar de disco, ejecutamos la opción disco y aparecen las letras A, B, o C para que indique en cual de ellos quiero trabajar.

Si notamos que tenemos muy poca memoria para hacer un trabajo, podemos listar los archivos y estos se publicaran en la mitad inferior de la pantalla, si alguno no nos es indispensable con la opción borrar archivo lo podemos eliminar, señalándolo con las teclas de movimiento y ejecutando el borrado con la letra "B".

Si queremos cambiar el nombre de la variable independiente, al ejecutar el "cambio de base de impresión", me da una ventana para escribir el nuevo nombre; y con este nombre la variable independiente aparece tanto en los listados de resultados como en los gráficos.

Verificamos estas opciones independientemente de

cualquier ejemplo que hagamos, pues se tratan de opciones de manejo de disco.

En el caso de que estemos simulando un proceso cuya variable independiente es diferente al tiempo, verificamos el cambio del nombre de esta variable en los listados de resultados y la impresión de gráfico.

3.3.2 PRUEBAS PARA VERIFICAR LA TRANSMISION Y RECEPCION DE DATOS.

Habíamos expuesto en la sección anterior que el archivo que transmitimos tiene el mismo nombre que el archivo de ingreso de datos con extensión ".DDE", éste lo crea el programa del PC. Luego, receptamos el archivo con igual nombre y extensión ".DDS". Para verificar la transmisión y recepción primero hicimos un programa sencillo que transmitirá un caracter del PC al COCOII y viceversa.

Los detalles de como se realiza la transmisión y recepción informamos en el literal 3.2.

A continuación hicimos un programa para IBM-PC que enviaba al COCOII el informe de conecciones por un archivo .DDE y recibimos el archivo .DDS, hasta que nos aseguramos que no teníamos

problemas en recepción y transmisión.

Resolvimos problemas como:

- Ir a un estado de espera, tanto la IBM-PC como la COCOII en el momento que el BUFFER de la PC se llenaba y teníamos que grabar estos datos en disco. Luego reanudar sin ningún problema desde el instante que el proceso se detenía.
- Interrumpir la transmisión y recepción tanto en IBM-PC como en COCOII en el momento que desconectamos el cable de comunicación y grabar los datos del BUFFER al disco, hasta donde el proceso se había interrumpido.
- Verificar el bit de paridad y la palabra de paridad para comprobar si la transmisión y recepción se estaba haciendo correctamente.

Una vez verificados estos detalles, adoptamos este programa como una subrutina del programa principal de manejo de datos y lo incluimos en la opción tercera del menú principal llamada "COMUNICACION". Al ejecutar esta opción nos advierte que verifiquemos si el cable de comunicación está correctamente conectado y si existe suficiente memoria en el disco para grabar los datos de salida; si no estamos seguros presionamos la tecla de escape y

volvemos al menú principal y si presionamos la tecla de ingreso se comienza a efectuar la comunicación. Al transmitir los datos se publica el número de datos que se transmitieron y cuando concluye la recepción se publica el número de datos que se recepta.

La verificación de esta opción se la efectuó con los ejemplos que se detallan en el capítulo 4.

3.3.3 PRUEBA PARA VERIFICAR EL CONTROL DE LOS INTEGRADORES DIGITALES EN TRS-80

Todas las verificaciones en el programa en la TRS-80 se realizan en el ZBUG y para mayor control lo hemos dividido el programa en estas partes:

- a. Recepción de datos de entrada.
- b. Convertir datos ASCII a binario
- c. Guardarlos en variables
- d. Definición de las conecciones
- e. Actualizar los valores dx y dy de cada integrador
- f. Procesar cada integrador
- g. Transmitir los datos

Los pasos e, f y g se repiten hasta que termina el proceso de simulación.

Las partes a, b, y c las verificamos efectuando una transmisión de datos, sabemos que estos se guardan a partir de la posición TABLA, por medio de ZBUG revisamos si efectivamente se han guardado en ese lugar. Luego, tenemos la conversión a binario, que se la realiza a partir de la posición VALOR, y clasificar cada dato para ingresarlo en la variable que le corresponda, también estas verificaciones se la realiza con el ZBUG ejecutando secuencialmente cada una de esta partes y observando en la memoria los valores para comprobar que sean correctos.

Aquí comienza el proceso de simulación pues cada uno de los registros de los integradores posee sus valores iniciales y ya se definieron las conecciones entre ellos.

Podemos verificar que el proceso se efectua sin ningun problema ejecutando un número determinado de iteraciones y accesar con el ZBUG a cada uno de los registros de los integradores para comprobar sus cambios correctos.

La transmisión de datos se la realiza según el número de veces que lo indique el IBM-PC y a que integradores deseamos conocer su información.

Verificamos que en la posición ASCII se encuentre los valores que se desean transmitir y que estos correspondan a los datos de los integradores de salida.

Para el proceso de cada integrador se utiliza el método de integración trapezoidal por ser éste el que conlleva menos error en los cálculos.

La verificación total del programa de datos se la realiza ejecutando el programa y comprobando que finalizado el proceso de simulación en la IBM-PC se ha recopilado los resultados correctos para cada variable programada como salida.

El programa del COCOII terminada la simulación se queda en un estado de espera, por si se desea efectuar una nueva simulación.

REFERENCIAS

1. WILLEN, DAVID C Y KRANTZ, JEFREY. 8088 Assembler Language Programming the IBM - PC. Howard W. Sams & Co. Inc. 1983. pp 172-173.
2. TES-80 Color Computer Technical Reference Manual. Radio Shack. Tandy Corporation. 1982. pp 33-34.
3. WILLEN, Op. Cit., pp 175-177.
4. TES-80 C.C.T.R.M., Op. Cit., pp 6-7.

CAPITULO IV

ANÁLISIS DE EJEMPLOS DE APLICACION DEL ANALIZADOR DIFERENCIAL DIGITAL.

LOS FILTROS DIGITALES

El término filtro digital se refiere al proceso computacional para el cual se muestrea una señal en secuencia de números digitales y se transforma en una segunda secuencia de números digitales que será la señal de salida.

Los filtros digitales aquí están físicamente compuestos de pequeños bloques digitales, tales como registros, sumadores, inversores (D.D.A.) y ellos tratan con señal cuantificadas (señales discretas). Su función es semejante a los filtros analógicos (señales continuas) y son muy buenas aproximaciones a ellos. Las ventajas de los filtros digitales es que son mucho mas pequeños precisos y estables en tiempo real.

ALGUNAS TÉCNICAS SOBRE EL DISEÑO DE FILTROS DIGITALES.

El procedimiento para diseñar filtros analógicos tales como filtros Butterworth y Chevyshev son tratados en forma somera en la sección siguiente.

4.2.1 FILTROS.1

Un filtro se lo puede definir como una red de dos puertos que tiene una señal de entrada $X(t)$ y una señal de salida $Y(t)$. Si el filtro esta compuesto de elementos lineales y t es una variable continua (tiempo) entonces X y Y están relacionados por una ecuación lineal ordinaria e integro-diferencial; podemos aplicar la transformada de Laplace, si no existen valores iniciales producidos por una energía inicial almacenada, de esta forma $Y(s) = H(s) X(s)$, donde $s = \sigma + j\omega$ frecuencia compleja.

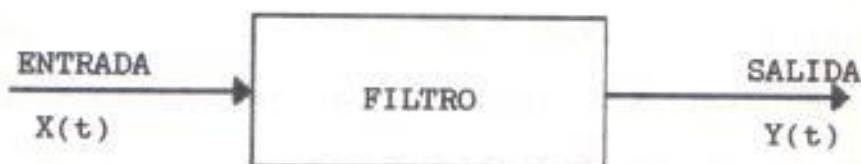


FIG. 4.1 REPRESENTACION SIMBOLICA DE UN FILTRO

Las cantidades $X(s)$ y $Y(s)$ son respectivamente las transformadas de Laplace de $Y(t)$ y $X(t)$; y, $H(s)$, la función de transferencia de la red. Si $s=j\omega$ (ω medida en rad/seg) la función de red es compleja y puede ser escrita en la forma:

$$H(j\omega) = |H(j\omega)| e^{j\phi(\omega)}$$

donde $|H(j\omega)|$ es la amplitud o magnitud y $\phi(\omega)$ es la fase. Las respuestas de amplitud y fase

son respectivamente representadas en gráfico $|H(j\omega)|$ y $\phi(\omega)$ versus ω ; y pueden ser usadas para caracterizar el filtro.

Vamos a estudiar los filtros pasa bajo pues son de especial interés en el campo de las comunicaciones, se define al filtro pasa bajo como un selector de frecuencias que pasa señales cuyas frecuencias están en el rango o banda de 0 a frecuencia de corte ω_c y bloquea o atenúa señales cuyas frecuencias son mayores que ω_c . El ancho de banda está definido como $B = \omega_c$.

Un filtro ideal es uno el cual tiene una respuesta lineal en su banda y detiene frecuencias fuera de su banda; cero pérdida en su ancho de banda y su magnitud es 0 en el resto de frecuencia.

Por ejemplo, el filtro pasa bajo ideal se define como:

$$|H(j\omega)| = A \quad 0 < \omega < \omega_c$$

$$|H(j\omega)| = 0 \quad \omega > \omega_c$$



FIG. 4.2 RESPUESTA DE MAGNITUD DE UN FILTRO IDEAL

4.2.2 EL FILTRO PASA BAJO EN GENERAL

En este filtro se normaliza la frecuencia de corte a $\omega_c=1$ rad/seg y si se quiere que la frecuencia de corte sea otra se podría desnormalizar utilizando escalamiento.

Así mismo consideramos la amplificación del filtro igual a la unidad, entonces tenemos una función general de los filtros pasa bajo.

$$|H(j\omega)|^2 = \frac{1}{1 + f(\omega^2)}$$

4.2.4 donde

$$f(\omega^2) \gg 1$$

$$\omega > 1$$

$$0 \leq f(\omega^2) \ll 1$$

$$0 \leq \omega < 1$$

Generalmente, $f(\omega^2)$ es un polinomio de grado $2n$ en ω y si obtenemos $H(s)$ el polinomio del denominador, será de grado n pues

$$H(s) = \frac{k}{Q(s)}$$

Esta función tiene n polos y ningún cero.

4.2.3 FILTRO PASA BAJO BUTTERWORTH.2

Una función adecuada es $f(\omega^2) = \omega^{2n}$ y esta la utiliza el filtro Butterworth donde n es el grado del filtro.

Así,

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \omega^{2n}}} \quad n = 1, 2, 3, \dots$$

para la frecuencia de corte $\omega_c = 1$

$$|H(j1)| = \frac{1}{\sqrt{2}}$$

$$\text{para } \omega \gg 1 \quad ; \quad |H(j\omega)| \approx \frac{1}{\omega^n}$$

Así la pérdida en decibelios será:

$$\alpha_{db}(\omega) \approx 20 \log \omega^n = 20 n \log \omega$$

4.2.4 FUNCION DE TRANSFERENCIA DEL BUTTERWORTH

Habiamos dicho que:

$$H(s) H(-s) = \frac{1}{1 + (-s^2)^n}$$

$$H(s) = \frac{1}{Q(s)}$$

por lo que $Q(s) Q(-s) = 1 + (-s^2)^n$

Los n polos de $H(s)$ los obtenemos de las raíces de $Q(s)$ así

$$1 + (-s^2)^n = 0$$

$$s^{2n} = -1 = e^{j(2k-1)\pi} \quad ; \quad \text{para } k=0, 1, 2, \dots, 2n-1$$

las raíces $s_k = \sigma_k + j\omega_k$ que la satisfacen

$$S_k^{2n} = e^{j(2k-1)\pi + jn\pi}$$

asi,

$$S_k = \sigma_k + jw_k = e^{j(2k+n-1)\pi/2n}$$

la parte real

$$\sigma_k = \cos \left[\frac{(2k+n-1)\pi}{2n} \right] = - \operatorname{sen} \left[\frac{(2k-1)\pi}{2n} \right]$$

$$w_k = \operatorname{sen} \left[\frac{(2k+n-1)\pi}{2n} \right] = \cos \left[\frac{(2k-1)\pi}{2n} \right]$$

Consideramos a σ_k como negativo.

4.2.5 FILTRO PASA BAJO CHEBYSHEV³

Los filtros Butterworth no tienen una buena característica en la vecindad de la frecuencia de corte $w = 1$, por lo cual $f(w^2)$ la forzamos a que tenga varios mínimos (ceros) en el ancho de banda, para que $|H(jw)|$ tenga un máximo valor de uno en un cierto número de puntos.

De aquí surge el filtro Pasa - bajo Chebyshev de orden n .

$$|H(jw)| = \frac{1}{\sqrt{1 + \epsilon^2 C_n^2(w)}} \quad n = 1, 2, 3, \dots$$

donde ϵ es una constante y

$$C_n(w) = \cos(n \cos^{-1} w)$$

por medio de diferencias y conociendo que:

$$C_0(w) = \cos 0 = 1$$

$$C_1(w) = \cos (\cos^{-1} w) = w$$

deducimos que

$$C_{n+1}(w) = 2w C_n(w) - C_{n-1}(w)$$

De aquí generamos todos los polinomios restantes de orden n .

El rizado en el ancho de banda se lo calcula

$$RW = 1 - \frac{1}{\sqrt{1 + \epsilon^2}}$$

y su pérdida en decibelios

$$RW_{db} = -20 \log_{10} \left[\frac{1}{\sqrt{1 + \epsilon^2}} \right]$$

$$= 10 \log_{10} [1 + \epsilon^2]$$

Para w suficientemente grande

$$\epsilon^2 |C_n(w)| \gg 1$$

Aproximadamente la amplitud será

$$|H(jw)| \approx \left[\frac{1}{\epsilon C_n(w)} \right]$$

Así, la pérdida es dada por

$$\alpha(w) \approx 20 \log \epsilon + 20 \log C_n(w)$$

Para w grande decimos que $C_n(w)$ es igual a $2^{n-1} w^n$ reemplazo en la expresión anterior

$$\alpha(w) \approx 20 \log \epsilon + 6(n-1) + 20 n \log w$$

4.2.6 FUNCION DE TRANSFERENCIA DEL FILTRO CHEBYSHEV

Por igual análisis del anterior filtro decimos

$$Q(s) Q(-s) = 1 + \epsilon^2 C_n^2(w) \Big|_{w^2 = -s^2}$$

equivalentemente decimos que $w = \pm js$ y los polos de $H(s)$ deben estar en la parte izquierda del plano complejo.

$$1 + \epsilon^2 C_n^2(-js) = 0$$

ó

$$C_n(-js) = \pm j/\epsilon$$

Si decimos que

$$-js = \cos(u + jv)$$

por lo cual expandiendo

$$-js = \cos u \cosh v - j \sin u \sinh v$$

debido a que $\cos jv = \cosh v$ y $\sin jv = j \sinh v$ volviendo a la ecuación anterior

$$C_n(-js) = \cos n(u + jv)$$

$$= \cos nu \cosh nv - j \sin nu \sinh nv = \pm j/\epsilon$$

igualando ambas partes

$$\cos nu \cosh nv = 0$$

$$- \operatorname{sen} nu \operatorname{senh} nv = \pm 1/\epsilon$$

para valores reales de v , $\cosh nv \neq 0$, tenemos que $\cos nu = 0$ y así

$$u = u_k = \frac{(2k - 1)\pi}{2n} \quad ; \quad k = 1, 2, \dots, 2n$$

entonces, $\operatorname{sen} u_k = \pm 1$ y la segunda se satisface cuando

$$\operatorname{senh} nv = 1/\epsilon$$

ó

$$v = \frac{1}{n} \operatorname{senh}^{-1} \frac{1}{\epsilon}$$

si escribimos a los polos como

$$s_k = j \cos u \cosh v + \operatorname{sen} u \operatorname{senh} v$$

donde u y v están dados por las expresiones anteriores

como

$$s_k = \sigma_k + j\omega_k$$

$$\sigma_k = - \operatorname{sen} \frac{(2k - 1)\pi}{2n} \operatorname{senh} v$$

$$\omega_k = \cos \frac{(2k - 1)\pi}{2n} \cosh v \quad ; \quad k = 1, 2, 3, \dots, n$$

4.3 EJEMPLOS SOBRE DISEÑO DE FILTROS DIGITALES.

Los ejemplos sobre diseño de filtros digitales tales como filtros Butterworth y Chebyshev aplicando técnicas DDA son tratados a continuación.

4.3.1 EJEMPLO 14

Diseñar un filtro pasa bajo utilizando la función de transferencia Butterworth.

Para tener una característica de magnitud tal que a una frecuencia tres veces la frecuencia de corte, la magnitud es menor en 25 db de su valor en la frecuencia cero.

Planteamos:

$$25 \approx 20 n \log w$$

$$25 \approx 20 n \log 3$$

$$n \approx 2.61$$

Entonces escogemos n para que cumpla la característica como $n = 3$ la función de transferencia del filtro Butterworth de tercer orden es:

$$G(s) = \frac{1}{s^3 + 2s^2 + 2s + 1} = \frac{Y(s)}{X(s)}$$

transformando una ecuación diferencial tenemos:

$$(s^3 + 2s^2 + 2s + 1) Y(s) = X(s)$$

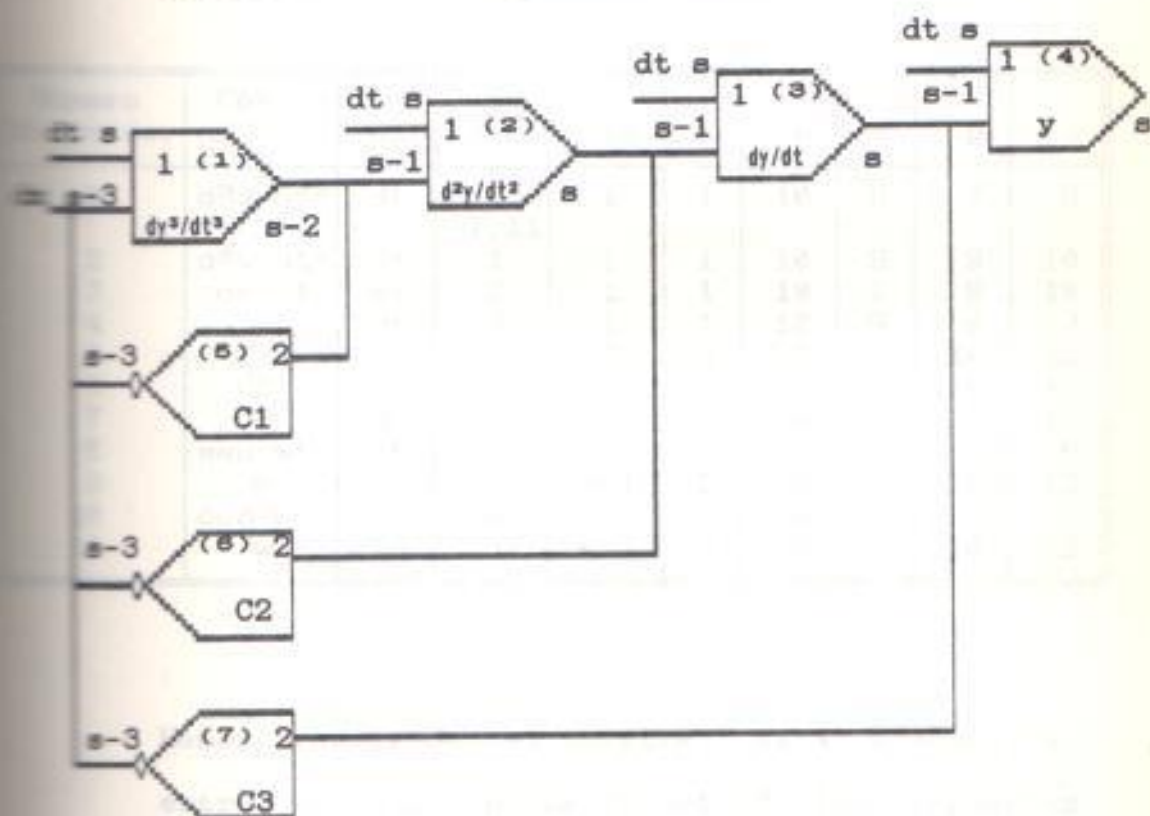
$$\frac{d^3y}{dt^3} + 2 \frac{d^2y}{dt^2} + 2 \frac{dy}{dt} + y = x$$

si obtenemos el diferencial de esta ecuación,
tenemos

$$d \left[\frac{d^3y}{dt^3} \right] + 2 d \left[\frac{d^2y}{dt^2} \right] + 2 d \left[\frac{dy}{dt} \right] + dy = dx$$

$$d \left[\frac{d^3y}{dt^3} \right] = - 2 d \left[\frac{d^2y}{dt^2} \right] - 2 d \left[\frac{dy}{dt} \right] - dy + dx$$

Las conecciones de los integradores están
realizadas en la siguiente figura:



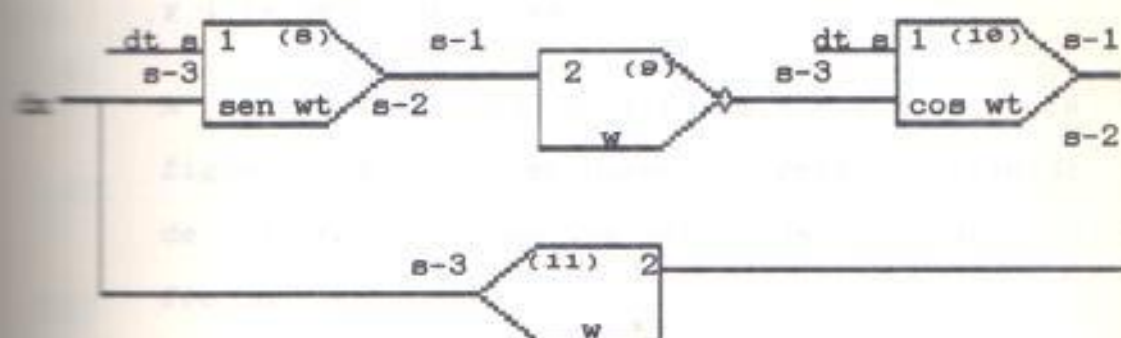


FIG. 4.3 DIAGRAMA DE SIMULACION DE UN FILTRO BUTTERWORTH TERCER ORDEN

Este es un informe de las conexiones de los integradores que presentamos para ingresarlos en la máquina D.D.A. y obtener la respuesta a la frecuencia de corte $\omega_c = 1$

Número Integrador	Contenido Y	dx	dy	Y_m	n	s	z	p	L
1	d^3y/dt^3	dt	-5,-6 -7,11	1	1	10	9	7	8
2	d^2y/dt^2	dt	1	1	1	10	9	9	10
3	dy/dt	dt	2	1	1	10	9	9	10
4	y	dt	3	1	1	10	9	9	10
5	C1	1	-	2	2	9	7	10	12
6	C2	2	-	2	2	9	7	10	12
7	C3	3	-	1	2	9	7	10	12
8	sen wt	dt	11	1	1	10	9	7	8
9	w	8	-	w=1	2	9	7	10	12
10	cos wt	dt	-9	1	1	10	9	7	8
11	w	10	-	w=1	2	9	7	10	12

Hemos realizado el muestreo para frecuencias entre 0.1 a 2 y hemos encontrado las respuestas para los integradores 8 como la señal de entrada

y 4 la señal de salida.

A continuación en las figuras 4.4.a hasta la figura 4.4.m, presentamos el gráfico obtenido de la entrada y salida para cada una de las frecuencias muestreadas.

Para observar el tiempo de estabilización en la respuesta introducimos nosotros en lugar de la señal sinusoidal la entrada escalón $X(t) = u(t)$.

Así observamos en la figura 4.5 la señal de respuesta del filtro y su tiempo de estabilización que es de $t=13.5$ seg.

Además para el análisis del filtro lo completamos con la simulación de la respuesta de magnitud del filtro en función de frecuencia y así partiendo de la ecuación:

$$M(w) = \frac{1}{\sqrt{1 + w^6}}$$

si $u = w^3$ y $v = 1 + u^2$

Tenemos que: $M = 1/\sqrt{v}$

FIG. 4.4.a

Simulando estas funciones tenemos que la interconexión de los integradores sera:

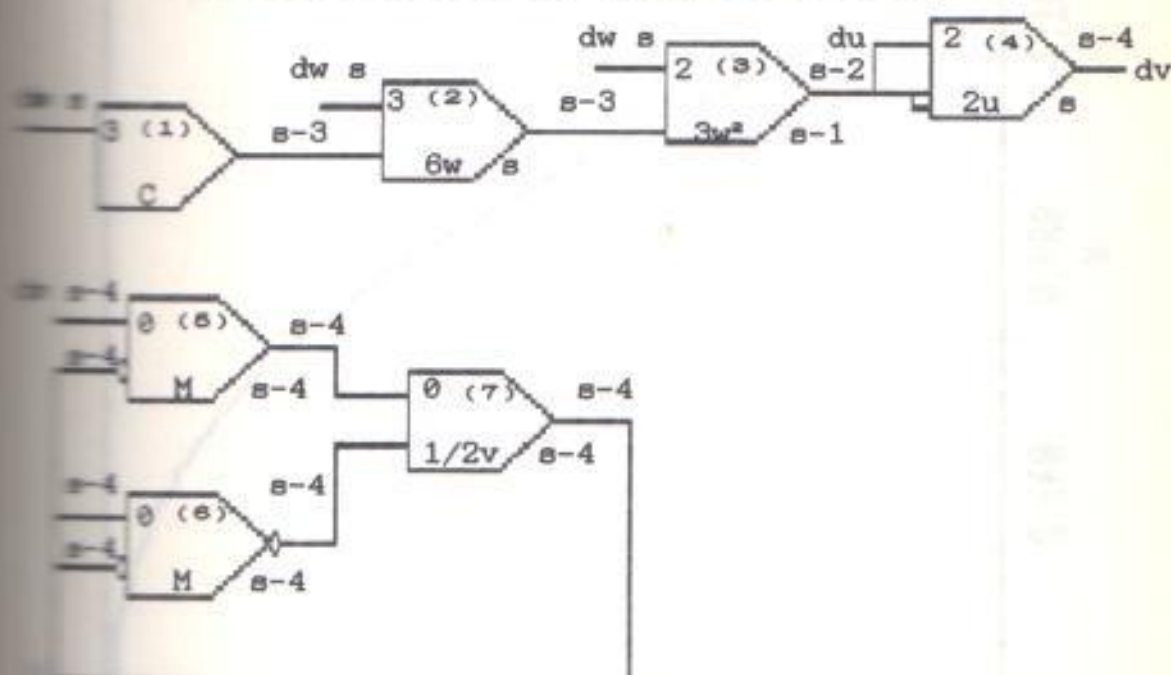


FIG. 4.6 DIAGRAMA DE LA RESPUESTA DE MAGNITUD DEL FILTRO DE BUTTERWORTH DE TERCER ODEN.

La respuesta de magnitud la obtenemos del integrador 5.

El informe de las conecciones de los integradores y su escalamiento lo hacemos a continuación:

Numero Integrador	Contenido Y	dx	dy	Y_m	n	s	z	p	L
1	C	dw	-	6	3	13	10	10	13
2	6w	dw	1	8	3	13	10	10	13
3	3w ²	dw	2	4	2	13	11	10	12
4	2u	3	3.3	4	2	11	9	11	13
5	M	4	-7	1	0	9	9	9	9
6	M	7	-7	1	0	9	9	9	9
7	1/2v	5	-6	1	0	9	9	9	9

El gráfico de la respuesta de magnitud en w de 0 a 1.1 se observa en la figura 4.7

4.3.2 EJEMPLO 2 *

Diseñe un filtro pasa bajo utilizando la característica Chebyshev tal que:

- a) El rizado no exceda el 7.8% de su valor máximo 1.
- b) La respuesta de magnitud es menor que 50 db para w de 4 veces la frecuencia de corte que es 1.

De acuerdo a la primera especificación tenemos:

$$RW = 1 - \frac{1}{\sqrt{1+\epsilon^2}} = 0.078$$

de aquí $\epsilon^2 = 0.176$ y $\epsilon = 0.420$

De la segunda especificación tenemos

$$\alpha(w) \approx 20 \log \epsilon + 6(n-1) + 20n \log w$$

$$50 \approx 20 \log (0.42) + 6(n-1) + 20n \log 4$$

$$n \approx 3.45$$

por lo que escogemos a $n = 4$

Para obtener la función de transferencia hallamos los polos del filtro Chebyshev de cuarto orden y $\epsilon = 0.42$, así:

$$v = \frac{1}{n} \sinh^{-1} \left[\frac{1}{\epsilon} \right]$$

$$v = \frac{1}{4} \sinh^{-1} \left[\frac{1}{0.42} \right] = 0.4$$

y

$$\sigma_k = - \sin \frac{(2k-1)\pi}{2n} \sinh v ; \quad k=1,2,3,4$$

$$\sigma_1 = - \sin \frac{\pi}{8} \sinh (0.4) = - 0.157$$

$$\sigma_2 = - \sin \frac{3\pi}{8} \sinh (0.4) = - 0.379$$

$$\sigma_3 = - \sin \frac{5\pi}{8} \sinh (0.4) = - 0.379$$

$$\sigma_4 = - \sin \frac{7\pi}{8} \sinh (0.4) = - 0.157$$

y

$$w_k = \cos \frac{(2k-1)\pi}{2n} \cosh v ; \quad k = 1,2,3,4$$

$$w_1 = \cos \frac{\pi}{8} \cosh (0.4) = 0.999$$

$$w_2 = \cos \frac{3\pi}{8} \cosh (0.4) = 0.414$$

$$w_3 = \cos \frac{5\pi}{8} \cosh(0.4) = -0.414$$

$$w_4 = \cos \frac{7\pi}{8} \cosh(0.4) = -0.999$$

por lo que los cuatro polos son:

$$s_1 = -0.157 + j 0.999$$

$$s_2 = -0.379 + j 0.414$$

$$s_3 = -0.379 - j 0.414$$

$$s_4 = -0.157 - j 0.999$$

La función de transferencia es:

$$G(s) = \frac{k}{(s-s_1)(s-s_2)(s-s_3)(s-s_4)}$$

$$G(s) = \frac{k}{s^4 + 1.072 s^3 + 1.576 s^2 + 0.874 s + 0.322}$$

para calcular k sabemos que:

$$G(j0) = \frac{k}{0.322} = 0.922 \quad k = 0.297$$

entonces:

$$\frac{Y(s)}{X(s)} = \frac{0.297}{s^4 + 1.072 s^3 + 1.576 s^2 + 0.874 s + 0.322}$$

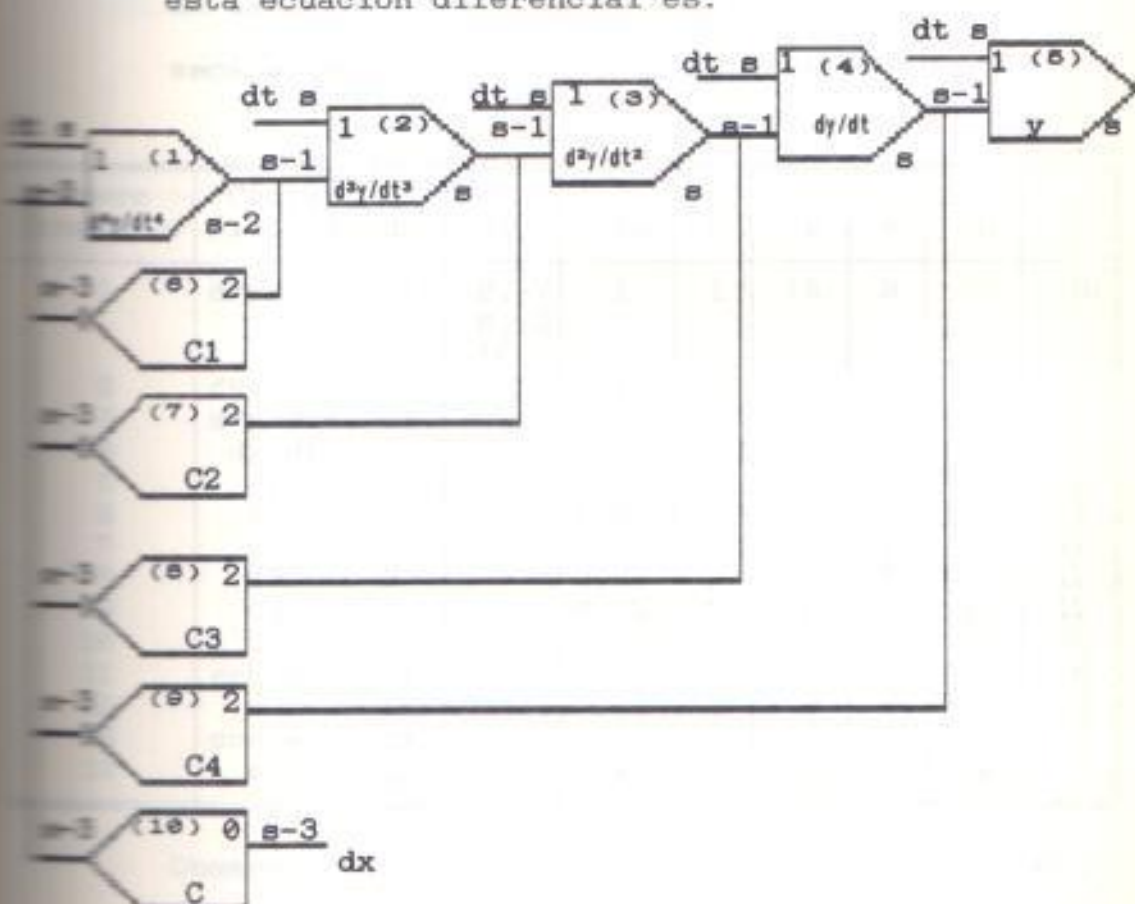
Pasando a la ecuación diferencial en tiempo tenemos:

$$\frac{d^4 y}{dt^4} + 1.072 \frac{d^3 y}{dt^3} + 1.576 \frac{d^2 y}{dt^2} + 0.874 \frac{dy}{dt} + 0.322 y = 0.297 x$$

diferenciando y despejando la mas alta derivada:

$$s \left[\frac{d^4 y}{dt^4} \right] = -1.072 s \left[\frac{d^3 y}{dt^3} \right] - 1.576 s \left[\frac{d^2 y}{dt^2} \right] - 0.874 s \left[\frac{dy}{dt} \right] - 0.322 dy + 0.297 dx$$

La conección de los integradores que simulan esta ecuación diferencial es:



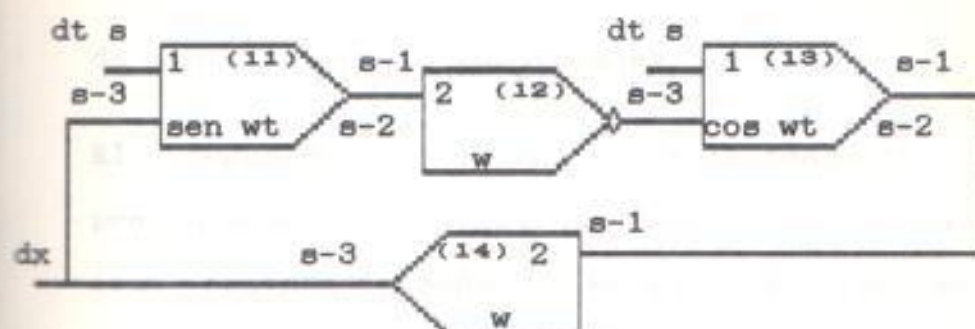


FIG. 4.8 DIAGRAMA DE SIMULACION DEL FILTRO CHEBYSHEV DE CUARTO ORDEN.

Para esta simulación variamos en los integradores 12 y 14 los w entre 0.1 y 2 y obtenemos las diversas respuestas en el integrador 5.

La tabla de conexiones de los integradores y su escalamiento se detalla a continuación:

Número Integrador	Contenido Y	dx	dy	Y _m	n	s	z	p	L
1	d ⁴ y/dt ⁴	dt	-6,-7 -8,-9 10	1	1	10	9	7	8
2	d ³ y/dt ³	dt	1	1	1	10	9	9	10
3	d ² y/dt ²	dt	2	1	1	10	9	9	10
4	dy/dt	dt	3	1	1	10	9	9	10
5	y	dt	4	1	1	10	9	9	10
6	C1	1	-	1.072	2	9	7	10	12
7	C2	2	-	1.576	2	9	7	10	12
8	C3	3	-	0.874	2	9	7	10	12
9	C4	4	-	0.322	2	9	7	10	12
10	C	14	-	0.297	0	7	7	10	10
11	sen wt	dt	14	1	1	10	9	7	8
12	w	11	-	w	2	9	7	10	12
13	cos wt	dt	-12	1	1	10	9	7	8
14	w	13	-	w	2	9	7	10	12

Observaremos los gráficos 4.10.a hasta 4.10.q obtenidos de la entrada y salida para cada una

de las frecuencias muestreadas.

El tiempo de estabilización de la respuesta lo presentamos en el gráfico 4.11 al introducir como entrada la señal de escalón y observemos que es $t_s = 24.5$ seg

Tenemos que la interconexión de los integradores es:

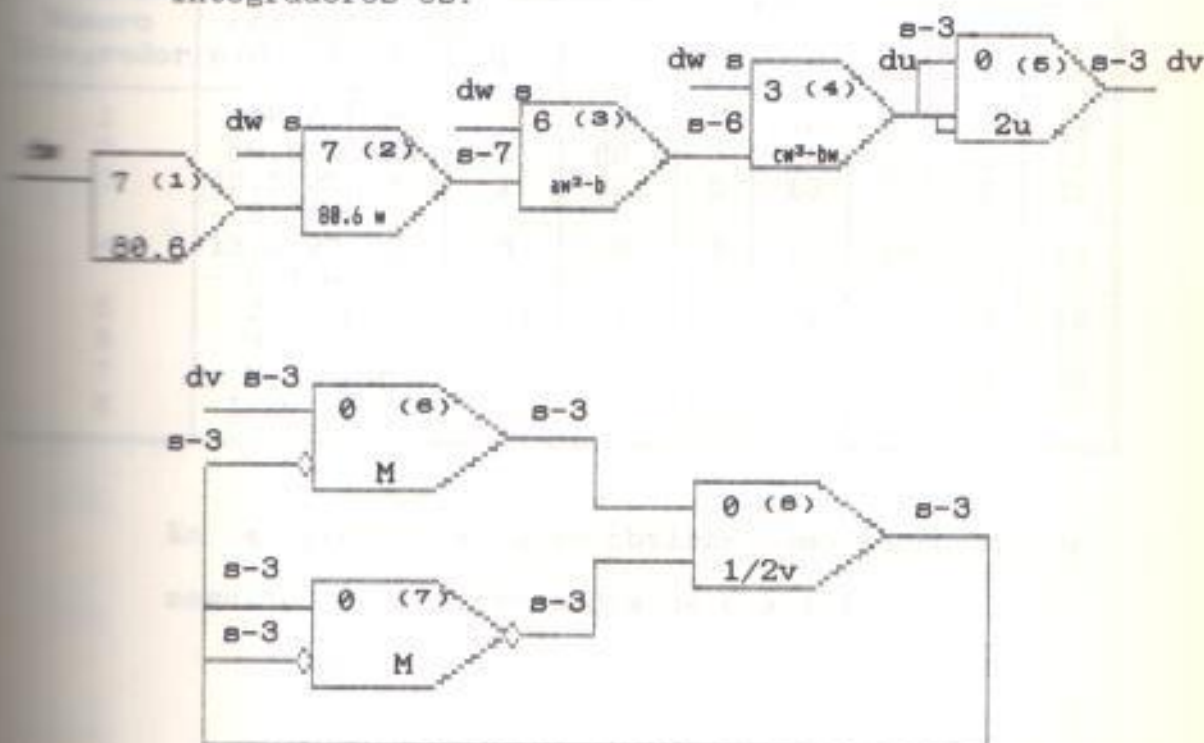


FIG. 4.9 DIAGRAMA DE SIMULACION DE LA RESPUESTA DE MAGNITUD DEL FILTRO CHEBYSHEV DE CUARTO ORDEN

La simulación de la respuesta de magnitud del filtro en función de frecuencia la realizamos de la ecuación:

$$M(\omega) = \frac{1}{\sqrt{1 + (0.42)^2 (8\omega^4 - 8\omega^2 + 1)^2}}$$

$$M(w) = \frac{1}{\sqrt{1 + (3.36 w^4 - 3.36 w^2 + 0.42)^2}}$$

Si $u = 3.36 w^4 - 3.36 w^2 + 0.42$ y $v = 1 + u^2$

La tabla de informe de la conexión de los integradores y escalamiento es:

Número Integrador	Contenido Y	dx	dy	Y _m	n	s	z	p	L
1	80.6	dw	--	80.6	7	13	6	7	14
2	80.6 w	dw	1	88.6	7	13	6	6	13
3	40.3 w ² - 6.70	dw	2	42	6	13	7	6	12
4	13.4 w ³ - 6.7 w	dw	3	8	3	13	10	7	10
5	2u	4	4.4	1	0	10	10	10	10
6	M	5	-8	1	0	10	10	10	10
7	M	8	-8	1	0	10	10	10	10
8	1/2v	6	-7	1	0	10	10	10	10

En el gráfico 4.12 se obtiene como respuesta de magnitud en la frecuencia de 0 a 1.1.



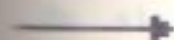
FIG. 4.12. Respuesta en magnitud.

SIMULACION DE RECURSOS DEL MEDIO AMBIENTE Y EL USO DE LAZOS ECONOMICOS (ECONUSE)

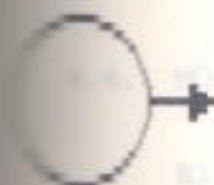
Esta información la obtuve de la obra "Computer
Minimodels and Simulation Exercises" de Howard T.
Odum. El basa sus ideas sobre la relación entre
economía y ecología. Odum sostiene "que el
desarrollo económico no debe conducirnos al deterioro
de la naturaleza y que una economía sana se sustenta
siempre en una ecología saludable".

Sus modelos pueden ser dibujados como un diagrama de
lenguaje de energía, usando los símbolos de las
siguientes figuras.⁶

ENLACE DE ENERGIA.- Un flujo de energía, a menudo con
un flujo de materiales.

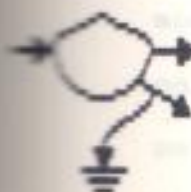


FUENTE DE ENERGIA.- Energía la que es acompañada por
cada uno de los recursos usados por los
ecosistemas tales como el sol, vientos,
lluvias, etc.



ALMACENAJE.- Un lugar donde la energía es almacenada.

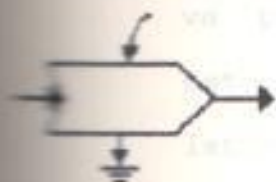
Ejemplos son los recursos tales como
biomasa forestales, materia orgánica,
suelos, etc.



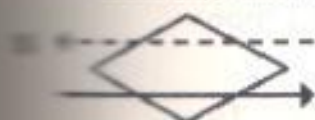
DESHECHOS.- Energía que es disipada.



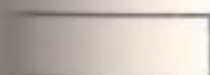
INTERACCION.- Procesos los cuales combinan diferentes tipos de flujos de energía o flujo de materia.



TRANSACCION.- Negocios, intercambio de dinero por energía, materiales, servicio, etc.



CAJA.- Símbolo miscelaneo para subsistemas.



Los diagramas son tres esencialmente: Fuentes externas, partes interiores (subsistemas) y camino de conexión (enlaces). Los diagramas muestran como las partes del modelo son relacionados.

En el ejemplo que detallo contiene un modelo que ilustra un sistema económico y como es integrado con el sistema ecológico.

4.4.1 MODELO ECONUSE⁷

El modelo ECONUSE es un ejemplo de un producto del medio ambiente que es usado en el proceso económico. El producto (Q), que en nuestro país podría ser típicamente pescado, camarón, madera, etc; éste es producido por la energía renovable del sol, lluvia y viento. Este producto a medida que se va acumulando, se lo

va procesando para venderlo, ya que necesita este dinero para comprar y sostener equipos y labor, además de esto nos serviría para comprar mercancías y servicios que se necesitan para continuar con el proceso.

Si llamamos I_e a la fuente de energía renovable externa y R la parte de I_e la cual es capaz de poder ser utilizada, planteamos la ecuación:

$$I_e = R + K_e * R * Q$$

Si cambiamos algebraicamente

$$R = \frac{I_e}{1 + K_e}$$

El incremento del producto Q , es decir el crecimiento de la biomasa es proporcional a la energía del sol disponible (R) y la cantidad de biomasa que ya se encuentra crecida (Q) $K_1 * R * Q$. La biomasa muerta y descompuesta es una proporción (K_2) de la biomasa almacenada (Q): $K_2 * Q$.

Además, tenemos que el rendimiento del producto listo para la venta (E) depende de ambos, de la cantidad del producto original (Q) y la cantidad del equipo procesador y la labor humana (A), por esta causa tenemos un

decremento del producto (Q): $K_3 * A * Q$.

La ecuación para el incremento del producto Q es:

$$\frac{dQ}{dT} = K_1 * R * Q - K_2 * Q - K_3 * A * Q$$

La cantidad de dinero que se acumula (M) es el dinero que llega de la venta del producto ($PE * E$), PE es el precio del producto, menos el dinero pagado por compra de mercancías y servicios ($K_5 * M$). Por lo que:

$$\frac{dM}{dT} = PE * E - K_5 * M$$

El incremento del equipo activo se debe principalmente a la compra de ellos por el dinero acumulado:

$K_5 * M / PG$ donde PG el precio que pagamos por ellos, y existe un decremento por la depreciación del equipo: $K_6 * A$ y por la proporción del equipo perdida por el rendimiento del producto: $K_7 * Q * A$, por lo que:

$$\frac{dA}{dT} = K_5 * M / PG - K_6 * A - K_7 * Q * A$$

Cada una de estas variables Q , M , A , son integradas en función de unidades de tiempo T , por medio de los acumuladores

$$Q^* = Q + \frac{dQ}{dT} * dT$$

$$A^* = A + \frac{dA}{dT} * dT$$

$$M^* = M + \frac{dM}{dT} * dT$$

El autor, experimentalmente ha encontrado las diferentes constantes, así:

$$K_1 = 0.02 \quad K_2 = 0.05 \quad K_3 = 0.00166$$

$$K_4 = 0.002 \quad K_5 = 0.1 \quad K_6 = 0.03$$

$$K_7 = 0.0001$$

Además asigna a los precios del producto y de los activos como:

$$PE = 1 \quad \text{y} \quad PG = 2$$

La energía renovable $I_e = 100$ y la proporción de la energía que se utiliza es $K_e = 0.38$.

La simulación la realiza en el dominio de unidades de tiempo $[0, 300]$.

La simulación se inicia con una baja cantidad de producto (Q), equipo activo (A) y dinero (M).

Si esto, es una operación de una piscina de camarones, primero la cantidad de camarones (Q), se cultiva ante el desarrollo de activos (A). Entonces los activos y venta se incrementan y se transforman en mas dinero (M) capaz de comprar mas mercancías y dar mas servicios, transportes y carreteras para construir activos económicos (A). Mas tarde un balance de producción y uso es logrado en un mas bajo incremento de uso basado en la limitación natural de la reproducción del camarón.

El autor ilustra su simulación con el siguiente gráfico de flujo de energía (EMERGY) inventado por él:

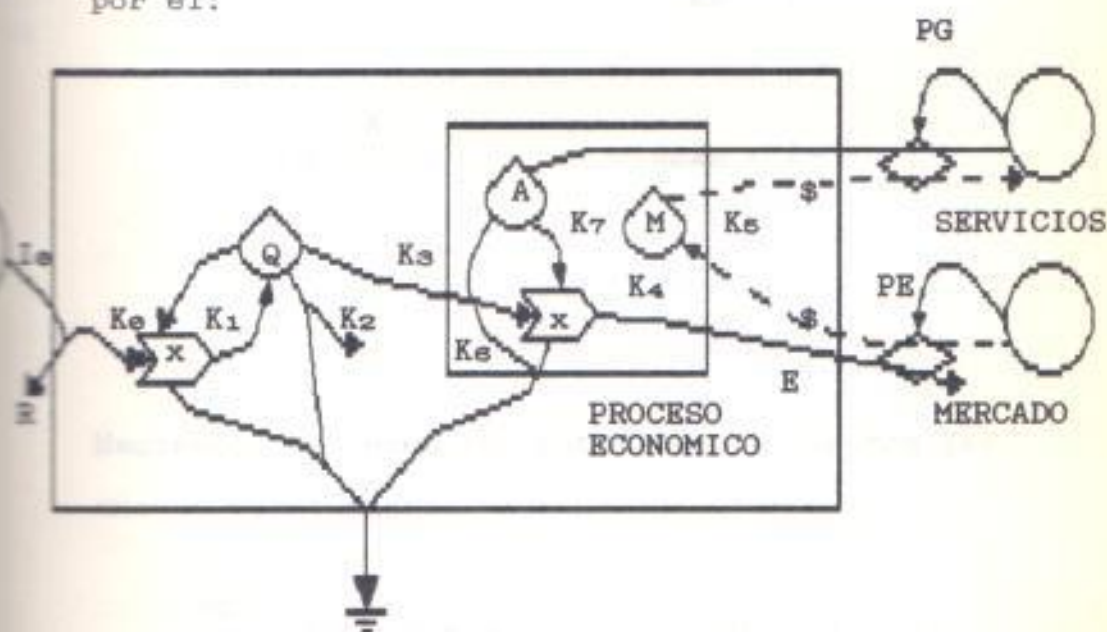


FIG. 4.13 DIAGRAMA DE SIMULACION DEL MODELO ECONUSE

Quiero adoptar este ejemplo para hacer una simulación digital en el D.D.A. Ante todo tenemos que actualizar los valores de las constantes, haciendo un escalamiento de las variables R, Q, M, A, dQ/dT , dM/dT y dA/dt , conociendo de antemano que en cada uno de los integradores digitales tenemos máximo 16 bits donde el mas significativo es el del signo.

Por lo que experimentalmente escogí a las variables digitales, así:

$$t = \frac{T}{10} \quad t : [0,30]$$

$$q = \frac{Q}{5} \quad q_{\max} < 2^4$$

$$a = \frac{A}{4} \quad a_{\max} < 2^4$$

$$m = \frac{M}{4} \quad m_{\max} < 2^4$$

Haciendo el reemplazo matemático tendremos las nuevas constantes:

$$\frac{dq}{dt} = C_1 * r * q - C_2 * q - C_3 * a * q$$

$$\frac{dm}{dt} = C_7 * a * q - C_8 * m$$

$$\frac{da}{dt} = C_4 * m - C_5 * a - C_6 * a * q$$

$$r = \frac{1}{0.4 + C_9 * q}$$

$$e = C_7 * a * q$$

Escribiendo las ecuaciones en forma diferencial para ser simuladas en el DDA tenemos:

$$\left[\frac{dq}{dt} \right] = C_1 r dq + C_1 q dr - C_2 dq - C_3 a dq - C_3 q da$$

$$\left[\frac{dm}{dt} \right] = C_7 a dq + C_7 q da - C_8 dm$$

$$\left[\frac{da}{dt} \right] = C_4 dm - C_5 da - C_6 a dq - C_6 q da$$

$$\text{para } r = \frac{1}{u},$$

$$\text{donde } u = 0.4 + C_9 q$$

$$du = C_9 dq$$

$$\text{y } de = C_7 a dq + C_7 q da$$

Con estas ecuaciones dibujamos el diagrama de conecciones de los integradores:

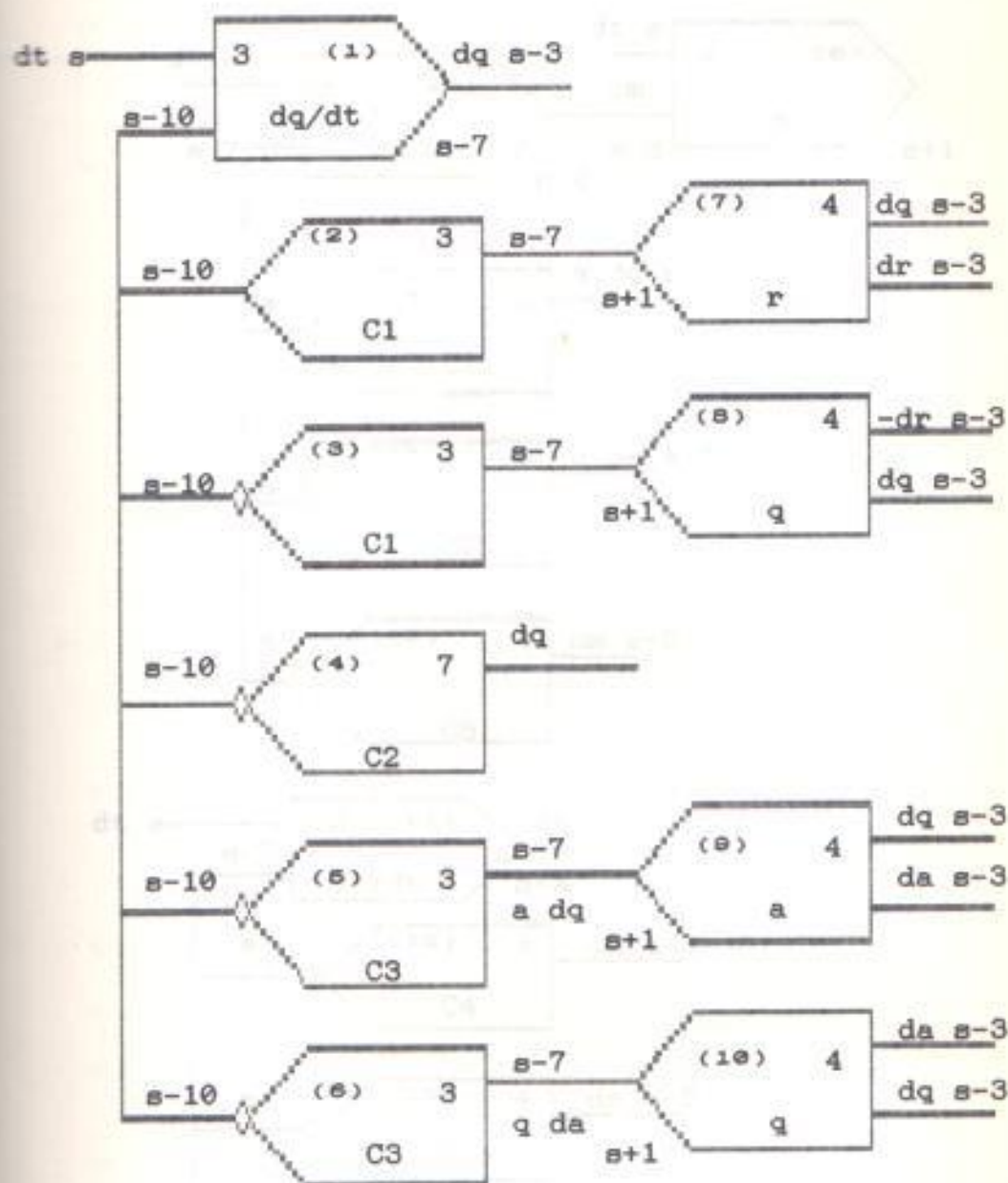


FIG. 4.14 DIAGRAMA DE CONECCION DE INTEGRADORES DIGITALES DE LA SIMULACION DEL MODELO DE ECONUSE (PRIMERA PARTE)

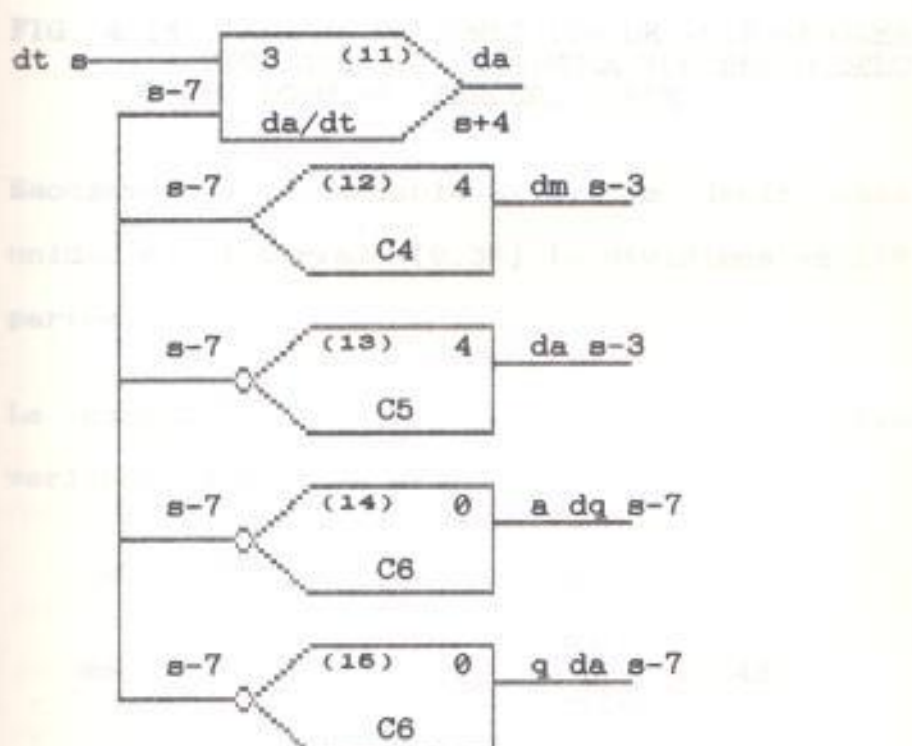
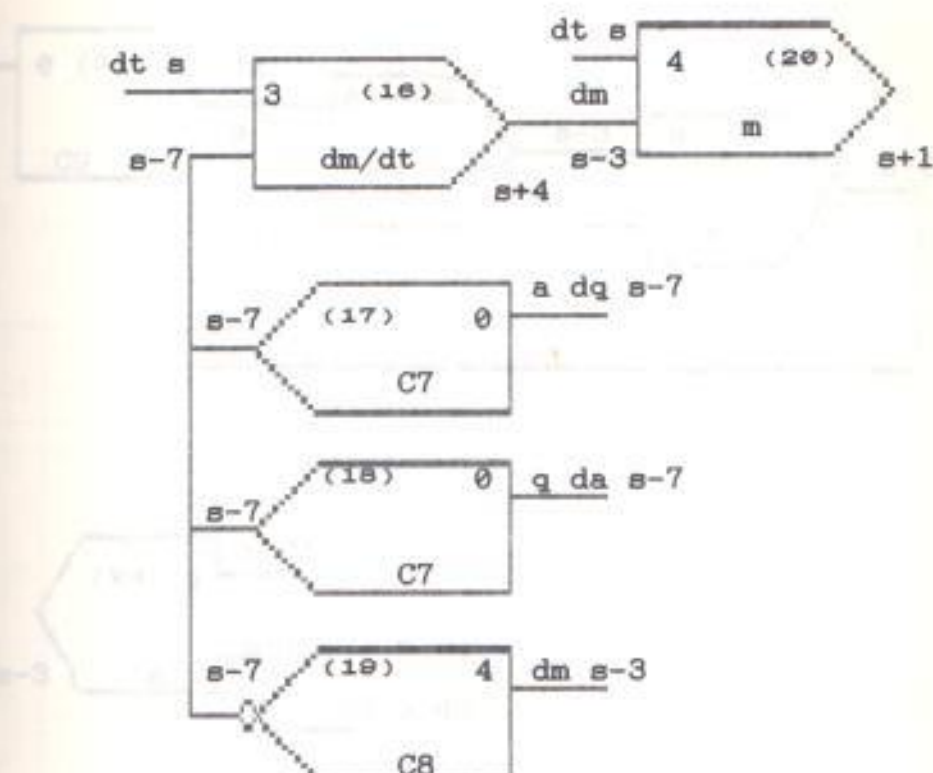


FIG. 4.14 DIAGRAMA DE CONECCION DE INTEGRADORES DIGITALES DE LA SIMULACION DEL MODELO DE ECOMUSE (SEGUNDA PARTE)

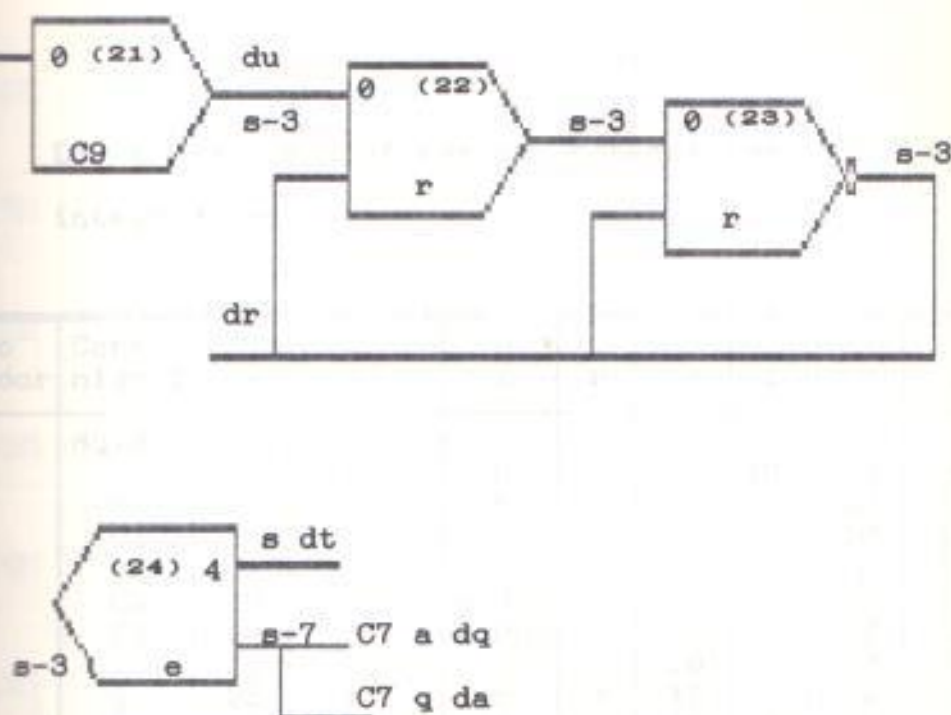


FIG. 4.14 DIAGRAMA DE CONECCION DE INTEGRADORES DIGITALES DE LA SIMULACION DEL MODELO DE ECONUSE (TERCERA PARTE)

Escogemos a la variable $S=13$, es decir cada unidad del intervalo $[0,30]$ lo dividimos en 2^{13} partes.

La condición inicial para cada una de las variables digitales es:

$$q_0 = 1$$

$$a_0 = 0.75$$

$$m_0 = 0.75$$

$$\left. \frac{dq}{dt} \right|_0 = 6.346$$

$$\left. \frac{da}{dt} \right|_0 = 0.146$$

$$\left. \frac{dm}{dt} \right|_0 = -0.675$$

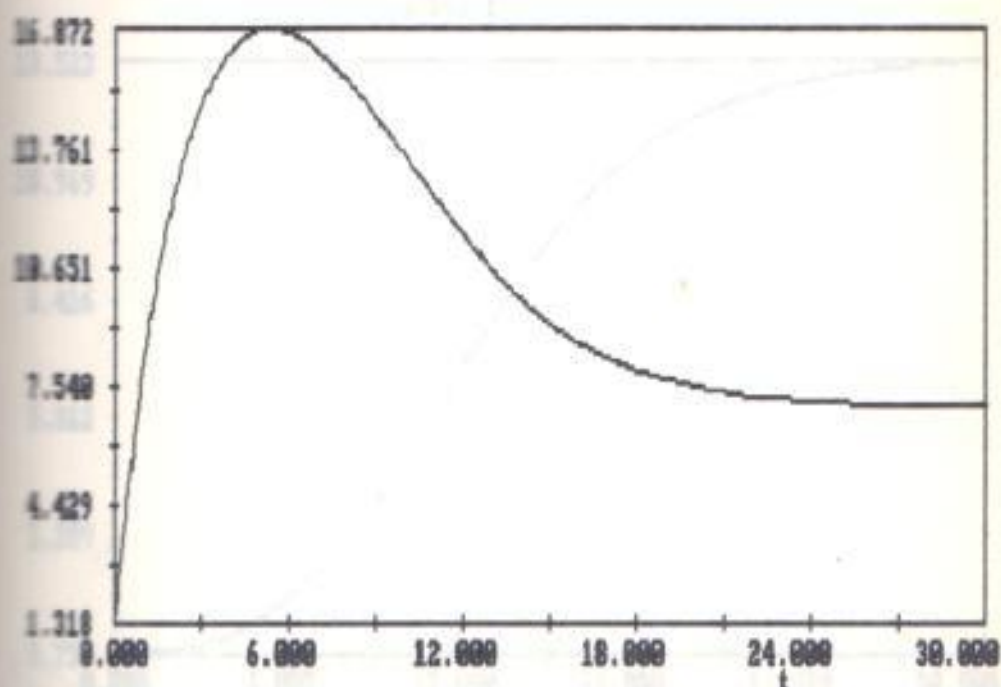
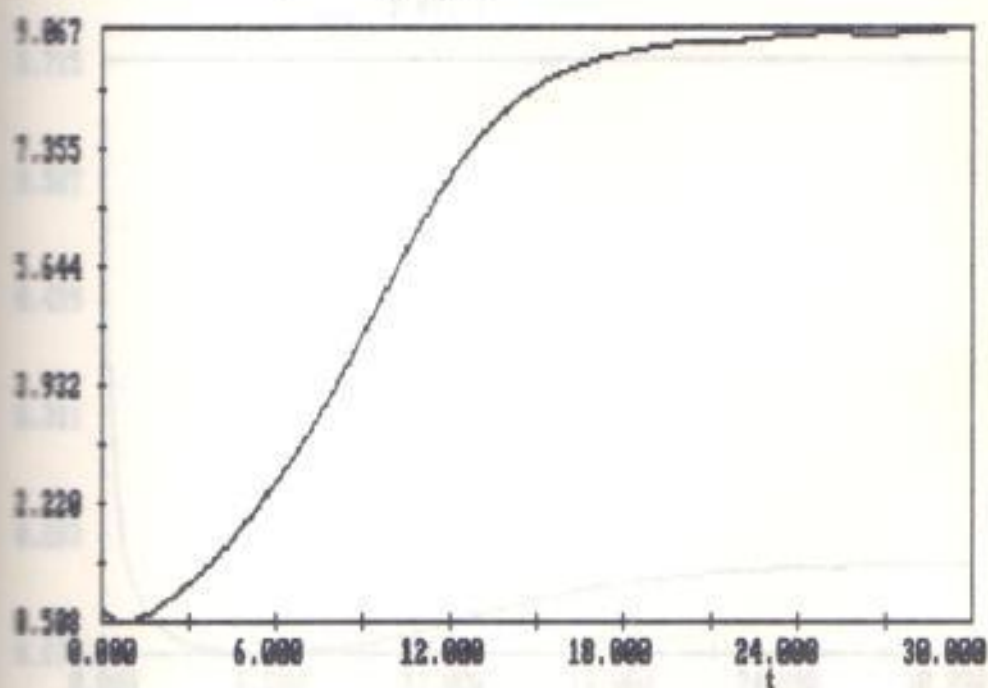
$$re = 0.862$$

$$ee = 0.078$$

La tabla de informe de conecciones de los integradores es:

Numero Integrador	Contenido Y	dx	dy	Y _m	n	s	z	p	L
1	dq/dt	dt	2,-3,-4 -5,-6	8	3	13	10	3	6
	C1	7	-	8	3	6	3	10	13
	C1	8	-	8	3	6	3	10	13
	C2	1	-	0.5	7	10	3	6	13
	C3	9	-	0.066	3	6	3	10	13
	C3	10	-	0.066	3	6	3	10	13
	r	1	-23	1	4	10	6	10	14
	q	23	1	16	4	10	6	10	14
	a	1	11	16	4	10	6	10	14
	q	11	1	16	4	10	6	10	14
	da/dt	dt	12,-13 -14,-15	1.1	3	13	10	6	9
	C4	16	-	0.5	4	10	6	9	13
	C5	11	-	0.3	4	10	6	9	13
	C6	9	-	0.005	0	6	6	13	13
	C6	10	-	0.005	0	6	6	13	13
	dm/dt	dt	17, 18, -19	0.9	3	13	10	6	9
	C7	9	-	0.1	0	6	6	13	13
	C7	10	-	0.1	0	6	6	13	13
	C8	16	-	1	4	10	6	9	13
	m	dt	16	16	4	13	9	10	14
	C9	1	-	0.76	0	10	10	13	13
	r	21	-23	1	0	10	10	10	10
	r	22	-23	1	0	10	10	10	10
	e	dt	17,18	10	4	13	9	6	10

Los gráficos de q, m, a, r y e los obtenemos de los integradores 8, 20, 9, 22 y 24 respectivamente y son los siguientes:

q Vs. t FIG. 4.15 PRODUCTO (Q) BAJO CONDICIONES OPTIMAS n Vs. t FIG. 4.16 DINERO (M) BAJO CONDICIONES OPTIMAS

a Vs. t

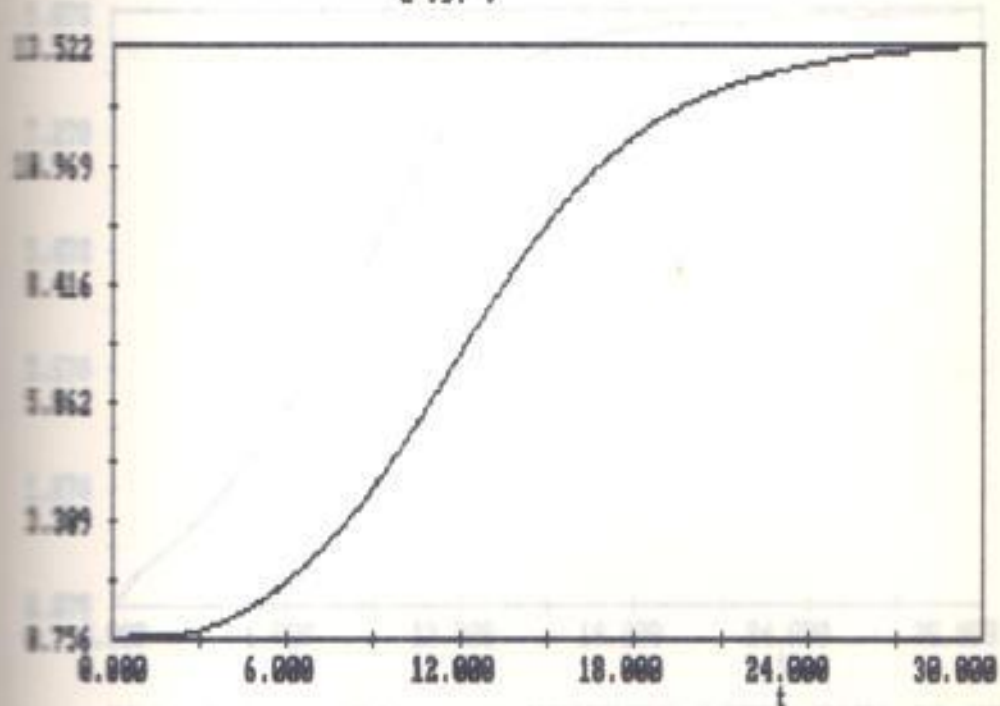


FIG. 4.17 ACTIVOS (A) BAJO CONDICIONES OPTIMAS

r Vs. t

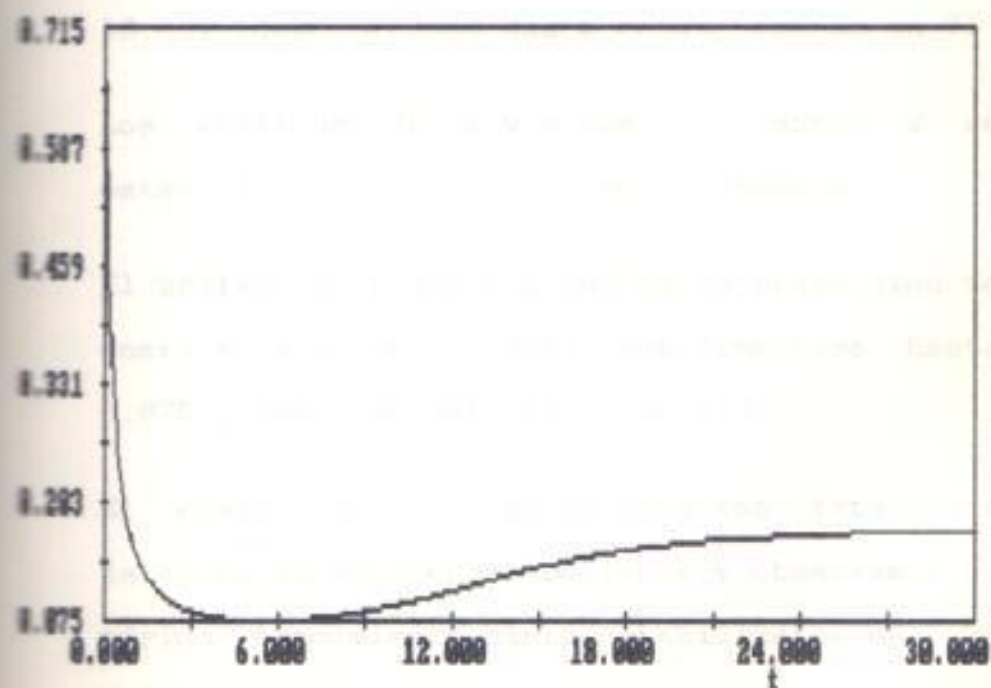


FIG. 4.18 PROPORCION DE ENERGIA UTILIZADA R

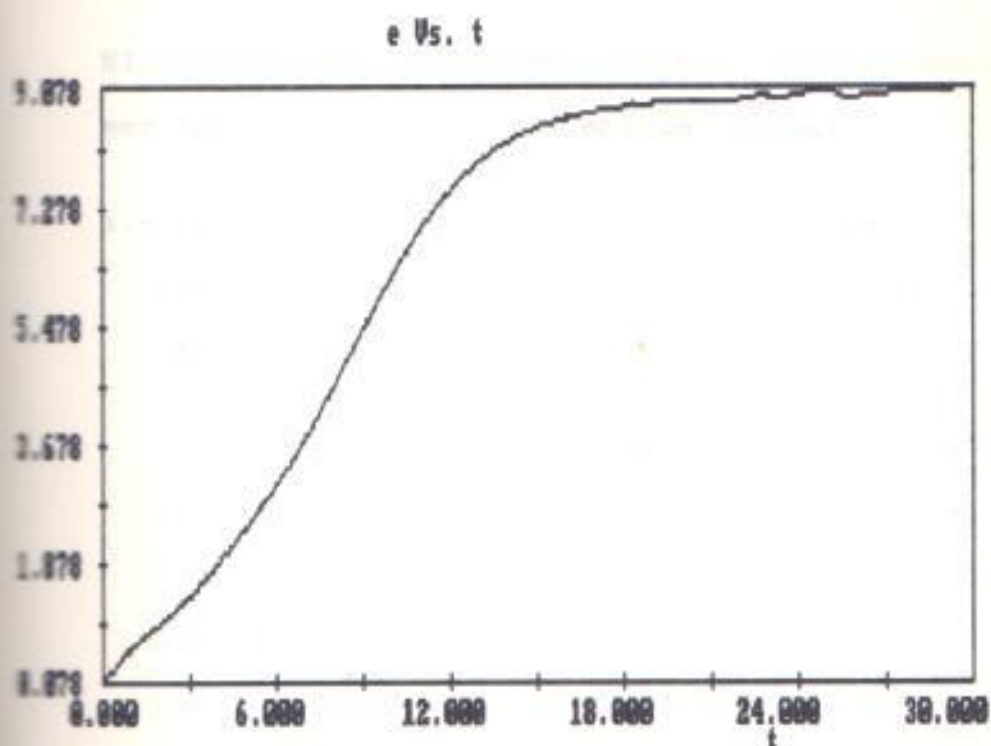


FIG. 4.19 CANTIDAD DE PRODUCTO LISTO PARA LA VENTA

Observamos en q que este tiene un rápido crecimiento hasta $t=6$ donde tiene un máximo de 16.8 y luego decrece hasta estabilizarse en 7.

Los gráficos de a y m son crecientes y se estabilizan en 13.5 y 9 respectivamente.

El gráfico de r que representa la proporción de energía capaz de ser utilizada disminuye hasta 0.075 y luego se estabiliza en 0.2.

El gráfico de e que es el producto listo para la venta es netamente creciente y observamos un rápido crecimiento hasta estabilizarse en $t=20$ a 9 u.

El autor sugiere una serie de preguntas para ser analizadas en el sistema de simulación.

a.- ¿Qué diferencia hace un 10% de incremento en el precio de mercancía y servicios hechos?

Tenemos que incrementar PG en 10% esto ocasiona que la C4 sea ahora 0.454 (integrador 12) y varían las condiciones iniciales en:

$$\left. \frac{da}{dt} \right|_0 = 0.111 \text{ (integrador 11)}$$

gráficos de q, m, a son:

q Vs. t

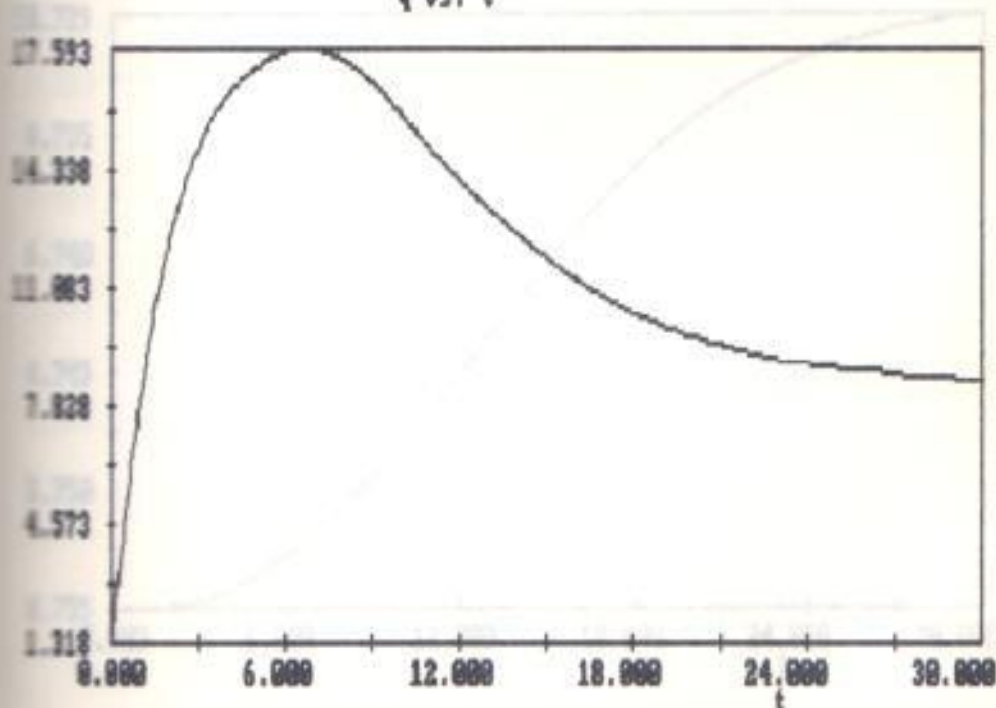


FIG. 4.20 PRODUCTO (Q) CON INCREMENTO DE PG EN 10%

n Vs. t

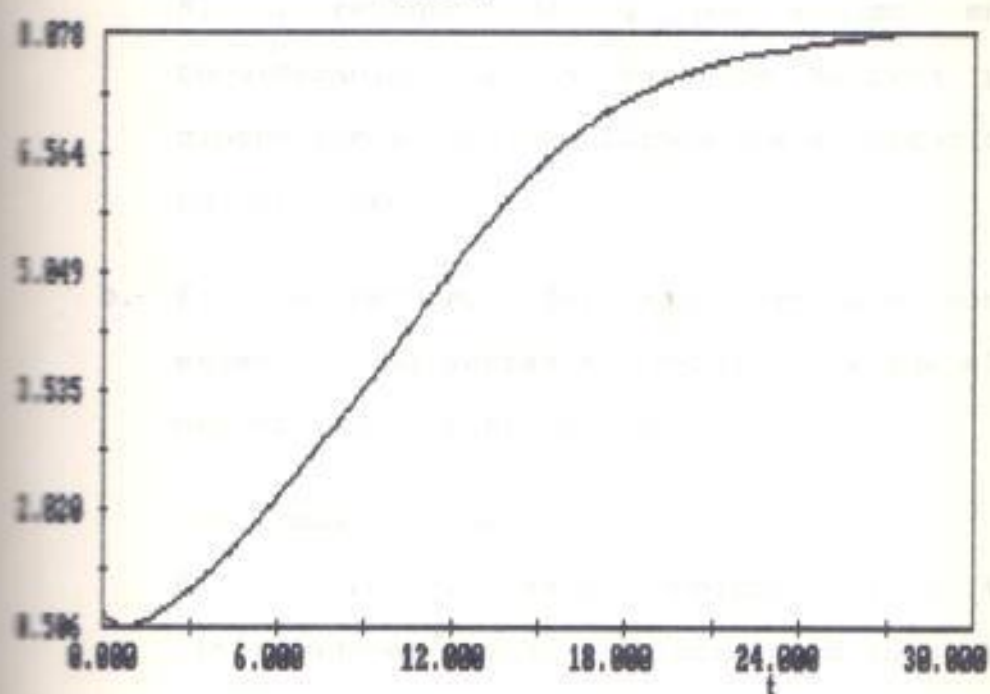


FIG. 4.21 DINERO (M) CON INCREMENTO DE PG EN 10%

a Vs. t

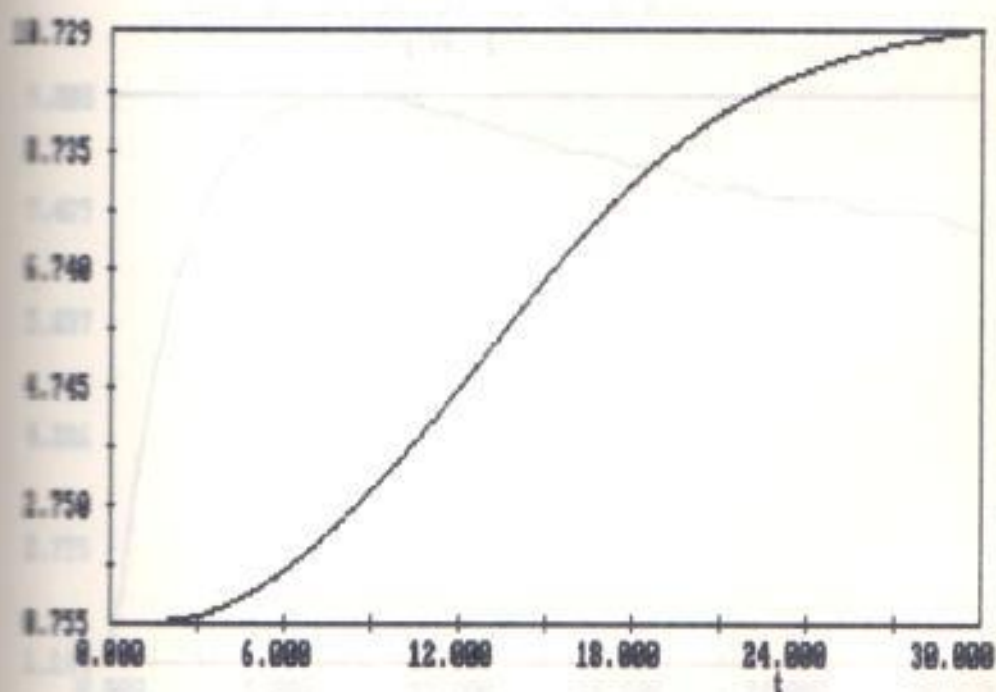


FIG. 4.22 ACTIVOS (A) CON INCREMENTO DE PG EN 10%

El incremento de q es mínimo en consideración de la baja de activos y dinero por lo que concluimos que el negocio no iría bien.

- b.- Si los recursos del medio ambiente son escasos, las ventas son menores. ¿Puede el usuario establecer un negocio?

Cambiamos I_0 a la mitad.

En nuestro simulador cambiamos $C_1 = 4$ (integradores 2 y 3) y la condición inicial de:

$$\left. \frac{dq}{dt} \right|_0 = 2.8981 \quad (\text{integrador 1})$$

Los gráficos de q , m , a son:

q Vs. t

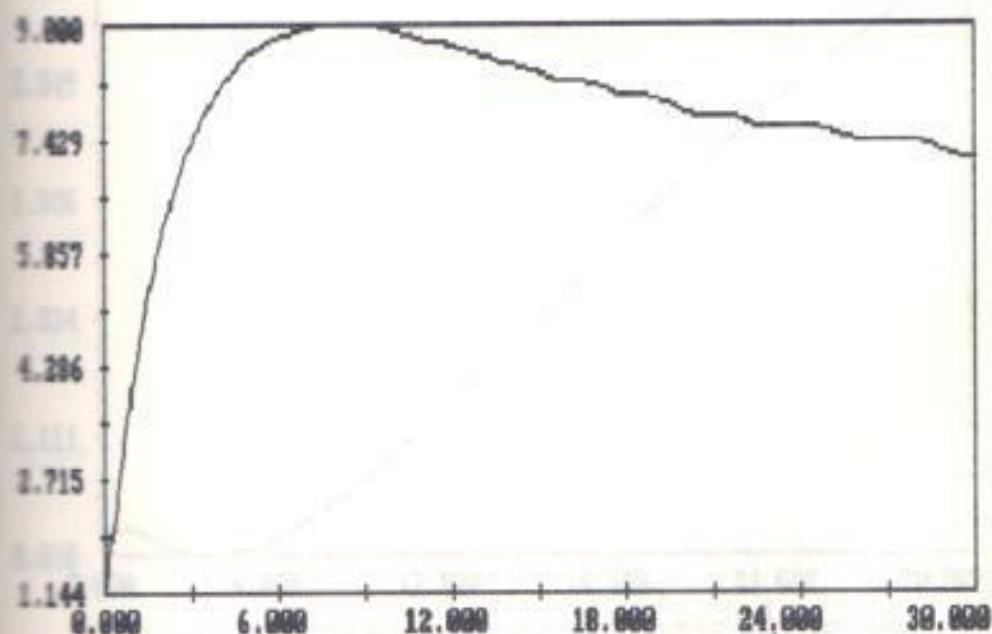


FIG. 4.23 PRODUCTO (Q) SI RECURSOS I_0 SE REDUCEN A LA MITAD

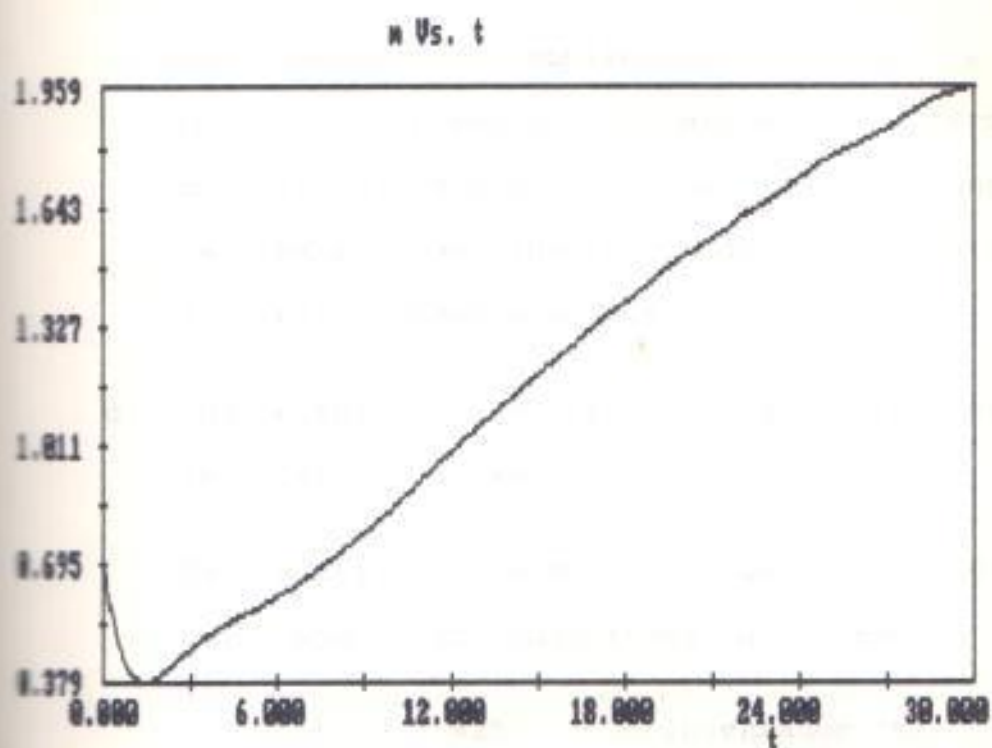


FIG. 4.24 DINERO (M) SI RECURSOS I_0 SE REDUCEN A LA MITAD

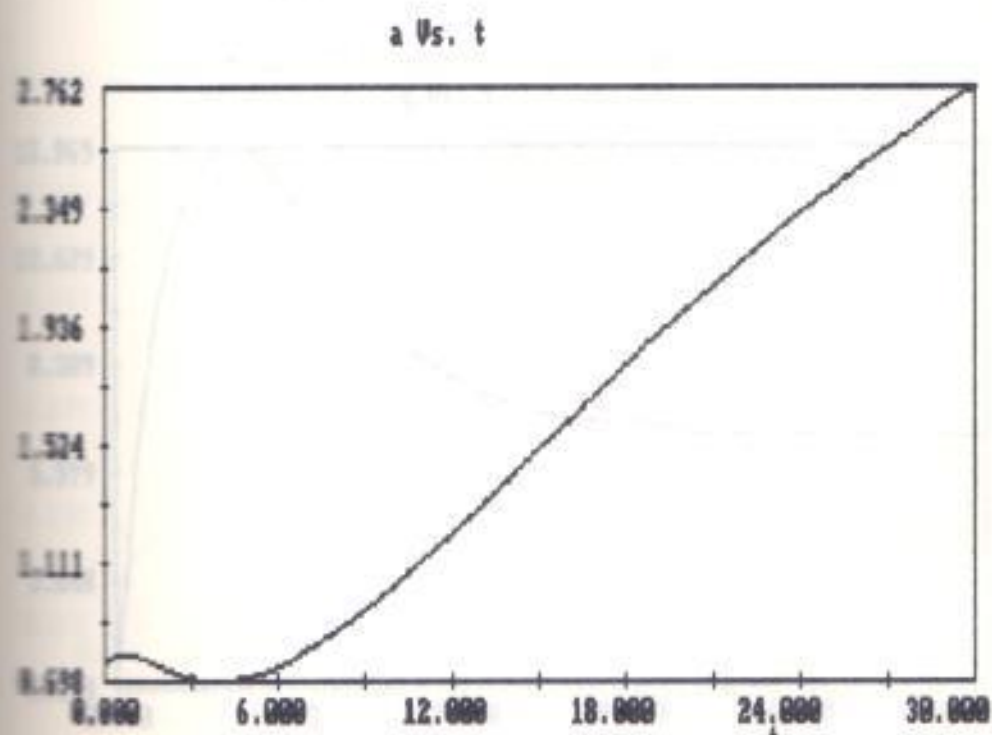


FIG. 4.25 ACTIVOS (A) SI RECURSOS I_0 SE REDUCEN A LA MITAD.

Los gráficos cambian notablemente el producto q se reduce a la mitad y el dinero m y los activos son escasos, por lo que tendremos que invertir nuestro dinero inicial en otros negocios.

c.- ¿Qué diferencia existe si usted inicia con 10 veces mas dinero?

Esto significa que $m_0 = 7.5$ (integrador 20) esto cambia las condiciones iniciales en:

$$\left. \frac{dm}{dt} \right|_0 = -7.425 \quad (\text{integrador 16})$$

$$\left. \frac{da}{dt} \right|_0 = 3.521 \quad (\text{integrador 11})$$

Los gráficos de q , m , a son:

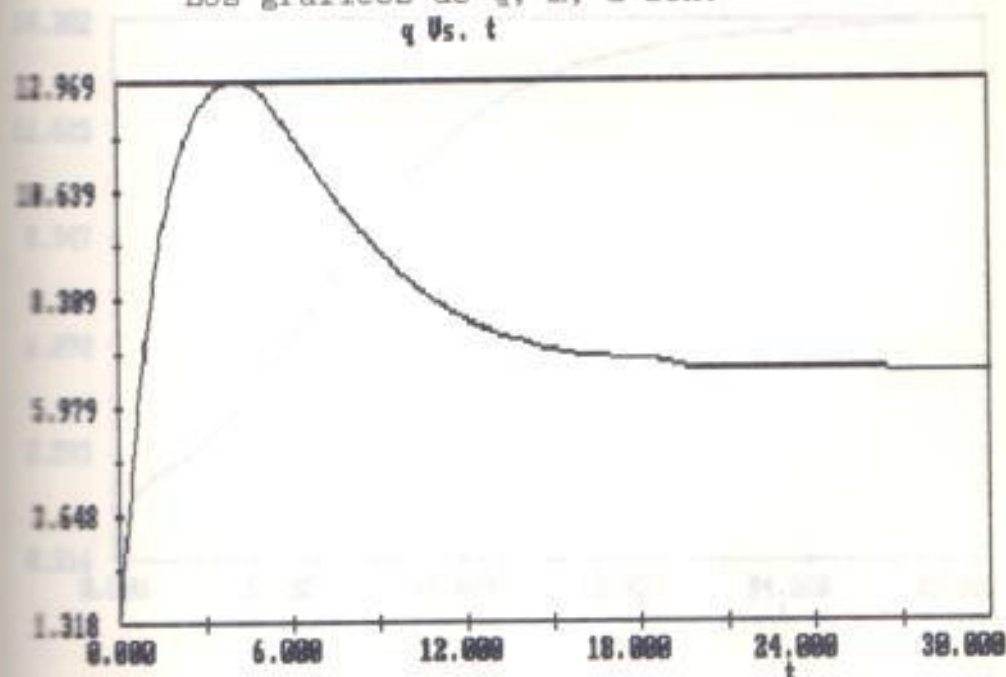


FIG. 4.26 PRODUCTO (Q) SI INVERSION INICIAL ES 10 VECES MAYOR

n Vs. t

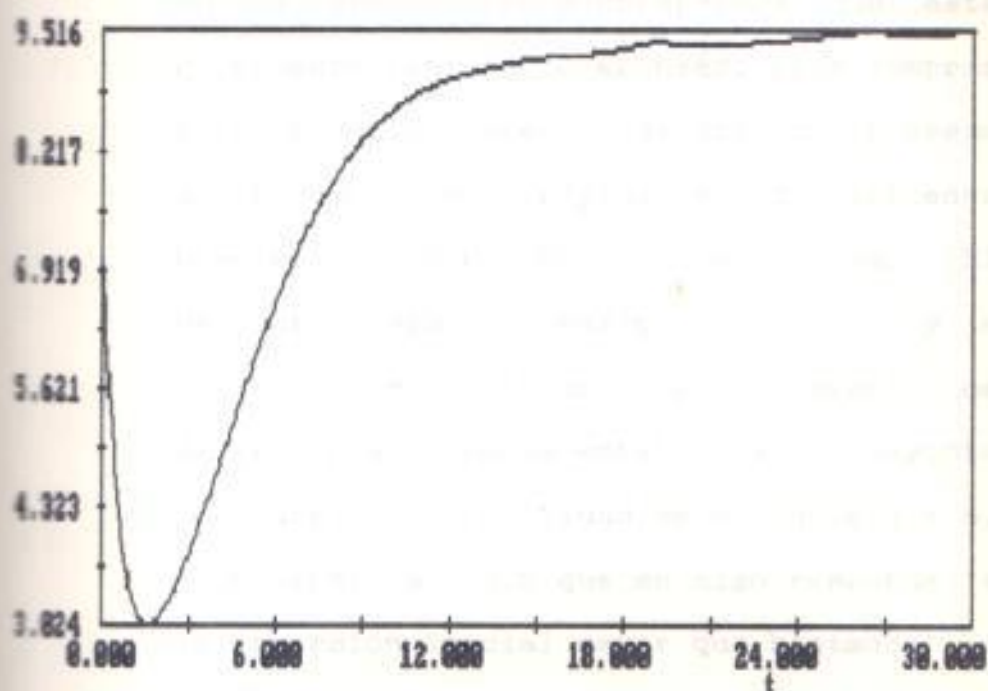


FIG. 4.27 DINERO (M) SI INVERSION INICIAL ES 10 VECES MAYOR

a Vs. t

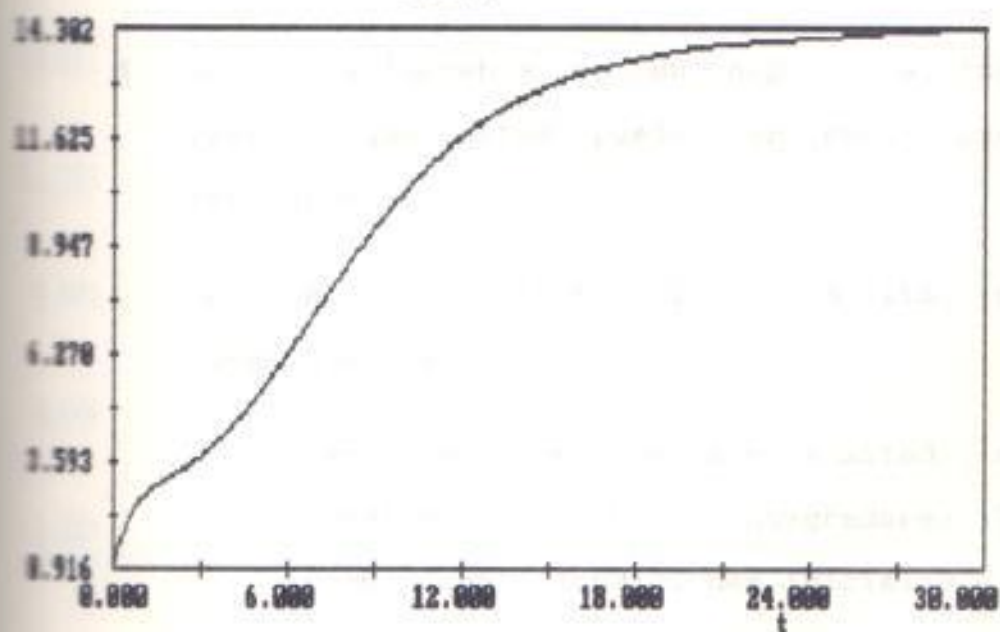


FIG. 4.28 ACTIVOS (A) SI INVERSION INICIAL ES 10 VECES MAYOR

En el gráfico de m observamos que este decreciente tanto como es usado para comprar activos rápidamente. Los activos procesan el producto mas rápido y lo mantienen creciendo tanto como el dinero baja. El producto llega a su máximo rápidamente y la venta de este produce un incremento de dinero hasta que se estabiliza el negocio con una cierta cantidad de producto $q = 6$, y un dinero $m = 9.5$ que en algo responde a la inversión inicial mayor que hicimos.

Además notamos que los activos son mas altos que es una respuesta a nuestra inversión.

d.- ¿Cuán vulnerable es su negocio a los precios que usted puede dar para sus productos?

Se sugiere que PE reducirlo a la mitad y luego duplicarlo.

d.1.- Para que PE se reduzca a la mitad la constante $C_7 = 0.05$ (integradores 17 y 18) cambia condiciones iniciales en

$$\left. \frac{dm}{dt} \right|_0 = -0.712 \quad (\text{integrador 16})$$

Al correr la simulación se saturó los integradores 8 y 10 de q puesto que excede a los máximos 16 indicados, por lo que tiene que incrementar el n a 5 en estos integradores sacrificando la precisión de dq/dt a un solo bit. Por tal razón el gráfico de q se observa no suavizado en su máximo, que es un poco mayor de 18.

Los gráficos de q , m , a son:

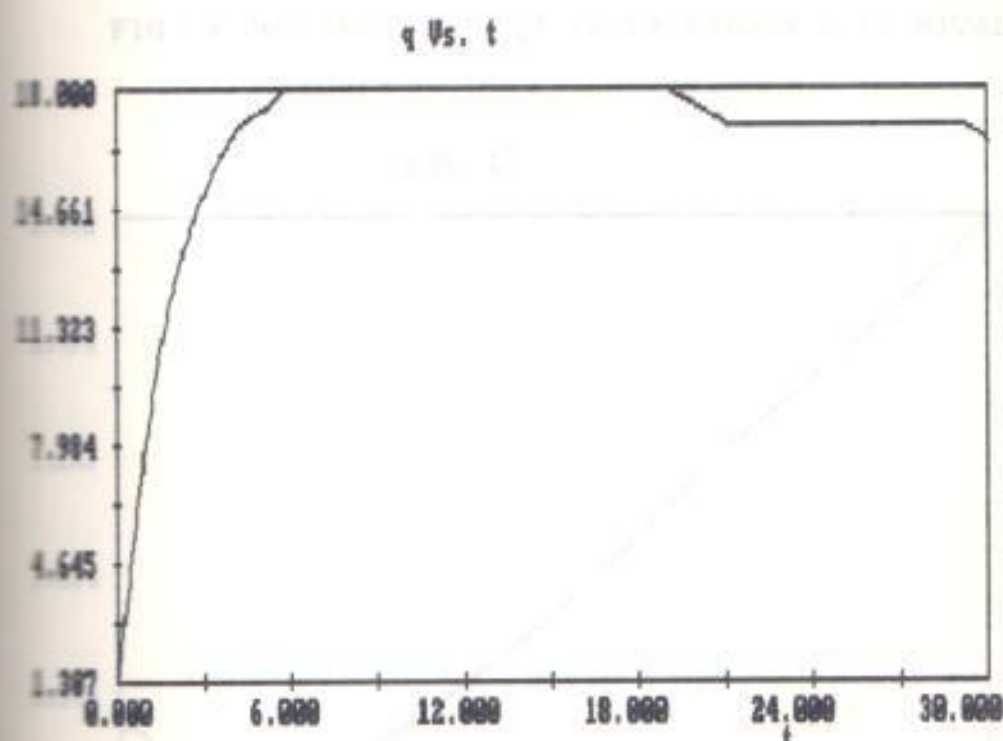


FIG. 4.29 PRODUCTO (Q) SI PE SE REDUCE A LA MITAD

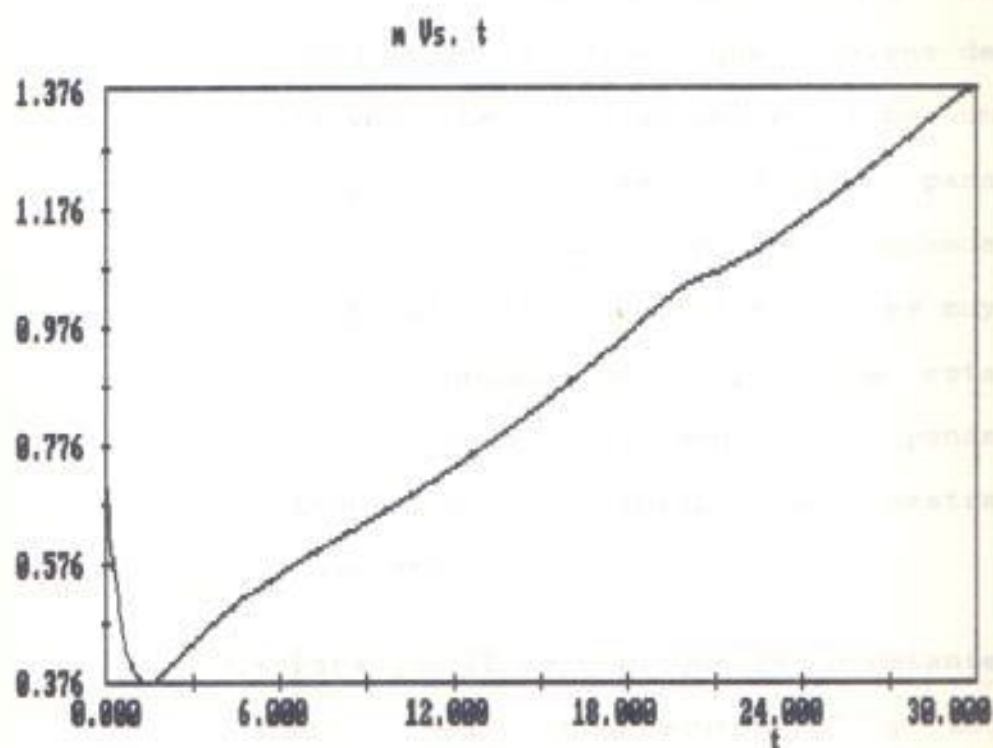


FIG. 4.30 DINERO (M) SI PE SE REDUCE A LA MITAD

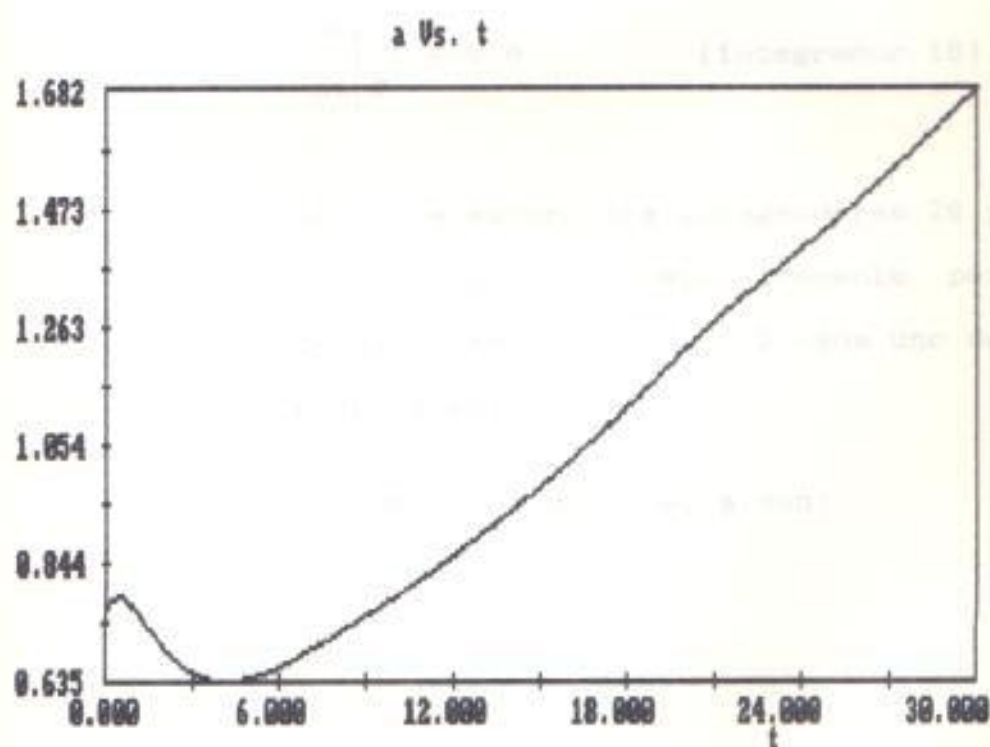


FIG. 4.31 ACTIVOS (A) SI PE SE REDUCE A LA MITAD.

Debido a que es mas barato el producto, el dinero que proviene de la venta de estos es menor y toma un largo tiempo ganar dinero para comprar activos, con esto notamos tanto el activo como el dinero es muy escaso cuando ya el producto esta listo para la venta. Apenas tendremos un reintegro de nuestra inversión.

d.2.- Para que PE se duplique la constante $C_7 = 0.2$ (integradores 17 y 18) cambia condición inicial en

$$\left. \frac{dm}{dt} \right|_0 = -0.6 \quad (\text{integrador 16})$$

Aquí se saturó los integradores 20 y 9 que son m y a respectivamente por lo que aumentamos a $n = 5$ cada uno de ellos y así:

Los gráficos de q, m, a son:

q Vs. t

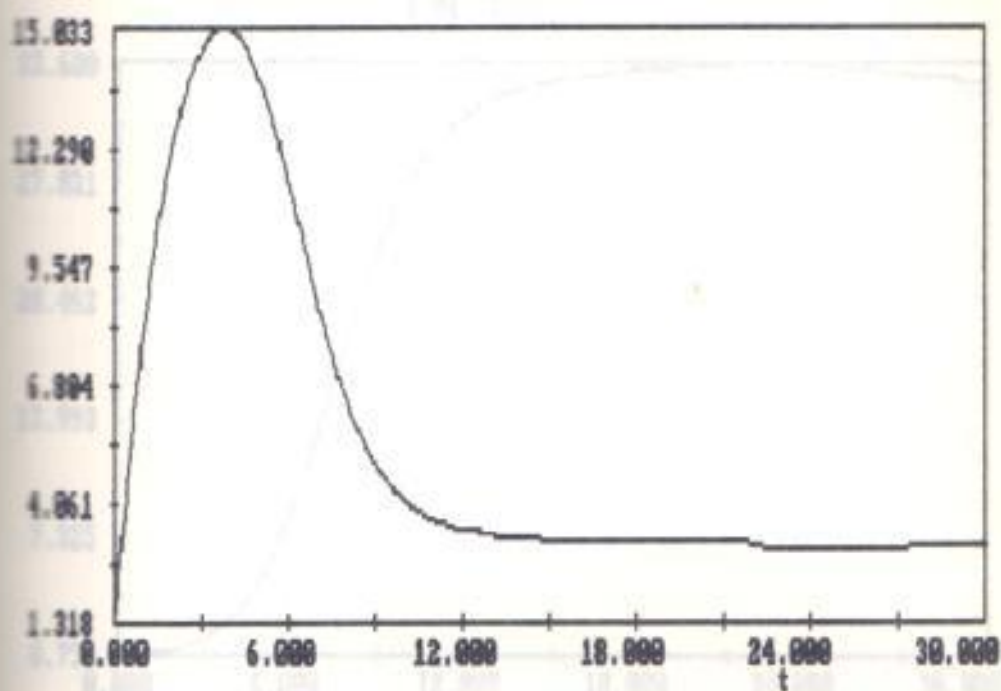


FIG. 4.32 PRODUCTO (Q) SI PE SE DUPLICA

n Vs. t

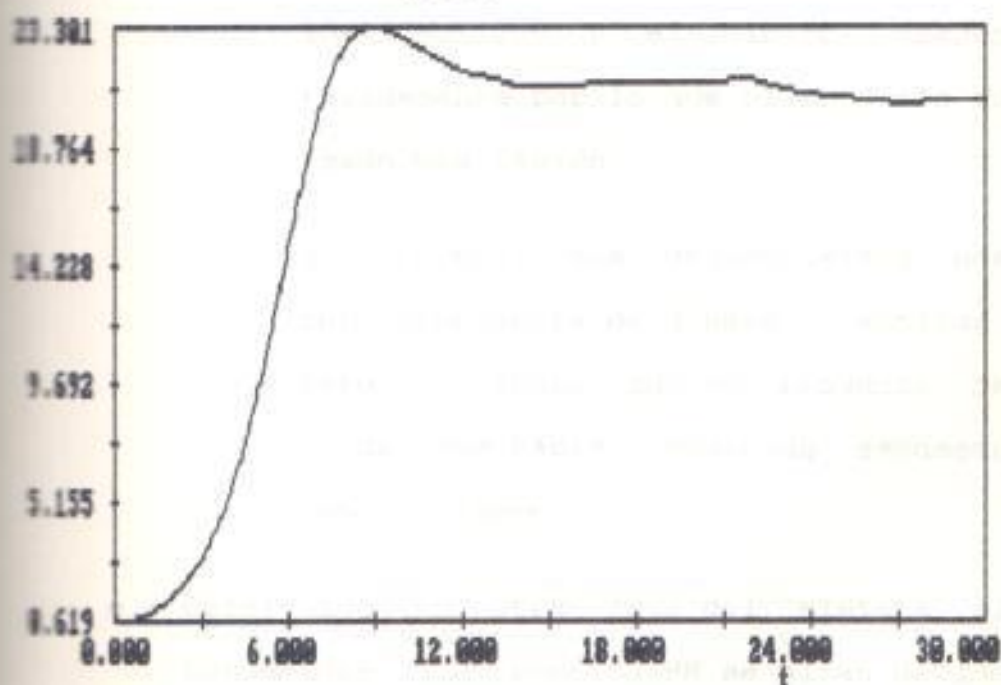


FIG. 4.33 DINERO (M) SI PE SE DUPLICA

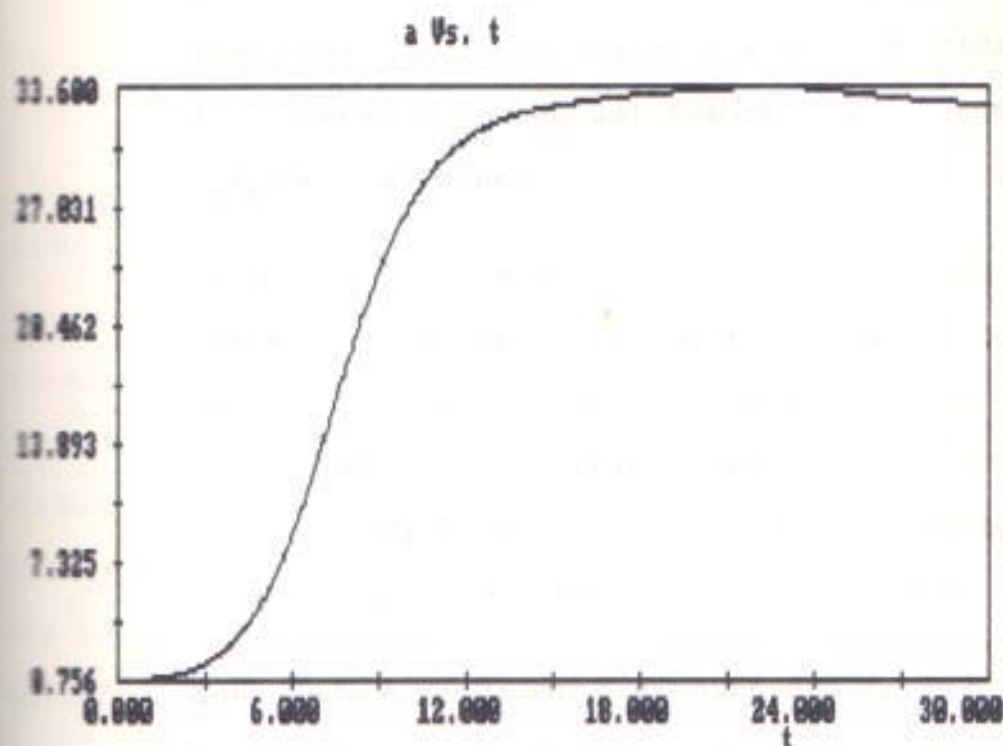


FIG. 4.34 ACTIVOS (A) SI PE SE DUPLICA

Como observamos al duplicar el precio los activos y el dinero crecen rapidamente por lo que el producto es usado mas rápido.

Bajo condiciones ideales existe una sobreabundancia de dinero y activos, pero ocasionan que el producto se venda mas rápido pudiendo escasear con el tiempo.

e.- Observamos en otra parte del sistema al consumidor. Si precios PE se alzan debido a escasez nacional del camarón, el stock de

Q en el estanque local es usado y disminuye. ¿Qué sucede al standard de vida de quienes dependen del camarón Q? ¿Qué compraría su dinero?

Al observar los gráficos de la segunda parte de la pregunta anterior en Q se estabiliza en 3.5 que es la mitad de los 7 que se estabilizaba normalmente; por lo que los consumidores pueden comprar solamente la mitad de lo que estaban acostumbrados.

Esto reduce seriamente el standard de vida de la gente que es dependiente de la venta y compra del camarón, lo que logicamente repercutiría en que el producto ellos lo pondrían mas caro.

REFERENCIAS

JOHNSON, DAVID E. Introduction to filter theory. Prentice Hall Electrical Engineering Series. 1976. pp 41-43.

JOHNSON, Op. Cit., pp 45-49.

JOHNSON, Op. Cit., pp 60-63.

CHIA, PEIR YU. Digital filter Design Technique and the realization of transfer and Immitance functions by using Digital Elements. Scientific Report No 22 National Aeronautics. 1967 . pp 36-37

CHIA, Op. Cit., pp 41-42.

ODUM, O. HOWARD T. Y ODUM, ELISABETH C. Minimodels and Simulations Exercises. Center for Wetlands University of Florida Gainesville. 1989 . pp 4-5

ODUM, Op. Cit., pp 171-172.

CONCLUSIONES Y RECOMENDACIONES.

El presente trabajo tuvo como finalidad dar un nuevo enfoque a la simulación convencional que se realizaba en la computadora analógica y concentrarnos en la simulación digital basada en la máquina DDA (Digital Differential Analyzer) cuyo único elemento es el integrador digital.

La máquina DDA es un instrumento muy poderoso para el análisis de variables que intervienen en un proceso. Su rapidez en el procesamiento de datos de resultados y su facilidad de expansión de elementos integradores hacen posible simular procesos en que intervienen muchas variables con buena efectividad.

Esperamos que nuestro trabajo se le de una aplicación en la investigación y las actividades docentes, especialmente al Laboratorio de Control Automático de la Facultad de Ingeniería Eléctrica. Previamente el estudiante debe de estar instruido en los conocimientos teóricos necesarios para que pueda realizar las conexiones y escalamiento de los integradores, esto es conocer el sistema que va a simular, plantear las ecuaciones que lo definen, y preparar el diagrama de conexiones de los integradores.

El programa que se desarrolló se lo ha tratado de realizar lo más amigable con el usuario, dando facilidad en el

ingreso de los datos, modificación y acceso en los
datos ya ingresados, facilidad en la emisión de un
reporte de la simulación efectuada: impreso de las
ecuaciones de los integradores, impreso de los datos
actuales de las diferentes variables, impreso de los
gráficos de dichas variables.

Esta versión se la podría mejorar adaptandolo a
computadoras que manejen mas bits de información y
para superar los 16 bits máximo de los registros
utilizados en esta máquina DDA, utilizar co-procesadores
aritméticos que servirán como unidad aritmética del
proceso de integración de la máquina DDA.

BIBLIOGRAFIA

1. BERGAMINI, EDUARDO W. A class of backward - Difference Digital Differential Analyzers for Incremental computer realizations. Technical report No 3606-7. Prepared under U.S. ARMY, U.S. NAVY and U.S. AIR FORCE. 1973
2. CHIA, PEIR YU. Digital filter Design Technique and the realization of transfer and Immitance functions by using Digital Elements. Scientific Report No 22 National Aeronautics. 1967
3. DAWOUD S. H. DAWOUD Y NADIA Z EL-ARABY. Parallel Digital Differential Analyzer with arbitrary stores Interconections. IEEE Transactions on Computer. Vol C-22 N-1 Enero 73 Pgs 41-46. 1973
4. HAKKALA, LAURY Y OJALA, LEO. Discretization Error Analysis in Linar DDA connections. IEEE Transactions on Computer. Vol C-23 N-9 Septiembre 74 pags 932-941. 1974
5. HERRERO, JOSE LUIS. Synthesis of filters. Englewood Cliffs. Prentice Hall. 1966
6. JOHNSON, DAVID E. Introduction to filter theory. Prentice Hall Electrical Engineering Series. 1976
7. MC. GHEE, ROBERT B Y NILSEN, RAGNAR N. The extended resolution Digital Differential Analyzer. A New

- computing Structure for solving Differential Equations. IEEE Transactions on Computer Vol C-19 N-1, enero 79, pages 1-9. 1979
8. MONROE, A.J. Digital Processes for Sampled Data Systems. pp 40-43. 1971
 9. ODUM, O. HOWARD T. Y ODUM, ELISABETH C. Minimodels and Simulations Exercises. Center for Wetlands University of Florida Gainesville. 1989
 10. PARASURAMAN B. Digital Incremental Computation using Automatic Programming Thecniques. Scientific Report No 40 National Aeronautics and Space Administration. 1972
 11. PETERSON, A.M. Simulation of Digital Integrators. 1972
 12. PHILIPPOT, P. Turbo Pascal para IBM - PC y compatibles. Procedimiento y funciones. Colección Informática de Gestión
 13. SCHULZ, E.J. Y PARASURAMAN, B. The Digital Incremental Computer. A New computing structure for the Numerical Solution of Differential Equation. Scientific Reports No 36. National Aeronautics and Space Administrations. 1971
 14. SCHULZ, E.J. Numerical Solution of Differential Equations. Technical Report No 3606-6 Prepared under U.S. ARMY, U.S. NAVY and U.S. AIR FORCE. 1971
 15. SIZER, T.R.H. C.ENG. M.I.E.E. The Digital Differential Analyser. Chapman and Hall Ltd. 1968
 16. WILLEN, DAVID C Y KRANTZ, JEFREY. 8088 Assembler Language Programming the IBM - PC. Howard W. Sams &

Co. Inc. 1983

17. WHITE, DONALD R.J. Electrical Filters, Synthesis Design and Applications. Don White Consultants Inc. 1980
18. TRS-80 Color Computer Technical Reference Manual. Radio Shack. Tandy Corporation. 1982
19. EDTASM+ Color Computer Editor Assembler with Zbug Tandy Corporation. 1982
20. TURBO PASCAL Reference Manual.

APENDICES

APENDICE A
DISEÑO DE PANTALLAS Y LISTADOS

DISEÑO DE PANTALLA								PDD-007	
Proyecto: DDA						Autor: RPP		Página: 1	
Identificación: DDASCR01						Versión: 1.0		Fecha: 10/90	
Programa: DDD.PAS			Librería			Diccionario			
	1	2	3	4	5	6	7	8	
1234567890123456789012345678901234567890123456789012345678901234567890									
1	MENU PRINCIPAL - ENTRADA DE DATOS POR TECLADO - ENTRADA DE DATOS POR ARCHIVO - COMUNICACION - SALIDA DE DATOS - AYUDAS - SALIDA HACIA EL SISTEMA OP. DDA								
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
TECLA ESPECIAL		DESCRIPCION							
	Up Dn	Teclas para el movimiento del cursor							
	ENTER	Tomar opción							

DISEÑO DE PANTALLA

PDD-007

Proyecto: DDA		Autor: RPP	Página: 1
Identificación: DDASCR02		Versión: 1.0	Fecha: 10/90
Programa: DDD.PAS	Librería	Diccionario	

	1	2	3	4	5	6	7	8
	1234567890123567890123456789012356789012345678901235678901234567890123567890							

1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

TECLA ESPECIAL		DESCRIPCION
	ENTER	Si el No. de integradores es mayor a seis pasa a leer el resto sino
		pasa a la siguiente pantalla
	ESC	Retorna a la pantalla anterior
	U D L R	Teclas para el movimiento del cursor

DISEÑO DE PANTALLA								PDD-007																																																																																										
Proyecto: DDA				Autor: RPP		Página: 1																																																																																												
Identificación: DDASCR03				Versión: 1.0		Fecha: 10/90																																																																																												
Programa: DDD.PAS		Librería		Diccionario																																																																																														
	1	2	3	4	5	6	7	8																																																																																										
1234567890123567890123456789012356789012345678901235678901234567890																																																																																																		
1	<table><tr><td>Número de Integradores</td><td>s-Bit de precisión</td><td>Valor Inicial</td><td>Valor Final</td><td>Paso de Impresión</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>								Número de Integradores	s-Bit de precisión	Valor Inicial	Valor Final	Paso de Impresión																																																																																					
Número de Integradores									s-Bit de precisión	Valor Inicial	Valor Final	Paso de Impresión																																																																																						
2																																																																																																		
3																																																																																																		
4																																																																																																		
5																																																																																																		
6																																																																																																		
7																																																																																																		
8																																																																																																		
9																																																																																																		
10	<table><tr><td>Int. #</td><td>Nombre de la variable</td><td>Sal</td><td>Factor de escala n</td><td>p-Bit de precisión</td><td>Condiciones Iniciales</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								Int. #	Nombre de la variable	Sal	Factor de escala n	p-Bit de precisión	Condiciones Iniciales																																																																																				
Int. #									Nombre de la variable	Sal	Factor de escala n	p-Bit de precisión	Condiciones Iniciales																																																																																					
11																																																																																																		
12																																																																																																		
13																																																																																																		
14																																																																																																		
15																																																																																																		
16																																																																																																		
17																																																																																																		
18																																																																																																		
19																																																																																																		
20																																																																																																		
21																																																																																																		
22																																																																																																		
23																																																																																																		
24																																																																																																		
25	<ENTER>=Continua <ESC>=Regresar < >=Movimiento en Pantalla																																																																																																	
TECLA ESPECIAL		DESCRIPCION																																																																																																
ENTER		Si el No. de integradores es mayor a seis pasa a leer el resto sino																																																																																																
		pasa a la siguiente pantalla.																																																																																																
ESC		Retorna a la pantalla anterior																																																																																																
U D L R		Teclas para el movimiento del cursor																																																																																																

DISEÑO DE PANTALLA								PDD-007
Proyecto: DDA					Autor: RPP		Página: 1	
Identificación: DDASCR04					Versión: 1.0		Fecha: 10/90	
Programa: DDD.PAS			Librería		Diccionario			
	1	2	3	4	5	6	7	8
1234567890123567890123456789012356789012345678901235678901234567890								
1					Graba : - No - Sí - Nuevo Archivo			
2								
3	MENU PRINCIPAL							
4								
5								
6								
7	- ENTRADA DE DATOS POR TECLADO							
8								
9								
10	- ENTRADA DE DATOS POR ARCHIVO							
11								
12								
13	- COMUNICACION							
14								
15								
16	- SALIDA DE DATOS							
17								
18								
19	- AYUDAS							
20								
21								
22	- SALIDA HACIA EL SISTEMA OP.							
23								
24								
25					DDA			
TECLA ESPECIAL		DESCRIPCION						
	Up Dn	Seleccionar opción						
	ENTER	Tomar opción						

PDD-007							
DISEÑO DE PANTALLA							
Proyecto: DDA				Autor: RPP		Página: 1	
Identificación: DDASCR05				Versión: 1.0		Fecha: 10/90	
Programa: DDD.PAS			Librería		Diccionario		
	1	2	3	4	5	6	7
	8						
	1234567890123567890123456789012356789012345678901235678901234567890						
1	MENU PRINCIPAL - ENTRADA DE DATOS POR TECLADO - ENTRADA DE DATOS POR ARCHIVO - COMUNICACION - SALIDA DE DATOS - AYUDAS - SALIDA HACIA EL SISTEMA OP. Datos de Entrada Datos de Salida Graficos Por Pantalla Por Impresora DDA						
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
TECLA ESPECIAL	DESCRIPCION						
Up Dn	Seleccionar opción						
ENTER	Tomar opción						

DISEÑO DE PANTALLA								PDD-007
Proyecto: DDA				Autor: RPP		Página: 1		
Identificación: DDASCR06				Versión: 1.0		Fecha: 10/90		
Programa: DDD.PAS		Librería		Diccionario				
	1	2	3	4	5	6	7	
	123456789012356789012345678901235678901234567890123567890123567890							
1	MENU PRINCIPAL - ENTRADA DE DATOS POR TECLADO - ENTRADA DE DATOS POR ARCHIVO - COMUNICACION - SALIDA DE DATOS - AYUDAS - SALIDA HACIA EL SISTEMA OP. Datos de Entrada Datos de Salida Graficos Por Pantalla Por Impresora SALIDAS Salir <nombre var> <# de los int. habilitados> DDA							
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
TECLA ESPECIAL	DESCRIPCION							
Up Dn	Movimiento del cursor							
L R	Movimiento del cursor							
ENTER	Selecciona integradores o acepta la salida de la pantalla							

DISEÑO DE PANTALLA								PDD-007
Proyecto: DDA						Autor: RPP		Página: 1
Identificación: DDASCR07						Versión: 1.0		Fecha: 10/90
Programa: DDD.PAS			Librería			Diccionario		
	1	2	3	4	5	6	7	8
	12345678901234567890123456789012345678901234567890123456789012345678901234567890							
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	MENU - LISTADO DE ARCHIVOS - DISCO - CAMBIO DE BASE DE IMPRESION - BORRAR ARCHIVO - SALIDA AL MENU PRICIPAL Mem. libre : Disco : Var. Base : DDA							
TECLA ESPECIAL		DESCRIPCION						
Up Dn		Seleccionar opción						
ENTER		Tomar opción						

DISEÑO DE PANTALLA								PDD-007
Proyecto: DDA				Autor: RPP		Página: 1		
Identificación: DDASCR08				Versión: 1.0		Fecha: 10/90		
Programa: DDD.PAS		Librería		Diccionario				
	1	2	3	4	5	6	7 8	
123456789012345678901234567890123456789012345678901234567890								
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	MENU - LISTADO DE ARCHIVOS - DISCO - CAMBIO DE BASE DE IMPRESION - BORRAR ARCHIVO - SALIDA AL MENU PRICIPAL no more files Mem. libre : Disco : Var. Base : < ENTER > Marcar Archivo < B > Borrar Archivo DDA							
TECLA ESPECIAL		DESCRIPCION						
Up Dn		Teclas de movimiento del cursor						
L R		Teclas de movimiento del cursor						
ENTER		Marcar archivo						
B		Borrar archivos marcados						

DISEÑO DE PANTALLA								PDD-007
Proyecto: DDA				Autor: RPP		Página: 1		
Identificación: DDASCR09				Versión: 1.0		Fecha: 10/90		
Programa: DDD.PAS		Librería		Diccionario				
	1	2	3	4	5	6	7	8
123456789012356789012345678901235678901234567890123567890123567890								
1	MENU - LISTADO DE ARCHIVOS - DISCO - CAMBIO DE BASE DE IMPRESION - BORRAR ARCHIVO - SALIDA AL MENU PRICIPAL DISCO : < A > < B > < C > DDA							
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
TECLA ESPECIAL		DESCRIPCION						
	Up Dn	Seleccionar disco						
	ENTER	Activar disco						

DISEÑO DE PANTALLA								PDD-007
Proyecto: DDA				Autor: RPP		Página: 1		
Identificación: DDASCR10				Versión: 1.0		Fecha: 10/90		
Programa: DDD.PAS		Librería		Diccionario				
	1	2	3	4	5	6	7 8	
123456789012345678901234567890123456789012345678901234567890								
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	MENU - LISTADO DE ARCHIVOS - DISCO - CAMBIO DE BASE DE IMPRESION - BORRAR ARCHIVO - SALIDA AL MENU PRICIPAL VARIABLE INDEPENDIENTE : - TIEMPO - DDA							
TECLA ESPECIAL		DESCRIPCION						
Up Dn		Seleccionar variable independiente						
ENTER		Fijar variable independiente						

DISEÑO DE SALIDA LISTADA										PDD-011A	
Proyecto: DDA					Autor: RPP			Página: 1			
Identificación: DDALST01					Versión: 1.0			Fecha: 10-OCT-90			
Programa: DDD.PAS			Nombre del Listado: Datos de Entrada								
	1	2	3	4	5	6	7	8			
	1234567890123456789012345678901234567890123456789012345678901234567890										
1	-----										
2	Archivo :										
3	DATOS DE ENTRADA										
4	-----										
5											
6	Número de integradores :										
7	S_bit de precision :										
8	Paso de impresión :										
9	Valor Inicial :										
10	Valor Final :										
11											
12	Nombre del Integrador	dy1	dy2	dy3	dy5	dy5	dx	n	p	C.Inic	SALE
13	-----										
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											
34											
35											
36											
37											
38											

DISEÑO DE SALIDA LISTADA

PDD-011A

Proyecto: DDA		Autor: RPP	Página: 1	
Identificación: DDALST02		Versión: 1.0	Fecha: 10-OCT-90	
Programa: DDD.PAS		Nombre del Listado: Datos de Salida		

	1	2	3	4	5	6	7	8
	1234567890123567890123456789012356789012345678901235678901234567890123567890							
1	<nombre archivo>							
2								
3	Paso	<Base>	<... numero de integradores ...>					
4			<... nombres de los integradores ...>					
5								
6	-----							
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								
37								
38								



APENDICE B



BIBLIOTECA

MANUAL DEL USUARIO

El Analizador Diferencial Digital (DDA) es un método de simulación digital de un proceso, utilizando los integradores digitales.

El usuario debe conocer muy bien para cada una de las variables del sistema a simular su rango de valores.

Manualmente debe seguir los siguientes pasos:

- a. Escribir las ecuaciones diferenciales que describan el sistema y sus limitantes.
- b. Manipular las ecuaciones para escribirlas en función de sus diferenciales.
- c. Basados en estas ecuaciones dibujar el diagrama de conecciones de los integradores
- d. Realizar el escalamiento del diagrama en función de la variable s y escoger un valor apropiado para dicha variable.
- e. Hacer un informe de las conecciones, escalamiento y valores iniciales de todos los integradores.

Proceda a energizar la COCOII y la IBM-PC. En la primera cargue el assembler del 6809 por medio del cartucho ensamblador, siga estos pasos:

- En la casetera ingrese el cassette que contiene el programa
- Ponga en modo PLAY
- Presione la tecla "L" del COCOII y la tecla <ENTER> dos veces.
- Comenzará a cargarse el programa y espere el mensaje de listo.
- Escriba "A/IM/WE" para que ensamble el programa.
- Presione la tecla "Z" para ir al ZBUG.
- Escriba "GSTART" y presione la tecla <ENTER>

La COCOII entra en un modo de espera lista para aceptar el ingreso de datos del IBM-PC.

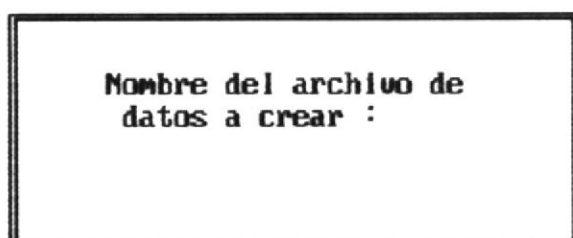
La IBM-PC enciendala con el sistema operativo DOS versión 3.0 o una versión mas actual.

Coloque el diskette que tenga el programa DDD.COM y ejecútelo escribiendo "DDD" y presionando la tecla <ENTER>.

Observe la pantalla DDA SCR01 (Diseño de pantalla) el menú

principal.

Si es una sesión que se quiere ingresar los datos de la simulación por primera vez, escoja la opción "INGRESO DE DATOS POR TECLADO". Se abre una ventana en la que usted debe ingresar el nombre del archivo que quiere que se graben sus datos de entrada fig. B.1; presione la tecla <ENTER> y observe que se publica la pantalla DDA SCR02 que es la de ingreso de datos y es donde usted debe escribir su tabla de informe.



A screenshot of a terminal window with a double-line border. Inside the window, the text "Nombre del archivo de" is on the first line and "datos a crear :" is on the second line. The text is in a monospaced font.

FIG. B.1 VENTANA INGRESO NOMBRE DE ARCHIVO

Esta pantalla es dinámica pues depende del valor del número de integradores para que se edite igual número de filas donde se coloca la información de cada integrador. En la primera fila usted detalla la información general de la máquina DDA que es:

- Número de Integradores: Ingrese un número entero máximo de 2 dígitos que representa la cantidad de integradores con los que realiza la simulación de su proceso.
- S - bit de precisión: Ingrese un número entero entre 0 y

15 que representa en bits que precisión tiene la variable independiente respecto a la unidad.

- Valor Inicial: Valor real que representa el punto inicial de la variable independiente donde comienza la simulación.
- Valor final: Valor real que representa el punto final de la variable independiente donde termina la simulación. Asegúrese que es mayor que el valor inicial.
- Paso de Impresión: Número natural que indica cuantas iteraciones pasan antes de emitir un resultado.

Al presionar la tecla <ENTER>, si el número de integradores es menor a 6, se editan igual número de filas pero si es mayor se editarán en múltiplos de 6 hasta que finalice con el último integrador.

En cada fila usted ingresa información particular de cada integrador, así:

- Nombre de variable: Caracteres alfanuméricos, máximo de 32, con el cual distinguimos a la variable que maneja el integrador en el registro Y.
- dy1, dy2, dy3, dy4, dy5: Números enteros positivos o negativos que representan el número del integrador del cual proviene la conexión al dy correspondiente, si no existe la conexión ingrese 0, y si la entrada es la

variable independiente, ingrese el número 101.

- dy: Número entero positivo que representa el número del integrador del cual proviene la conexión al dx, si la entrada es la variable independiente ingrese el número 101.

Al terminar usted de ingresar la información de las conexiones de todos los integradores, presione la tecla `ENTER>` y pasa a la pantalla DDA SCRO3 en la que ingresa la información de escalamiento de cada uno de los integradores.

De la misma forma que la anterior pantalla, ésta es dinámica pues se publican en cada pantalla 6 integradores, dependiendo del número de integradores.

La primera columna se publica el nombre de variable del integrador, pero como ya ingresó estos nombres, no tiene acceso a esta columna.

La información que se ingresa por columna es:

- n - Factor de Escala: Número entero entre 0 y 15 que representa el número de bit que acota los valores máximos y mínimos de la variable que maneja dicho integrador.
- p - Bit de Precisión: Número entero entre 0 y 15 que representa el número de bit de precisión del dy total

del integrador.

- SALE: Caracter "S" o "N" que indica si a este integrador lo definimos de salida o no.
- Condición Inicial: Número Real, que es el valor inicial que toma la variable en el integrador.

Finalizado el ingreso de los datos de entrada presione la tecla <ENTER> y aparece una ventana que le da la opción de grabar o no, Fig. B.2.

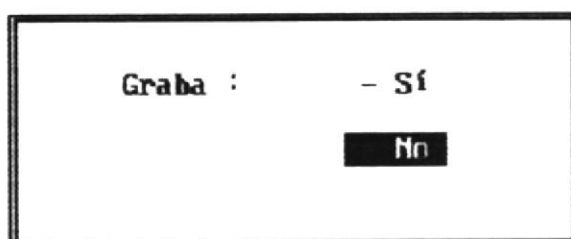


FIG. B.2 VENTANA CON OPCION DE "GRABAR"

Al escoger la opción con las teclas de movimiento presione la tecla <ENTER> y se grabaran todos los datos de entrada en un archivo en el disco de trabajo.

Si usted quiere modificar los datos de entrada entonces escoja la opción de "Ingreso de Datos por Archivo" del menú principal. Al presionar la tecla <ENTER> tenemos una ventana que le pide ingresar el nombre del archivo en que supuestamente usted tiene grabados los datos de entrada. Si no recuerda el nombre al presionar la tecla <ENTER> se publica una ventana con los nombres de los



archivos que usted ha creado en sesiones anteriores Fig. B.3 y que se encuentran disponibles; estos se presentan en grupos de 10 y al presionar la tecla "C" pasa a otro grupo; con las teclas de movimiento selecciona el nombre del archivo que quiere editar y al presionar la tecla <ENTER> entra a la pantalla de edición de los datos de entrada de la simulación y puede facilmente cambiar cualquier dato.



FIG. B.3 VENTANA PARA ELEGIR NOMBRE DEL ARCHIVO.

Cuando termine de ingresar los nuevos datos presiona la tecla <ENTER> y se observa la pantalla DDA SCRO4 que le pregunta si quiere grabar o no y ademas añade una nueva opción que es la de crear un nuevo archivo, pues usted podría querer grabar estos datos en otro archivo sin cambiar el original con el objeto de futuras comparaciones de resultado entre ellos; al escoger esta opción digita el

nombré par le conseil de la ville

<continues
EC03A, BDA
EC04, BDA
EC03, BDA
EC031, BDA
H, BDA
EC02, BDA
EC01US, BDA
EC01SE, BDA
VANDER, BDA
PAI, BDA

Le conseil de la ville a élu le 10 mai 1961

nombre que quiera y presione <ENTER> para que se graben los datos en este nuevo archivo.

La ejecución de la simulación se la realiza al escoger la opción de COMUNICACION en el menú principal y usted debe recurrir a ella únicamente cuando se han ingresado los datos de entrada o actualizado estos; al presionar la tecla <ENTER> se observa una ventana de advertencia Fig. B.4, pues usted debe estar seguro que el cable de comunicación entre la COCOII y la IBM-PC se encuentre debidamente conectado, además como los datos de repuestas se graban en el disco de trabajo debe asegurarse que existe la suficiente capacidad de memoria para guardar estos resultados.

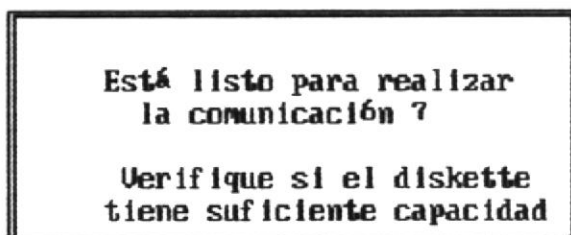


FIG. B.4 VENTANA DE ADVERTENCIA PARA LA COMUNICACION.

Si no esta seguro presione la tecla <ESC> y regresa al menú principal y si presiona la tecla <ENTER> se observa el mensaje "COMUNICACION REALIZANDOSE", pantalla DDA SCRO6.

En breve se observa el número de bytes que se transmite y luego cuando se termine de ejecutar el proceso se muestra

el número de bytes que se reciben; al presionar <ENTER> se va al menú principal.

Para hacer un reporte de los datos de entrada y resultados escoja la opción "SALIDA DE DATOS" del menú principal, presione <ENTER> y tiene 3 opciones que son:

- Datos de entrada.
- Datos de salida.
- Gráficos.



BIBLIOTECA

Al escoger cualquiera de estas opciones verá una ventana en la que se selecciona el dispositivo de salida que son los standard : "PANTALLA" o "IMPRESORA" pantalla DDA SCRO6.

En "DATOS DE ENTRADA" por pantalla se publican los datos que ingresó para la interconexión de los integradores. El formato en que se emiten estos datos lo observa en el diseño de listado DDA LIS01.

Para el listado por impresora utilice papel continuo standard. Energize la impresora seteada en el modo de escritura que desee y pongala en línea antes de presionar la tecla <ENTER> para comenzar a imprimir.

En "DATO DE SALIDA" por pantalla, se publican valores numéricos de las variables que manejan los integradores de salida. Con la tecla <ENTER> se observa una ventana cuyo formato es (pantalla DDA SCRO7):

- En la primera línea como título: "SALIDAS"
- En la segunda línea se encuentra el nombre de la variable que direcciona el cursor.
- En la tercera y cuarta línea los números de los integradores que se los programa previamente como salida.

Con las teclas de movimiento usted selecciona el número del integrador que desee reportar los valores de su variable, a medida que mueve el cursor usted observa que cambia el campo de la segunda línea con el nombre de la variable que está indicando. Si presiona la tecla <ENTER> cambia el color del número indicado al igual que el nombre de la variable respectiva. Usted puede seleccionar el número de variables que quiera, pero por pantalla solo se van a observar seis.

En la primera línea al lado derecho está la opción "SALIR", si usted se mueve con el cursor hasta esa posición después de seleccionar a los integradores y presiona <ENTER>, pasa a la pantalla DDA LIS02 que es un informe completo de las variables de los integradores.

Cada columna representa una variable del integrador y las dos columnas iniciales son el número de iteración y el valor numérico de la variable independiente.

Si no son presentados todos los valores de las variables

en una pantalla, con el hecho de presionar <ENTER> usted pasará a los siguientes valores hasta que termine el proceso, esto es el valor final de la variable independiente. Presionando <ENTER> regresa al menú principal.

Si ha seleccionado mas de 6 integradores para su publicación, el resto de los valores de las variables de los integradores faltantes aparece en cuanto se termine de revisar los 6 primeros.

En "DATO DE SALIDA" por impresora, siga los mismos pasos que la emisión por pantalla, el listado se imprime con el mismo formato anterior. Asegurese que la impresora se encuentre energizada y en línea, utilice papel continuo tamaño standard. Presione dos veces la tecla <ENTER> y se iniciará la impresión; no olvide que si escogió mas de 6 integradores, se imprimen las 6 primeras variables en un listado y el resto en otro.

La comprensión de la variación de las variables usted la adquiere al observar un gráfico. En la opción de "GRAFICO" por pantalla vuelve a aparecer la ventana que escoge a las variables DDA SCRO7, que se va a representar en el gráfico, por medio del número del integrador, pero ahora, con la limitante que en una pantalla puede graficar máximo tres variables al mismo tiempo, de igual forma la opción "SALIR" le da la posibilidad de ir a observar el

gráfico solicitado. Si escogió solo graficar una variable, en el eje de las abscisas se marca la escala de la variable independiente entre el valor inicial y final que usted definió; en el eje de las ordenadas se marca el rango de la variable a graficar entre un mínimo y un máximo que la máquina las reconoce automáticamente, Fig. B.5. Como título del gráfico lleva el nombre de la variable versus nombre de la variable independiente y en cada eje se escribe el nombre de la variable respectiva.

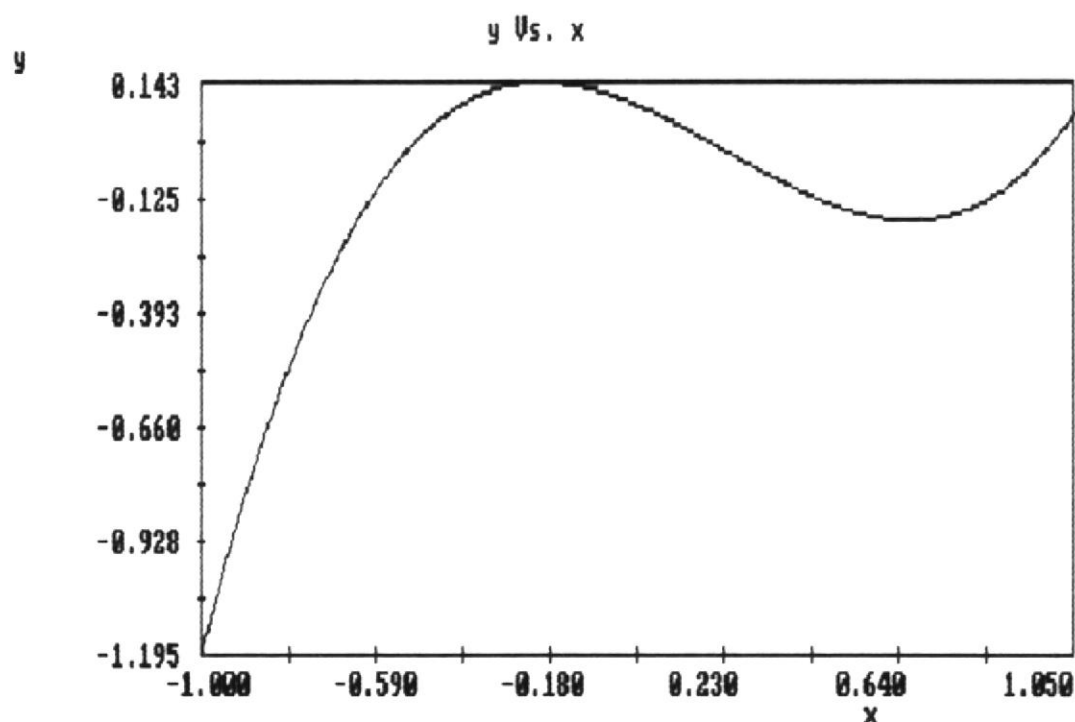


FIG. B.5 GRAFICO DE UNA VARIABLE DEL INTEGRADOR DIGITAL.

Cuando escoge para graficar dos o tres variables, a cada una le asigna un distintivo que puede ser un triángulo, cuadrado y equis, y se los coloca en el punto inicial y final del gráfico; en la parte del título van los nombres de las variables representadas y su símbolo respectivo,

100.0 100.0 100.0 100.0 100.0 100.0

100.0 100.0 100.0 100.0 100.0 100.0

Fig. B.6. Como existen dos o tres variables estas se marcan en dos o tres ejes de ordenada respectivo y son dos a la izquierda y una a la derecha, cada uno de estos lleva la escala entre los mínimos y máximos del rango de valores de cada variable.

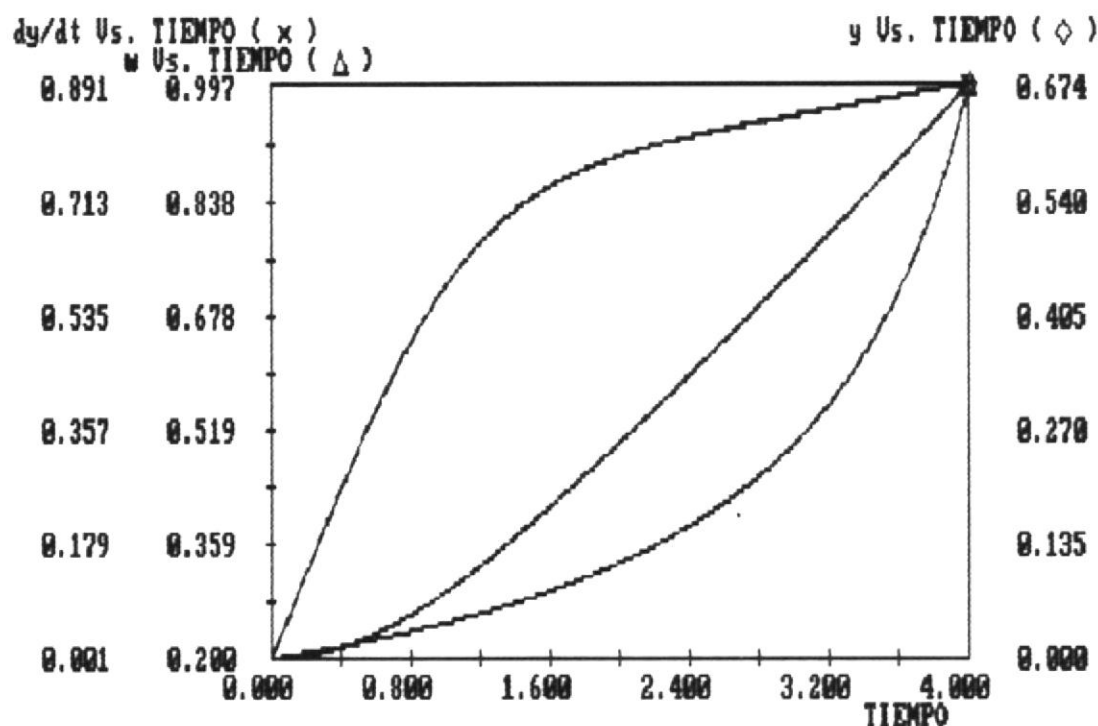


FIG. B.6 GRAFICOS SUPERPUESTOS DE TRES VARIABLES.

Si usted quiere imprimir estos gráficos, antes de ejecutar el programa "DDA", usted debe ejecutar el programa del DOS, GRAPHICS.COM.

En la opción "GRAFICOS" por impresora siga los mismos pasos que el anterior, el gráfico deseado va a salir impreso en una hoja standard. Si la impresora no está energizada y no está en línea, la opción ejecutada no lo hará y regresa al menú principal.



La opción "AYUDAS" nos da la pantalla DDA SCRO8 que está definida en tres partes, dos superiores y una inferior. La primera ventana superior izquierda es la que le brinda el menú de ayuda. La segunda ventana superior derecha le informa del status de la máquina DDA y la ventana inferior es para publicar los nombres de los archivos existentes en su directorio.

La ventana de Status le da tres informes:

- Memoria libre en su disco
- Disco de acceso, que puede ser A, B o C
- Nombre de la variable independiente

La ventana del menú le da las siguientes opciones:

- LISTADO DE ARCHIVOS, al presionar <ENTER> en la ventana inferior se publican en cinco columnas todos los archivos que se encuentran en su directorio.
- DISCO, al presionar <ENTER> en la ventana superior derecha aparecen las letra A, B, C, y con las teclas de movimiento usted escoge su disco de trabajo, presiona <ENTER> y la ventana de status cambia con la nueva información.
- CAMBIO DE BASE DE IMPRESION, presiona <ENTER> y en la ventana superior derecha se publica en dos líneas "TIEMPO" y una línea en blanco, si quiere cambiar el nombre de esta variable se posiciona en esta línea y

escribe el nuevo nombre, presiona <ENTER> dos veces y se actualiza en la ventana de status; también en los listados y gráficos aparece este nombre designando a la variable independiente.

- BORRAR ARCHIVO, presiona <ENTER> y el cursor pasa a la ventana inferior donde esta el listado de archivo, con las teclas de movimiento usted se ubica en los nombres de los archivos que quiera borrar y los selecciona presionando la tecla <ENTER>, automáticamente el nombre del archivo cambia de color, una vez terminada la selección, presione la tecla y borrará todos los archivos marcados. Si no desea borrarlos presiona <ESC> y regresa al menú de ayuda.
- SALIDA AL MENU PRINCIPAL, como se indica presionando <ENTER> en esta posición regresa al menú principal.

Una vez terminada su sesión de trabajo, retorna al DOS ubicandose en la opción "SALIDA HACIA EL SISTEMA OPERATIVO" del menú principal y presionando <ENTER>.

APENDICE C
LISTADO DE LOS PROGRAMAS

C.1 LISTADO DEL PROGRAMA EN EL CPU PARA EL MANEJO DE
LOS DATOS DE ENTRADA, SALIDA Y COMUNICACION DE
LA MAQUINA DDA

```
(*****)
(* ProgramPrincipal constituye el programa de arranque del DDA que contiene *)
(* las invocaciones de las opciones del menu principal: Leer datos por te- *)
(* clado, Leer datos por archivo, Comunicacion, Salida de Datos y Ayudas. *)
(* La salida de datos tiene disponible hacerlo para datos de entrada y sa- *)
(* lida, ademas de las graficas. Los datos de entrada, salida y graficos *)
(* pueden hacerlo tanto a la pantalla como a la impresora. *)
(*****)
```

```
PROGRAM programprincipal;
```

```
{ $i graph.p }
```

```
TYPE
```

```
disquette = (A,B,C);
switch    = (on,off);
entry     = STRING[12];
pathname  = STRING[63];
workstring = STRING[80];
arr       = ARRAY[1..40] OF REAL;
path_mask = ARRAY[1..63] OF CHAR;
lstring   = ARRAY[1..80] OF STRING[22];
resulta   = RECORD
    AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
END;
archivos  = RECORD
    primero:STRING[22];
    segundo:ARRAY[1..6] OF INTEGER;
    tercero:ARRAY[1..3] OF REAL;
END;
filvar    = FILE OF archivos;
```

```
VAR
```

```
{** Variables simples **}
```

```
imprime,cc,ress1,ress,ok,ok1: BOOLEAN;
ReadOk,WriteOk : BYTE;
ii,s,kat,sat,sss,kkk,lll,fin,nite,graf,result,nint,nint2,lar,prnt,
buffer,largo,color,i,j,k,accion,l,x,z,x1,hal,x2,y1,y2 : INTEGER;
numr,xst,xfn : REAL;
ch,res : CHAR;
numero1,numero2 : STRING[18];
filo,numero,xnom,sttr,s1,s2,s3,s4,s5,s6,fsttr,dsttr,vsttr,xsttr,
vr,fr,dr : STRING[80];
disc1,disc2,disco : disquette;
mask : path_mask;
base,basel,lin1,lin2,lin3,lin4,lin5 : workstring;
```

```
{** Arreglos **}
```

```
span,simp,gpan,gimp,pan,imp : ARRAY[1..20] OF BOOLEAN;
dta : ARRAY[1..43] OF BYTE;
vasa : ARRAY[1..20] OF INTEGER;
```



BIBLIOTECA

```

dy1,dy2,dy3,dy4,dy5,dx,y,cx,d,cy : ARRAY[1..40] OF INTEGER;
cinic : ARRAY[1..80] OF REAL;
sal,nom : ARRAY[1..80] OF STRING [22];
ddas,borra : ARRAY[1..80] OF entry;

```

```

{** Archivos **}

```

```

dofile,infile : filvar;
textfile : TEXT;
archi : archivos;

```

```

{** Archivos donde se encuentran los procedimientos y funciones **}

```

```

{$i rosado.pas}
{$i rutinas.pas}
{$i entrada.pas}
{$i hcomuni.pas}
{$i leearchi.pas}

```

```

BEGIN

```

```

    disco:=defdrive;
    base:='TIEMPO';
    base1:='FRECUENCIA';
    FOR i:=1 TO 80 DO
        BEGIN
            ddas[i]:='no more file';
            borra[i]:='no more file';

```

```

        END;

```

```

    FOR i:=1 TO 20 DO

```

```

        BEGIN

```

```

            imp[i]:=FALSE;
            pan[i]:=FALSE;
            gimp[i]:=FALSE;
            gpan[i]:=FALSE;
            simp[i]:=FALSE;
            span[i]:=FALSE;

```

```

        END;

```

```

    accion:=5;

```

```

    REPEAT

```

```

        men;

```

```

        largo:=6;

```

```

        color:=1;

```

```

        curso(accion);

```

```

        CASE accion OF

```

```

            2 : BEGIN {** Lectura de datos de entrada por teclado **}

```

```

                largo:=8;

```

```

                color:=5;

```

```

                ventana(5);

```

```

                gotoxy(7,3);

```

```

                Write('Nombre del archivo de');

```

```

                gotoxy(7,4);

```

```

                Write(' datos a crear :');

```



BIBLIOTECA


```

ok:=FALSE;
i:=0;
gotoxy(12,6);
s:=12;
REPEAT { Inicia proceso de lectura }
  IF keypressed THEN
    BEGIN
      Read(kbd,ch);
      IF (ch = #27) THEN BEGIN i:=-199 ;ok:=TRUE;END
      ELSE
        IF ok = FALSE THEN
          BEGIN
            ok:=TRUE;
            filo:='';
            s:=12;
            REPEAT
              IF (((ch<#26)AND(ch<>#8)) OR ((ch>#30) AND
                (ch<#122))) THEN
                BEGIN
                  filo:=concat(filo,ch);
                  gotoxy(s,6);
                  Write(ch);
                END;
              Read (kbd,ch);
              IF ((ch=#8) OR (ch=#127)) THEN
                BEGIN
                  gotoxy(s,6);
                  Write(' ');
                  gotoxy(s,6);
                  s:=s-1;
                  filo:=copy(filo,1,s-11);
                END
              ELSE
                s:=s+1;
              IF s=11 THEN s:=12;
            UNTIL (ch= #13);
          END;
        END;
      UNTIL ok;
      IF i<>-199 THEN
        BEGIN
          l:=8;
          IF length(filo)<8 THEN l:=length(filo)+1;
          FOR i:=1 TO 8 DO
            IF filo[i]='.' THEN l:=i;
            filo:=concat(copy(filo,1,l-1),'.dda');
            textbackground(5);
            res:=FALSE;
            leearchivo(filo);
            men;
            ventana(5);
            gotoxy(7,3);
            Write(' Graba :');

```

```

gotoxy(21,3);
Write(' - Si ');
gotoxy(21,5);
Write(' - No ');
l:=3;
ok:=FALSE;
REPEAT
  gotoxy(21,l);
  j:=1;
  IF l=3 THEN
    BEGIN
      textbackground(15);
      textcolor(0);
      Write(' - Si ');
    END;
  IF l=5 THEN
    BEGIN
      textbackground(15);
      textcolor(0);
      Write(' - No ');
    END;
  cc:=FALSE;
  REPEAT
    IF keypressed THEN
      BEGIN
        Read(kbd,ch);
        IF (ch = #13) THEN
          BEGIN
            ok:=TRUE;
            cc:=TRUE;
          END;
        IF (ch = #27) AND keypressed THEN
          BEGIN
            Read(kbd,ch);
            IF (ch = 'H') THEN
              BEGIN
                l:=l-2;
                cc:=TRUE;
              END;
            IF (ch = 'P') THEN
              BEGIN
                l:=l+2;
                cc:=TRUE;
              END;
            IF l<3 THEN l:=5;
            IF l>5 THEN l:=3;
          END;
        END;
      UNTIL cc;
      textbackground(5);
      textcolor(15);
      gotoxy(21,j);
      IF j=3 THEN

```



```

        Write (' - Si ');
        IF j=5 THEN
            Write (' - No ');
        UNTIL ok;
        IF l=3 THEN graba(filo);
    END;
END;
5 : BEGIN {** Lectura de datos entrada por archivo **}
    largo:=8;
    color:=5;
    disc2:=defdrive;
    setdrive(disco);
    ventana(8);
    gotoxy(4,3);
    Write('Nombre del archivo de datos');
    gotoxy(12,5);
    ok:=FALSE;
    REPEAT
        IF keypressed THEN
            BEGIN
                Read(kbd,ch);
                IF (ch = #13) THEN
                    BEGIN {** Se muestra archivos creados para escoger **}
                        ok:=TRUE;
                        j:=1;
                        ddas[1]:=firstentry('???????.dda',$20);
                        IF (ddas[j]<>'No more file') OR
                            (ddas[j]<>'No such file') THEN
                            BEGIN
                                REPEAT
                                    j:=j+1;
                                    ddas[j]:=nextentry($4F);
                                    IF ddas[j]<>'no more file' THEN
                                        BEGIN
                                            i:=0;
                                            REPEAT
                                                i:=i+1;
                                                UNTIL (ddas[j][i]='.') OR (i=13);
                                                ddas[j]:=concat(copy(ddas[j],1,i+3),
                                                                    '.');
                                            END;
                                        UNTIL ddas[j]='no more file';
                                        lar:=j-1;
                                        window(54,8,68,20);
                                        textbackground(7);
                                        textcolor(5);
                                        clrscr;
                                        j:=j-1;
                                        IF lar>10 THEN j:=10;
                                        window(55,9,67,22);
                                        gotoxy(1,1);
                                        FOR i:=1 TO j DO
                                            IF ddas[i]<>'no more file' THEN

```

```

    WriteLn(ddas[i]);
l:=1;
ii:=0;
ok:=FALSE;
textcolor(0);
gotoxy(1,12);
WriteLn('<C>ontinua ');
window(55,9,67,19);
REPEAT
    cc:=FALSE;
    textbackground(5);
    textcolor(0);
    gotoxy(1,1);
    WriteLn(ddas[ii+1]);
    gotoxy(1,1);
    lll:=1;
    REPEAT
    IF keypressed THEN
        BEGIN
            Read(kbd,ch);
            IF (ch='c') OR (ch='C') THEN
                BEGIN
                    cc:=TRUE;
                    IF j=lar THEN ok:=TRUE;
                    j:=j+10;
                    IF j>lar THEN j:=lar;
                    gotoxy(1,1);
                    textbackground(7);
                    textcolor(5);
                    clrscr;
                    ii:=ii+10;
                    l:=1;
                    IF not ok THEN FOR i:=(ii+1) TO j DO
                        WriteLn(ddas[i]);
                    END;
                END;
            IF (ch = #13) THEN
                BEGIN
                    ok:=TRUE;
                    cc:=TRUE;
                END;
            IF (ch = #27) AND keypressed THEN
                BEGIN
                    Read(kbd,ch);
                    IF (ch = 'H') THEN
                        BEGIN
                            l:=l-1;
                            cc:=TRUE;
                        END;
                    IF (ch = 'P') THEN
                        BEGIN
                            l:=l+1;
                            cc:=TRUE;
                        END;
                END;
            END;
        END;
    END;

```



```

        IF l<1 THEN l:=j-ii;
        IF l>j-ii THEN l:=1;
    END
    ELSE IF (ch = #27) THEN
        BEGIN
            ok:=TRUE;
            cc:=TRUE;
            i:=-199;
        END;
    END;
    UNTIL cc;
    gotoxy(1,111);
    textbackground(7);
    textcolor(5);
    IF not ok THEN WriteLn(ddas[ii+111]);
    111:=1;
UNTIL ok;
setdrive(disc2);
filo:=ddas[ii+111];
ok:=TRUE;
END
ELSE
BEGIN
    i:=-199;
    ok:=TRUE;
END;
END;
IF (ch = #27) THEN BEGIN i:=-199 ;ok:=TRUE;END
ELSE
    IF ok = FALSE THEN
        BEGIN
            ok:=TRUE;
            filo:='';
            s:=12;
            REPEAT
                IF (((ch<#26)AND(ch<>#8)) OR ((ch>#30)
                    AND (ch<#122))) THEN
                    BEGIN
                        filo:=concat(filo,ch);
                        gotoxy(s,5);
                        Write(ch);
                    END;
                Read (kbd,ch);
                IF ((ch=#8) OR (ch=#127)) THEN
                    BEGIN
                        gotoxy(s,5);
                        Write(' ');
                        gotoxy(s,5);
                        s:=s-1;
                        filo:=copy(filo,1,s-11);
                    END
                ELSE
                    IF (ch <> #27) THEN s:=s+1;

```

```

        IF s=11 THEN s:=12;
        IF ((ch=#27) AND (not keypressed)) THEN i:=-199;
UNTIL ((ch= #13) OR ((ch=#27) AND (not keypressed)));
l:=8;
IF length(filo)<8 THEN l:=length(filo)+1;
FOR i:=1 TO 8 DO
    IF filo[i]='.' THEN l:=i;
    filo:=concat(copy(filo,1,l-1),'.dda');
END;
END;
UNTIL ok;
IF i<>-199 THEN
BEGIN
    dassign(filo,6,6);
    IF ok THEN
    BEGIN
        ress:=TRUE;
        i:=0;
        leearchivo(filo);
        close(infile);
        ress:=FALSE;
        men;
    IF i<>-199 THEN BEGIN
        ventana(5);
        gotoxy(7,2);
        Write(' Graba :');
        gotoxy(12,5);
        Write(' - Si ');
        gotoxy(12,3);
        Write(' - No ');
        gotoxy(12,7);
        Write(' - Nuevo Archivo');
        l:=3;
        ok:=FALSE;
        REPEAT
            gotoxy(12,l);
            j:=1;
            IF l=5 THEN
            BEGIN
                textbackground(15);
                textcolor(0);
                Write(' - Si ');
            END;
            IF l=3 THEN
            BEGIN
                textbackground(15);
                textcolor(0);
                Write(' - No ');
            END;
            IF l=7 THEN
            BEGIN
                textbackground(15);
                textcolor(0);

```



```

        Write(' - Nuevo Archivo');
END;
cc:=FALSE;
REPEAT
    IF keypressed THEN
        BEGIN
            Read(kbd,ch);
            IF (ch = #13) THEN
                BEGIN
                    ok:=TRUE;
                    cc:=TRUE;
                END;
            IF (ch = #27) AND keypressed THEN
                BEGIN
                    Read(kbd,ch);
                    IF (ch = 'H') THEN
                        BEGIN
                            l:=l-2;
                            cc:=TRUE;
                        END;
                    IF (ch = 'P') THEN
                        BEGIN
                            l:=l+2;
                            cc:=TRUE;
                        END;
                    IF l<3 THEN l:=7;
                    IF l>7 THEN l:=3;
                END;
            END;
        UNTIL cc;
        textbackground(5);
        textcolor(15);
        gotoxy(12,j);
        IF j=5 THEN
            Write(' - Si ');
        IF j=3 THEN
            Write(' - No ');
        IF j=7 THEN
            Write(' - Nuevo Archivo');
        UNTIL ok;
        IF l=5 THEN graba(filo);
        IF l=7 THEN
            BEGIN
                FOR i:=2 TO 7 DO BEGIN
                    gotoxy(6,i);
                    Write(' ');
                END;
                gotoxy(4,4);
                Write('Nombre del nuevo Archivo:');
                gotoxy(6,6);
                numero:='';
                Read(numero);
                IF numero<>' ' THEN

```

```

        BEGIN
            l:=8;
            IF length(numero)<8 THEN l:=length(numero)+1;
            FOR i:=1 TO 8 DO
                IF numero[i]='.' THEN l:=i;
                filo:=concat(copy(numero,1,l-1),'.dda');
                graba(filo);
            END;
        END;
    END;
END;
END;
8:BEGIN {** Comunicacion **}
    largo:=8;
    color:=5;
    ventana(11);
    gotoxy(7,3);
    Write('Esta listo para realizar');
    gotoxy(7,4);
    Write(' la comunicacion ?');
    gotoxy(7,6);
    Write(' Verifique si el diskette');
    gotoxy(7,7);
    Write('tiene suficiente capacidad');
    REPEAT UNTIL keypressed;
    ventana(11);
    gotoxy(4,3);
    textcolor(14);
    Write('<Esc>');
    textcolor(15);
    WriteLn(' para retornar al menu');
    textcolor(14);
    gotoxy(4,5);
    Write('<Enter>');
    textcolor(15);
    WriteLn(' para comunicacion');
    i:=1;
    REPEAT
        IF keypressed THEN
            BEGIN
                Read(kbd,ch);
                IF (ch=#27) AND (not keypressed) THEN BEGIN
                    ok:=FALSE;
                    i:=2;
                END;
                IF (ch=#13) THEN BEGIN
                    ok:=TRUE;
                    i:=2;
                END;
            END;
        UNTIL i=2;
    IF ok THEN

```

```

BEGIN
    ventana(11);
    gotoxy(4,3);
    textcolor(14);
    WriteLn(' COMUNICACION REALIZANDOSE');
    WriteLn;
    textcolor(15);
    l:=8;
    IF length(filo)<8 THEN l:=length(filo)+1;
    FOR i:=1 TO 8 DO
        IF filo[i]='.' THEN l:=i;
        filo:=concat(copy(filo,1,l-1),'.dde');
        comunica(filo);
    END;
END;
11:BEGIN {** Opcion para mostrar datos a pantalla/impresora **}
    largo:=10;
    color:=5;
    ventana(12);
    gotoxy(8,3);
    Write(' Datos de Entrada');
    gotoxy(8,5);
    Write(' Datos de Salida ');
    gotoxy(8,7);
    Write(' Graficos');
    l:=3;
    ok:=FALSE;
    REPEAT
        gotoxy(8,1);
        j:=1;
        IF l=3 THEN
            BEGIN
                textbackground(15);
                textcolor(0);
                Write(' Datos de Entrada ');
            END;
        IF l=5 THEN
            BEGIN
                textbackground(15);
                textcolor(0);
                Write(' Datos de Salida ');
            END;
        IF l=7 THEN
            BEGIN
                textbackground(15);
                textcolor(0);
                Write(' Graficos ');
            END;
        cc:=FALSE;
        REPEAT
            IF keypressed THEN
                BEGIN
                    Read(kbd,ch);

```

```

IF (ch = #13) THEN
BEGIN
    ok:=TRUE;
    cc:=TRUE;
END;
IF (ch = #27) AND keypressed THEN
BEGIN
    Read(kbd,ch);
    IF (ch = 'H') THEN
    BEGIN
        l:=l-2;
        cc:=TRUE;
    END;
    IF (ch = 'P') THEN
    BEGIN
        l:=l+2;
        cc:=TRUE;
    END;
    IF l<3 THEN l:=7;
    IF l>7 THEN l:=3;
END;
END;
UNTIL cc;
textbackground(5);
textcolor(15);
gotoxy(8,j);
IF j=3 THEN
Write (' Datos de Entrada ');
IF j=5 THEN
Write (' Datos de Salida ');
IF j=7 THEN
Write (' Graficos ');
UNTIL ok;
kat:=1;
textbackground(15);
textcolor(0);
gotoxy(8,kat);
IF j=3 THEN
Write (' Datos de Entrada ');
IF j=5 THEN
Write (' Datos de Salida ');
IF j=7 THEN
Write (' Graficos ');
textcolor(15);
IF kat=3 THEN
BEGIN
    hal:=46;
    color:=6;
    largo:=8;
    ventana(18);
    gotoxy(10,3);
    Write(' Por Pantalla ');
    gotoxy(10,5);

```



BIBLIOTECA

```

Write(' Por Impresora ');
l:=3;
ok:=FALSE;
REPEAT
gotoxy(10,1);
j:=1;
IF l=3 THEN
BEGIN
    textbackground(15);
    textcolor(0);
    Write(' Por Pantalla ');
END;
IF l=5 THEN
BEGIN
    textbackground(15);
    textcolor(0);
    Write(' Por Impresora ');
END;
cc:=FALSE;
REPEAT
    IF keypressed THEN
    BEGIN
        Read(kbd,ch);
        IF (ch = #13) THEN
        BEGIN
            ok:=TRUE;
            cc:=TRUE;
        END;
        IF (ch = #27) AND keypressed THEN
        BEGIN
            Read(kbd,ch);
            IF (ch = 'H') THEN
            BEGIN
                l:=l-2;
                cc:=TRUE;
            END;
            IF (ch = 'P') THEN
            BEGIN
                l:=l+2;
                cc:=TRUE;
            END;
            IF l<3 THEN l:=5;
            IF l>5 THEN l:=3;
        END;
    END;
UNTIL cc;
textbackground(6);
textcolor(15);
gotoxy(10,j);
IF j=3 THEN
Write(' Por Pantalla ');
IF j=5 THEN
Write(' Por Impresora ');

```

```

UNTIL ok;
IF j=3 THEN datosent(filo);
IF j=5 THEN datosentimp(filo);
END;
IF (kat=5) OR (kat=7) THEN
BEGIN
IF (kat=7) THEN
BEGIN
FOR i:=1 TO 20 DO
BEGIN
imp[i]:=gimp[i];
pan[i]:=gpan[i];
END;
END;
IF (kat=5) THEN
BEGIN
FOR i:=1 TO 20 DO
BEGIN
pan[i]:=span[i];
imp[i]:=simp[i];
END;
END;
hal:=46;
color:=6;
largo:=8;
ventana(18);
gotoxy(10,3);
Write(' Por Pantalla ');
gotoxy(10,5);
Write(' Por Impresora ');
l:=3;
ok:=FALSE;
REPEAT
gotoxy(10,l);
j:=1;
IF l=3 THEN
BEGIN
textbackground(15);
textcolor(0);
Write(' Por Pantalla ');
END;
IF l=5 THEN
BEGIN
textbackground(15);
textcolor(0);
Write(' Por Impresora ');
END;
cc:=FALSE;
REPEAT
IF keypressed THEN
BEGIN
Read(kbd,ch);
IF (ch = #13) THEN

```



```

BEGIN
    ok:=TRUE;
    cc:=TRUE;
END;
IF (ch = #27) AND keypressed THEN
BEGIN
    Read(kbd,ch);
    IF (ch = 'H') THEN
    BEGIN
        l:=l-2;
        cc:=TRUE;
    END;
    IF (ch = 'P') THEN
    BEGIN
        l:=l+2;
        cc:=TRUE;
    END;
    IF l<3 THEN l:=5;
    IF l>5 THEN l:=3;
    END;
END;
UNTIL cc;
textbackground(6);
textcolor(15);
gotoxy(10,j);
IF j=3 THEN
BEGIN
    imprime:=FALSE;
    Write (' Por Pantalla ');
END;
IF j=5 THEN
BEGIN
    imprime:=TRUE;
    Write (' Por Impresora ');
END;
UNTIL ok;
sat:=1;
textbackground(15);
textcolor(0);
graf:=0;
gotoxy(10,sat);
IF sat=3 THEN
Write (' Por Pantalla ');
IF sat=5 THEN
Write (' Por Impresora ');
IF sat=3 THEN
BEGIN
    gotoxy(8,79);
    textbackground(3);
    window(41,1,79,21);
    FOR i:=1 TO 6 DO
        WriteLn(' ');
    WriteLn;

```

```

WriteLn;
WriteLn;
WriteLn;
largo:=6;
color:=4;
hal:=45;
ventana(18);
gotoxy(6,2);
l11:=0;
k:=0;
Write('SALIDAS           Salir');
gotoxy(3,4);
FOR i:=1 TO nint DO
BEGIN
  IF sal[i]='S' THEN
  BEGIN
    textbackground(4);
    textcolor(15);
    Write(' ');
    IF pan[l11+1]=TRUE THEN
    BEGIN
      textbackground(15);
      textcolor(0);
      graf:=graf+1;
    END;
    IF i<10 THEN Write(' ');
    Write(i);
    l11:=l11+1;
    vasa[l11]:=i;
    IF l11=10 THEN gotoxy(3,5);
    IF vasa[l11]<10 THEN kkk:=l11;
    IF (l11>=11) AND (vasa[l11]<10) THEN k:=l11-10;
    END;
  END;
  l:=4;
  ok:=FALSE;
  REPEAT
    textbackground(4);
    textcolor(15);
    IF (pan[round((l-4)/3)+1]=TRUE) THEN
    BEGIN
      textbackground(15);
      textcolor(0);
    END;
    gotoxy(6,3);
    Write(' ');
    gotoxy(7,3);
    i:=vasa[round((l-4)/3)+1];
    IF l<4+(l11*3) THEN Write(nom[i]);
    gotoxy(l,4);
    IF (l>=34) THEN gotoxy(4+(l-34),5);
    IF vasa[round((l-4)/3)+1]<10 THEN Write(' ');
    IF l=4+l11*3 THEN gotoxy(22,2);
  
```



BIBLIOTECA

```

j:=1;
cc:=FALSE;
REPEAT
  IF keypressed THEN
    BEGIN
      Read(kbd,ch);
      IF (ch = #27) AND keypressed THEN
        BEGIN
          Read(kbd,ch);
          IF (ch = 'K') THEN
            BEGIN
              l:=l-3;
              cc:=TRUE;
            END;
          IF (ch = 'M') THEN
            BEGIN
              l:=l+3;
              cc:=TRUE;
            END;
          IF l<4 THEN l:=4+(l*3);
          IF l>4+(l*3) THEN l:=4;
        END;
      IF (ch = #13) THEN
        BEGIN
          IF l=l*3+4 THEN ok:=TRUE
          ELSE
            BEGIN
              s:=round((l-4)/3)+1;
              IF ((pan[s]=FALSE) AND
                ((graf<3) OR (kat<>7))) THEN
                BEGIN
                  pan[s]:=TRUE;
                  graf:=graf+1;
                  textbackground(15);
                  textcolor(0);
                END
              ELSE
                BEGIN
                  IF ( pan[s]) THEN graf:=graf-1;
                  pan[s]:=FALSE;
                  textbackground(4);
                  textcolor(15);
                END;
              gotoxy(1,4);
              IF (l>=34) THEN gotoxy(4+(l-34),5);
              IF vasa[s]<10 THEN Write(' ');
              Write(vasa[s]);
            END;
          cc:=TRUE;
        END;
      END;
    END;
  UNTIL cc;
UNTIL ok;

```

```

END;
IF sat=5 THEN
BEGIN
    graf:=0;
    gotoxy(8,79);
    textbackground(3);
    window(41,1,79,21);
    FOR i:=1 TO 6 DO
    WriteLn(' ');
    WriteLn;
    WriteLn;
    WriteLn;
    WriteLn;
    WriteLn;
    largo:=6;
    color:=4;
    hal:=45;
    ventana(18);
    gotoxy(6,2);
    lll:=0;
    k:=0;
    Write('SALIDAS           Salir');
    gotoxy(3,4);
    FOR i:=1 TO nint DO
    BEGIN
        IF sal[i]='S' THEN
        BEGIN
            textbackground(4);
            textcolor(15);
            Write(' ');
            IF imp[lll+1]=TRUE THEN
            BEGIN
                textbackground(15);
                textcolor(0);
                graf:=graf+1;
            END;
            IF i<10 THEN Write(' ');
            Write(i);
            lll:=lll+1;
            vasa[lll]:=i;
            IF lll=10 THEN gotoxy(3,5);
            IF vasa[lll]<10 THEN kkk:=lll;
            IF (lll>=11) AND (vasa[lll]<10) THEN k:=lll-10;
            END;
        END;
        l:=4;
        ok:=FALSE;
        REPEAT
            textbackground(4);
            textcolor(15);
            IF (imp[round((l-4)/3)+1]=TRUE) THEN
            BEGIN
                textbackground(15);
                textcolor(0);

```



```

END;
gotoxy(6,3);
Write(' ');
gotoxy(7,3);
i:=vasa[round((l-4)/3)+1];
IF l<4+(l11*3) THEN Write(nom[i]);
gotoxy(1,4);
IF (l>=34) THEN gotoxy(4+(l-34),5);
IF i<10 THEN Write(' ');
IF l=4+l11*3 THEN gotoxy(22,2);
j:=1;
cc:=FALSE;
REPEAT
  IF keypressed THEN
    BEGIN
      Read(kbd,ch);
      IF (ch = #27) AND keypressed THEN
        BEGIN
          Read(kbd,ch);
          IF (ch = 'K') THEN
            BEGIN
              l:=l-3;
              cc:=TRUE;
            END;
          IF (ch = 'M') THEN
            BEGIN
              l:=l+3;
              cc:=TRUE;
            END;
          IF l<4 THEN l:=4+(l11*3);
          IF l>4+(l11*3) THEN l:=4;
        END;
      IF (ch = #13) THEN
        BEGIN
          IF l=l11*3+4 THEN ok:=TRUE
          ELSE
            BEGIN
              s:=round((l-4)/3)+1;
              IF ((imp[s]=FALSE) AND
                ((graf<3) OR (kat<>7))) THEN
                BEGIN
                  imp[s]:=TRUE;
                  graf:=graf+1;
                  textbackground(15);
                  textcolor(0);
                END
              ELSE
                BEGIN
                  IF (imp[s]) THEN graf:=graf-1;
                  imp[s]:=FALSE;
                  textbackground(4);
                  textcolor(15);
                END;
            END;
          END;
        END;
      END;
    END;
  END;

```

```

                                gotoxy(1,4);
                                IF (l>=34) THEN gotoxy(4+(l-34),5);
                                IF vasa[s]<10 THEN Write(' ');
                                Write(vasa[s]);
                                END;
                                cc:=TRUE;
                                END;
                                END;
                                UNTIL cc;
                                UNTIL ok;
                                END;
END;
IF (kat=5) AND (sat=3) THEN
BEGIN
    i:=0;
    REPEAT
        i:=i+1;
        UNTIL filo[i]='.';
        sttr:=(concat(copy(filo,1,i-1),'.DDS'));
        Write(sttr);
        listar(sttr);
    END;
    IF (kat=5) AND (sat=5) THEN
    BEGIN
        i:=0;
        REPEAT
            i:=i+1;
            UNTIL filo[i]='.';
            sttr:=(concat(copy(filo,1,i-1),'.DDS'));

            Write(sttr);
            REPEAT UNTIL keypressed;
            listarimp(sttr);
            Write('hola');
        END;
        IF (kat=7) AND (sat=5) THEN
        BEGIN
            FOR i:=1 TO 20 DO
                gimp[i]:=imp[i];
            END;
            IF (kat=7) AND (sat=3) THEN
            BEGIN
                FOR i:=1 TO 20 DO
                    gpan[i]:=pan[i];
                END;
                IF (kat=5) AND (sat=5) THEN
                BEGIN
                    FOR i:=1 TO 20 DO
                        simp[i]:=imp[i];
                    END;
                    IF (kat=5) AND (sat=3) THEN
                    BEGIN
                        FOR i:=1 TO 20 DO

```



```

        span[i]:=pan[i];
    END;
    IF ((graf<>0) AND ( kat=7)) THEN
    BEGIN
        l:=8;
        IF length(filo)<8 THEN l:=length(filo)+1;
        FOR i:=1 TO 8 DO
            IF filo[i]='.' THEN l:=i;
        filo:=concat(copy(filo,1,l-1),'.dds');

        {** Se grafica los resultados de la comunicacion **}

        grafica(filo,nint,xst,xfn,cx[ii],cy[ii],2);
    END;
END;
14 : BEGIN {** Opciones del menu de ayuda **}
    menayu;
    sss:=2;
    ok1:=FALSE;
    REPEAT
        cursoayu(sss);
    CASE sss OF
        2: BEGIN {** Listar directorio **}
            window(2,12,78,22);
            textbackground(0);
            textcolor(15);
            disc2:=defdrive;
            kat:=0;
            leer;
            setdrive(disc2);
        END;
        3: BEGIN {** Cambiar de directorio **}
            mdirectorio;
            disc2:= DefDrive;
            kkk:=6;
            disc1:=disco;
            IF disc1=A THEN kkk:=4;
            IF disc1=B THEN kkk:=5;
            IF disc1=C THEN kkk:=6;
            cursodir(kkk);
            IF kkk=4 THEN disco:=A;
            IF kkk=5 THEN disco:=B;
            IF kkk=6 THEN disco:=C;
            setdrive(disc2);
            msistema;
        END;
        4: BEGIN {** Cambiar base de impresion **}
            mbase;
            IF base='TIEMPO' THEN kkk:=5;
            IF base<>'TIEMPO' THEN kkk:=6;
            cursobase(kkk);
            IF kkk=5 THEN base:='TIEMPO';
            IF kkk=6 THEN base:=base1;

```

```

        msistema;
    END;
5: BEGIN {** Borrar archivos escogidos **}
    window(1,1,79,25);
    textbackground(15);
    textcolor(0);
    gotoxy(8,24);
    Write('< ENTER >   Marcar Archivo           < B >   Borrar');
    window(2,12,79,23);
    textbackground(0);
    textcolor(15);
    disc2:=defdrive;
    borrar;
    window(1,1,79,25);
    textbackground(15);
    textcolor(0);
    gotoxy(8,24);
    Write('');
    setdrive(disc2);
    END;
    6: ok1:=TRUE
    END;
    UNTIL ok1;
    END;
17: BEGIN
    window(1,1,80,25);
    textbackground(0);
    textcolor(14);
    clrscr;
    END;
    END;
    UNTIL accion=17 ;
END.

```

```
(*****
(* Nos deja en Segment y en Offset el segmento y el offset de la DTA.      *)
(*****)
```

```
PROCEDURE getdta(VAR segment,offset:INTEGER);
```

```
TYPE
```

```
    resulta = record
        AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
    END;
```

```
VAR
```

```
    parms:resulta;
```

```
BEGIN
```

```
    parms.AX:=$2F00;
    msdos(parms);
    segment:=parms.ES;
    offset:=parms.BX;;
```

```
END;
```

```
(*****
(*      Posiciona la DTA en la direccion segment:Offset.                  *)
(*****)
```

```
PROCEDURE setdta(segment,offset:INTEGER);
```

```
TYPE
```

```
    resulta = record
        AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
    END;
```

```
VAR
```

```
    parms:resulta;
```

```
BEGIN
```

```
    parms.AX:=$1A00;
    parms.DS:=segment;
    parms.DX:=offset;
    msdos(parms);
```

```
END;
```

```
(*****
(* Este procedimiento permite obtener el primer archivo de la secuencia de *)
(* peticion de directorio del DOS, para ello usa la interrupcion 21-4EH.   *)
(* Se valida los casos de no encontrar archivo o que no existen mas archi- *)
(* vos correspondientes a la mascara pasada a traves de "path"             *)
(*****)
```

```
FUNCTION firstentry(path:pathname;option:INTEGER):entry;
```

```
VAR
```

```

    parms:resulta;
    entry_buffer:entry;
    seg_dta,ofs_dta,saved_segment,saved_offset,i:INTEGER;
BEGIN
    entry_buffer:=^
    fillchar(dta,sizeof(dta),0);
    fillchar(mask,sizeof(mask),0);
    fillchar(entry_buffer,sizeof(entry_buffer),0);
    getdta(saved_segment,saved_offset);
    seg_dta:=seg(dta);
    ofs_dta:=ofs(dta);
    setdta(seg_dta,ofs_dta);
    FOR i:=1 TO length(path) DO
        mask[i]:=path[i];
    parms.AX:=$4E00;
    parms.CX:=option;
    parms.DS:=seg(mask);
    parms.DX:=ofs(mask);
    msdos(parms);
    IF odd(parms.flags) THEN
        CASE lo(parms.AX) OF
            2: firstentry := 'No such file';
            18: firstentry:= 'No more file';
        END
    ELSE
        BEGIN
            i:=1;
            REPEAT
                entry_buffer[i]:=char(mem[seg(dta):ofs(dta)+29+i]);
                i:=succ(i);
            UNTIL (i>13);
            entry_buffer[0]:=char(pred(i));
            firstentry:=entry_buffer;
        END;
        setdta(saved_segment,saved_offset);
    END;

```

```

(*****)
(* Esta funcion se usa junto al anterior cuando se desea obtener el *)
(* siguiente archivo de la secuencia de peticion. *)
(*****)

```

```

FUNCTION nextentry(option:INTEGER):entry;
TYPE
    resulta = record
        AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
    END;

VAR
    parms:resulta;
    entry_buffer:entry;
    seg_dta,ofs_dta,saved_segment,saved_offset,i:INTEGER;

```

```

BEGIN
    entry_buffer:=^;
    fillchar(entry_buffer,sizeof(entry_buffer),0);
    getdta(saved_segment,saved_offset);
    seg_dta:=seg(dta);
    ofs_dta:=ofs(dta);
    setdta(seg_dta,ofs_dta);
    parms.AX:=$4F00;
    parms.CX:=option;
    msdos(parms);
    IF odd(parms.flags) AND ( lo(parms.AX) = 18)
        THEN
            nextentry:= 'no more file'
        ELSE
            IF not (odd(parms.flags)) THEN
                BEGIN
                    i:=1;
                    REPEAT
                        entry_buffer[i]:= char(mem[seg(dta):ofs(dta)+29+i]);
                        i:=succ(i);
                    UNTIL{ not(entry_buffer[i-1] in [' '.."''])} (i>13);
                    entry_buffer[0]:=char(pred(i));
                    nextentry:=concat(entry_buffer,' ');
                END;
                setdta(saved_segment,saved_offset);
            END;

    (*****
    (* Esta funcion borra un archivo del directorio actual. Retornando la      *)
    (* palabra "borrado" cuando ha sido exitosa la operacion o sino el mensa- *)
    (* de error correspondiente.                                              *)
    (*****

FUNCTION borraentry(path:pathname;option:INTEGER):entry;

VAR
    parms:resulta;
    entry_buffer:entry;
    seg_dta,ofs_dta,saved_segment,saved_offset,i:INTEGER;
BEGIN
    entry_buffer:=^;
    fillchar(dta,sizeof(dta),0);
    fillchar(mask,sizeof(mask),0);
    fillchar(entry_buffer,sizeof(entry_buffer),0);
    getdta(saved_segment,saved_offset);
    seg_dta:=seg(dta);
    ofs_dta:=ofs(dta);
    setdta(seg_dta,ofs_dta);
    FOR i:=1 TO length(path) DO
        mask[i]:=path[i];
    parms.AX:=$4E00;
    parms.CX:=option;
    parms.DS:=seg(mask);

```



```
parms.DX:=ofs(mask);
msdos(parms);
IF odd(parms.flags) THEN
CASE lo(parms.AX) OF
    2: borraentry := 'No such file';
    18: borraentry:= 'No more file';
    5: borraentry:= 'No Permitido';
END
ELSE
    BEGIN
        parms.AX:=$4100;           {Funcion para borrar (64)}
        msdos(parms);
        borraentry := 'Borrado    ';
        END;
    setdta(saved_segment,saved_offset);
END;
```




```
(*****)
(* Muestra la ventana de status del sistema: Memoria libre, disco actual y *)
(* la base de impresion. *)
(*****)
```

```
OVERLAY PROCEDURE sistema;
```

```
BEGIN
```

```
    window(46,2,79,12);
    textbackground(1);
    textcolor(15);
    gotoxy(18,3);
    disc1:=disco;
    numr:=DiskSpace(disc1);
    Write(numr:10:2, ' Kb');
    gotoxy(22,5);
    IF disc1=A THEN Write('A>'); { Imprime el directorio }
    IF disc1=B THEN Write('B>');
    IF disc1=C THEN Write('C>');
    gotoxy(22,7);
    Write(base);
                                { Imprime la variable base }
```

```
END;
```

```
(*****)
(* Este modulo sensa el disco de trabajo que desea seleccionar el usuario *)
(*****)
```

```
OVERLAY PROCEDURE cursodir(VAR a:INTEGER);
```

```
VAR
```

```
    ch:CHAR;
    cc,o:BOOLEAN;
```

```
BEGIN
```

```
    o:=FALSE;
    l:=a;
    REPEAT
        window(46,2,79,10);
        gotoxy(18,1);
        j:=1;
        IF l=4 THEN
            BEGIN
                textbackground(0);
                textcolor(15);
                Write(' < A > ');
                window(46,2,79,10);
            END;
        IF l=5 THEN
            BEGIN
                textbackground(0);
                textcolor(15);
                Write(' < B > ');
                window(46,2,79,10);
            END;
        IF l=6 THEN
```

```

BEGIN
    textbackground(0);
    textcolor(15);
    Write(' < C > ');
    window(46,2,79,10);
END;

gotoxy(18,1);
cc:=FALSE;
REPEAT
    IF keypressed THEN
        BEGIN
            Read(kbd,ch);
            IF (ch = #13) THEN
                BEGIN
                    o:=TRUE;
                    cc:=TRUE;
                END;
            IF (ch = #27) AND keypressed THEN
                BEGIN
                    Read(kbd,ch);
                    IF (ch = 'H') THEN
                        BEGIN
                            l:=l-1;
                            cc:=TRUE;
                        END;
                    IF (ch = 'P') THEN
                        BEGIN;
                            l:=l+1;
                            cc:=TRUE;
                        END;
                    IF l<4 THEN l:=6;
                    IF l>6 THEN l:=4;
                END;
            END;
        UNTIL cc;
        window(46,2,79,10);
        gotoxy(18,j);
        IF (j=4) AND (NOT o) THEN
            BEGIN
                textbackground(11);
                textcolor(15);
                Write(' < A > ');
            END;
        IF (j=5) AND (NOT o) THEN
            BEGIN
                textbackground(11);
                textcolor(15);
                Write(' < B > ');
            END;
        IF (j=6) AND (NOT o) THEN
            BEGIN
                textbackground(11);

```

```

        textcolor(15);
        Write(' < C > ');
    END;

    UNTIL o;
    a:=1;

    END;

(*****)
(* Este modulo dibuja el marco en donde se muestra los disco disponibles *)
(* para ser seleccionados por el usuario (A,B,C) *)
(*****)

OVERLAY PROCEDURE mdirectorio;
BEGIN
    textbackground(11);
    textcolor(15);
    window(46,2,79,10);
    clrscr;
    window(46,2,79,12);
    gotoxy(2,1);
    Write('^',copy(lin1,1,30),'^');
    gotoxy(2,2);
    Write('^| ^');
    gotoxy(2,3);
    Write('^ DISCO : | ^');
    gotoxy(12,3);
    gotoxy(2,4);
    Write('^ < A > | ^');
    gotoxy(2,5);
    Write('^ < B > | ^');
    gotoxy(2,6);
    Write('^ < C > | ^');
    gotoxy(2,7);
    Write('^ | ^');
    gotoxy(2,8);
    Write('^ | ^');
    gotoxy(2,9);
    Write('^',copy(lin1,1,30),'^');
END;

(*****)
(* Este modulo permite seleccionar la variable independiente por defecto que*)
(* es el TIEMPO y ademas permite ingresar otra variable por teclado, en cuyo*)
(* caso esa seria la que actuaria en el resto de la ejecucion del DDD.PAS *)
(*****)

OVERLAY PROCEDURE cursobase(VAR a:INTEGER);
VAR
    ch:CHAR;
    c,o:BOOLEAN;
BEGIN

```

```

o:=FALSE;
l:=a;
REPEAT
window(46,2,79,10);
gotoxy(15,1);
j:=1;
IF l=5 THEN
BEGIN
textbackground(0);
textcolor(15);
Write(' - TIEMPO ');
window(46,2,79,10);
END;
IF l=6 THEN
BEGIN
textbackground(0);
textcolor(15);
Write(' ');
gotoxy(15,1);
Write(' - ');
Read(base1);
window(46,2,79,10);
END;

gotoxy(15,1);
c:=FALSE;
REPEAT
IF keypressed THEN
BEGIN
Read(kbd,ch);
IF (ch = #13) THEN
BEGIN
o:=TRUE;
c:=TRUE;
END;
IF (ch = #27) AND keypressed THEN
BEGIN
Read(kbd,ch);
IF (ch = 'H') THEN
BEGIN
l:=l-1;
c:=TRUE;
END;
IF (ch = 'P') THEN
BEGIN;
l:=l+1;
c:=TRUE;
END;
IF l<5 THEN l:=6;
IF l>6 THEN l:=5;
END;
END;
UNTIL c;

```

```

window(46,2,79,10);
gotoxy(15,j);
IF (j=5) AND (NOT o) THEN
BEGIN
    textbackground(11);
    textcolor(15);
    Write(' - TIEMPO ');
END;
IF (j=6) AND (NOT o) THEN
BEGIN
    textbackground(11);
    textcolor(15);
    Write(' ');
    gotoxy(15,j);
    Write(' - ');
    Write(base1);
END;
UNTIL o;
a:=1;
END;

```

```

(*****
(* Este modulo dibuja el marco de seleccion de variable independiente *)
(*****

```

```

OVERLAY PROCEDURE mbase;
BEGIN
    textbackground(11);
    textcolor(15);
    window(46,2,79,10);
    clrscr;
    window(46,2,79,12);
    gotoxy(2,1);
    Write('┌',copy(lin1,1,30),'┐');
    gotoxy(2,2);
    Write('│' );
    gotoxy(2,3);
    Write('│' VARIABLE INDEPENDIENTE : '│');
    gotoxy(12,3);
    gotoxy(2,4);
    Write('│' );
    gotoxy(2,5);
    Write('│' - TIEMPO '│');
    gotoxy(2,6);
    Write('│' - '│');
    gotoxy(18,6);
    Write(base1);
    gotoxy(2,7);
    Write('│' );
    gotoxy(2,8);
    Write('│' );
    gotoxy(2,9);
    Write('└',copy(lin1,1,30),'┘');

```



END;

```
(*****  
(* Este modulo obtiene el nombre de los archivos del directorio actual. *)  
(*****)
```

OVERLAY PROCEDURE leer;

BEGIN

clrscr;

kat:=0; {ultimo en pantalla}

z:=0; {total}

FOR i:=1 TO 80 DO

ddas[i]:='';

j:=1;

setdrive(disco);

ddas[1]:=firstentry('*. *', \$20);

IF (ddas[j]<>'No more file') OR

(ddas[j]<>'No such file') THEN

BEGIN

REPEAT

j:=j+1;

ddas[j]:=nextentry(\$4F);

UNTIL ddas[j]='no more file';

gotoxy(2,1);

l:=1;

k:=2;

z:=j-1;

kat:=z;

IF z>55 THEN kat:=55;

FOR i:=1 TO z DO

BEGIN

IF ddas[i]<>'no more file' THEN

BEGIN

lar:=length(ddas[i]);

l1l:=0;

REPEAT

l1l:=l1l+1;

UNTIL (ddas[i][l1l]='.') OR (ddas[i][l1l]=' ') OR (l1l=lar);

OR (l1l=lar);

IF ddas[i][l1l]='.' THEN l1l:=l1l+3;

ddas[i]:=copy(ddas[i],1,l1l);

END;

END;

FOR i:=1 TO kat DO

BEGIN

gotoxy(k,1);

IF ddas[i]<>'no more file' THEN

BEGIN

lar:=length(ddas[i]);

l1l:=0;

REPEAT

l1l:=l1l+1;

UNTIL (ddas[i][l1l]='.') OR (ddas[i][l1l]=' ') OR (l1l=lar);


```

                                OR (lll=lar);
                                IF ddas[i][lll]='.' THEN lll:=lll+3;
                                Write(copy(ddas[i],1,lll));
                                ddas[i]:=copy(ddas[i],1,lll);
                                END;
                                k:=k+15;
                                IF k>62 THEN
                                BEGIN
                                    l:=l+1;
                                    k:=2;
                                END;
                                END;
                                END;
                                END;

END;

(*****
(* LeeArchivo implementa la lectura de los datos de entrada desde el tecla- *)
(* do. Los datos de entrada que se leen son: Numero de Integradores      *)
(* (Nint), Numero de iteraciones (Nite), Valor inicial (Xst), Valor Final  *)
(* (Xfn) y el Paso de impresion (Prnt); disponibles en la pantalla DDASCR02. *)
(* Luego se leen el nombre del integrador (nom), sus conexiones dy1, dy2,  *)
(* dy3, dy4, dy5 y dx; se leen tantas veces como el numero de integradores *)
(* se fijaron en Nint. Luego se lee la salida que se desea (sal), factor de *)
(* escala (cx), P_bit (cy) y la condicion inicial (cinic); disponibles en *)
(* la pantalla DDASCR03. Este modulo permite avanzar una pantalla con      *)
(* ENTER y retornar a la pantalla anterior con ESC                          *)
(*****)

OVERLAY PROCEDURE leearchivo(infilename:workstring);
VAR
dato2,dato,primero,regresar,second,escape,aumenta:BOOLEAN;
ultimo:INTEGER;
BEGIN
    nite:=512;
    numero:='velocidad';
    escape:=FALSE;
    primero:=TRUE;
    dato:=FALSE;
    dato2:=FALSE;
    aumenta:=FALSE;
    regresar:=FALSE;
    FOR i:=1 TO 80 DO
    BEGIN
        dy1[i]:=0;
        dy2[i]:=0;
        dy3[i]:=0;
        dy4[i]:=0;
        dy5[i]:=0;
        dx[i]:=101;
        nom[i]:='';
        cx[i]:=0;
        cy[i]:=0;
        cinic[i]:=0;
    
```

```

sal[i]:='N';
END;
xfn:=0;
xst:=0;
prnt:=0;
nite:=0;
nint:=0;
REPEAT
IF (NOT second) OR ( primero) THEN
BEGIN
    window(1,1,80,25);
    textbackground (9);
    textcolor(14);
    clrscr;
    textcolor(12);
    sttr:=concat('archivo : ',infilename);
    lar:=length(sttr);
    gotoxy((80-lar) div 2,1);
    Write(sttr);
    textcolor(14);
    WriteLn;
    window(3,2,78,7);
    textbackground(11);
    clrscr;
    window(3,2,80,25);
    gotoxy(5,24);
    textbackground(9);
    textcolor(15);
    Write(' <ENTER>=Continua    <ESC>=Regresar    <' ,chr(24),chr(26));
    Write(chr(25),chr(27),'>=Movimiento en Pantalla ');
    textcolor(14);
    textbackground(11);
    gotoxy(1,1);
    fr:='';
    dr:='';
    vr:='';
    fsttr:='';
    xsttr:='';
    vsttr:='';
    dsttr:='';
    FOR i:=1 TO 59 DO
    BEGIN
        fr:=concat(fr,chr(205));
        dr:=concat(dr,chr(196));
        vr:=concat(vr,' ');
    END;
    FOR i:=1 TO 4 DO
    BEGIN
        fsttr:=concat(fsttr,copy(fr,1,14),chr(209));
        xsttr:=concat(xsttr,copy(fr,1,14),chr(207));
        dsttr:=concat(dsttr,copy(dr,1,14),chr(197));
        vsttr:=concat(vsttr,copy(vr,1,14),chr(179));
    END;

```

```

fsttr:=concat(fsttr,copy(fr,1,14));
xsttr:=concat(xsttr,copy(fr,1,14));
dsttr:=concat(dsttr,copy(dr,1,14));
vsttr:=concat(chr(186),vsttr,copy(vr,1,14),chr(186));
WriteLn(concat(chr(201),fsttr,chr(187)));
textbackground(12);
sttr:=concat('  N',chr(163),'mero de ',chr(179),'  N',chr(163));
sttr:=concat(sttr,'mero de ',chr(179),'  Valor  ');
sttr:=concat(chr(186),sttr,chr(179),'  Valor  ',chr(179));
sttr:=concat(sttr,'  Paso de ',chr(186));
WriteLn(sttr);
sttr:=concat(' Integradores ',chr(179),'  Iteraciones ',chr(179));
sttr:=concat(sttr,'  Inicial  ');
sttr:=concat(chr(186),sttr,chr(179),'  Final  ',chr(179));
sttr:=concat(sttr,'  Impresi',chr(162),'n  ',chr(186));
WriteLn(sttr);
textbackground(11);
WriteLn(concat(chr(199),dsttr,chr(182)));
WriteLn(vsttr);
WriteLn(concat(chr(200),xsttr,chr(188)));
lar:=9;
window(1,1,80,25);
FOR i:=1 TO 80 DO
  d[i]:=0;
  IF escape AND ress THEN
    BEGIN
      reset(infile);
      Read(infile,archi);
      d[1]:=nint;
      d[2]:=nite;
      d[5]:=prnt;
    END;
  IF (NOT (escape) ) AND (ress) THEN
    BEGIN
      Read(infile,archi);
      k:=length(numero);
      gotoxy(9,6);
      nint2:=archi.segundo[1];
      d[1]:=nint2;
      nint:=nint2;
      nite:=archi.segundo[2];
      d[2]:=nite;
      xst:=archi.tercero[1];
      xfn:=archi.tercero[2];
      prnt:=archi.segundo[3];
      d[5]:=prnt;
    END;
  IF ress OR (escape) THEN
    BEGIN
      d[1]:=nint;
      gotoxy(9,6);
      Write(nint);
      gotoxy(24,6);
    END;

```



```

        END;
        IF lar>69 THEN
        BEGIN
            lar:=9;
            i:=1;
        END;
        IF lar=39 THEN lar:=35;
        IF lar=65 THEN lar:=69;
        IF lar=54 THEN lar:=50;
        IF lar=20 THEN lar:=24;
    END
    ELSE
    IF (ch=#27) THEN
    BEGIN
        regresar:=TRUE;
        ok:=TRUE;
        cc:=TRUE;
        END
    ELSE
    IF (ch<#44) OR (ch>#122) OR ((ch>#57)AND (ch<>#60)
        AND(ch<#65)) THEN
    BEGIN
        cc:=TRUE;
        lar:=1;
        i:=j;
        dsttr:='x';
    END
    ELSE
    BEGIN
        IF ressl=FALSE THEN
        BEGIN
            Write(' ');
            IF (i=3) OR (i=4) THEN Write (' ');
            ressl:=TRUE;
            gotoxy(lar,6);
        END;
        Write(ch);
        x:=x+1;
        IF (x>7) AND ((i<=2) OR (i=5)) THEN cc:=TRUE;
        IF (x>9) AND ((i=3) OR (i=4)) THEN cc:=TRUE;
        sttr:=concat(sttr,ch);
        dsttr:=sttr;
    END;
    END;
    END;
    UNTIL cc;
    IF (j<=2) OR (j=5) THEN val(dsttr,k,result);
    IF (j=3) OR (j=4) THEN
        val(dsttr,numr,result);
    IF (dsttr<>'') THEN
    BEGIN
        gotoxy(1,6);
        Write(' ');

```

```

        IF (j=3) OR (j=4) THEN Write(' ');
        gotoxy(1,6);
        IF (j=2) OR (j=5) THEN
        BEGIN
            IF (result=0) AND (k>=0) THEN d[j]:=k ;
            Write(d[j]);
        END;
        IF (j=1) THEN
        BEGIN
            IF (result=0) AND (k>=0) AND (k<80) THEN d[j]:=k ;
            Write(d[j]);
        END;
        IF (j=3) THEN
        BEGIN
            IF result=0 THEN xst:=numr;
            Write(xst:9:3);
        END;
        IF (j=4) THEN
        BEGIN
            IF result=0 THEN xfn:=numr;
            Write(xfn:9:3);
        END;
        END;
        UNTIL ok;
    END
ELSE
    BEGIN
        escape:=FALSE;
        d[1]:=nint;d[2]:=nite;d[5]:=prnt;END;
    IF NOT(regresar) THEN
    BEGIN
        textbackground(9);
        gotoxy(4,25);
        IF NOT second THEN Write
(
        textbackground(11);
        nint:=d[1];
        nite:=d[2];
        prnt:=d[5];
    IF nint>0 THEN
    BEGIN
        IF NOT second THEN aumenta:=FALSE;
        fsttr:=``;
        dsttr:=``;
        xsttr:=``;
        vsttr:=``;
        FOR i:=1 TO 6 DO
        BEGIN
            fsttr:=concat(fsttr,chr(209),copy(fr,1,6));
            dsttr:=concat(dsttr,chr(197),copy(dr,1,6));
            vsttr:=concat(vsttr,chr(179),copy(vr,1,6));
            xsttr:=concat(xsttr,chr(207),copy(fr,1,6));
        END;

```



```

fsttr:=concat(chr(201),copy(fr,1,6),chr(209),copy(fr,1,25),fsttr);
fsttr:=concat(fsttr,chr(187));
dsttr:=concat(chr(199),copy(dr,1,6),chr(197),copy(dr,1,25),dsttr);
dsttr:=concat(dsttr,chr(182));
vsttr:=concat(chr(186),copy(vr,1,6),chr(179),copy(vr,1,25),vsttr);
vsttr:=concat(vsttr,chr(186));
xsttr:=concat(chr(200),copy(fr,1,6),chr(207),copy(fr,1,25),xsttr);
xsttr:=concat(xsttr,chr(188));
sttr:=concat(' dy ',chr(179),' dy ',chr(179),' dy ',chr(179));
sttr:=concat(sttr,' dy ',chr(179),' dy ');
sl:=concat(' Int. ',chr(179),' Nombre de ',chr(179));
sttr:=concat(sl,sttr);
sttr:=concat(sttr,chr(179),' dx ');
window(3,9,78,12+(nint*2));
textbackground(11);
window(3,9,80,25);
gotoxy(1,1);
WriteLn(fsttr);
textbackground(12);
WriteLn(concat(chr(186),sttr,chr(186)));
sttr:=concat(' 1 ',chr(179),' 2 ',chr(179),' 3 ',chr(179));
sttr:=concat(sttr,' 4 ',chr(179),' 5 ');
sl:=concat(' # ',chr(179),' la Variable ',chr(179));
sttr:=concat(sl,sttr);
sttr:=concat(sttr,chr(179),' ');
WriteLn(concat(chr(186),sttr,chr(186)));
textbackground(11);
WriteLn(dsttr);
j:=1;
kkk:=1;
ultimo:=0;
k:=1;
escape:=FALSE;
REPEAT
REPEAT
  IF j+5>nint THEN
  BEGIN
    l:=j+5-nint;
    buffer:=l;
    FOR i:=1 TO l DO
    BEGIN
      gotoxy(1,5);
      textbackground(9);
      delline;
      delline;
      gotoxy(1,15-(i-1)*2);
      WriteLn;
      WriteLn;
      textbackground(11);
    END;
  END;
  i:=j;
  gotoxy(1,5);

```

```

str(j,numero);
IF j<9 THEN WriteLn(concat(copy(vsttr,1,3),numero,copy(vsttr,5,80)));
IF j>9 THEN WriteLn(concat(copy(vsttr,1,3),numero,copy(vsttr,6,80)));
WHILE (j+1<=nint) AND (j mod 6<>0) DO
BEGIN
  str(j+1,numero);
  WriteLn(dsttr);
  IF j<9 THEN
    WriteLn(concat(copy(vsttr,1,3),numero,copy(vsttr,5,80)));
  IF j>8 THEN
    WriteLn(concat(copy(vsttr,1,3),numero,copy(vsttr,6,80)));
  j:=j+1;
END;
i:=0;
IF (j mod 6<>0) THEN i:=6-(j mod 6);
IF escape=TRUE THEN aumenta:=TRUE;
j:=j+1+i;
IF (j+1>=nint) OR ((j-1) mod 6 =0) THEN WriteLn(xsttr);
IF (NOT dato) AND (NOT ress) THEN
BEGIN
  FOR i:=1 TO 6 DO
    d[i]:=0;
    d[6]:=101;
  END;
  lll:=1;
  REPEAT
  IF (NOT(aumenta)) AND (kkk<nint2+1) AND (ress) THEN
  BEGIN
    Read(infile,archi);
    nom[kkk]:=archi.primerio;
  END;
  IF aumenta OR second OR dato OR ((NOT aumenta) AND (kkk<nint2+1)
  AND ress) THEN
  BEGIN
    gotoxy(10,3+(lll*2));
    d[1]:=dy1[kkk];
    d[2]:=dy2[kkk];
    d[3]:=dy3[kkk];
    d[4]:=dy4[kkk];
    d[5]:=dy5[kkk];
    d[6]:=dx[kkk];
    Write(nom[kkk]);
    FOR i:=1 TO 6 DO
      BEGIN
        gotoxy(30+(i*7),3+(lll*2));
        IF (NOT aumenta) AND (kkk<nint2+1) AND (ress) THEN
          d[i]:=archi.segundo[i];
        IF dato OR escape OR aumenta OR (kkk<nint2+1) THEN
          Write(d[i]);
        END;
      END;
    lll:=lll+1;
    dy1[kkk]:=d[1];

```

```

dy2[kkk]:=d[2];
dy3[kkk]:=d[3];
dy4[kkk]:=d[4];
dy5[kkk]:=d[5];
dx[kkk]:=d[6];
kkk:=kkk+1;
UNTIL ((kkk-1) mod 6=0) OR(( kkk>nint2+1)AND(NOT(aumenta)) AND
      (NOT dato))) OR (kkk=nint+1);
kkk:=k+6;
IF (NOT escape) AND (kkk>nint2) THEN
FOR x:=1 TO 6 DO d[x]:=0;
IF kkk=ultimo THEN escape:=FALSE;
  gotoxy(5,17);
  textbackground(9);
  textcolor(15);
  Write(' <ENTER>=Continua      <ESC>=Regresar      <' ,
        chr(24),chr(26));
  Write(chr(25),chr(27),'>=Movimiento en Pantalla ');
  textcolor(14);
  textbackground(11);
i:=0 ;
y2:=k;
l1l:=5;
lar:=10;
ok:=FALSE;
REPEAT
  gotoxy(lar,l1l);
  cc:=FALSE;
  sttr:='';
  sl:='';
  l:=lar;
  sss:=l1l;
  s:=y2;
  y1:=i;
  x:=1;
  ressl:=FALSE;
  REPEAT
    IF keypressed THEN
      BEGIN
        Read(kbd,ch);
        IF (ch = #13) THEN
          BEGIN
            ok:=TRUE;
            cc:=TRUE;
            sttr:='';
          END
        ELSE
          BEGIN
            IF (ch = #27) AND keypressed THEN
              BEGIN
                Read(kbd,ch);
                sttr:='';
                IF (ch = 'K') THEN

```

```

BEGIN
    lar:=lar-7;
    i:=i-1;
    cc:=TRUE;
END;
IF (ch = 'M') THEN
BEGIN;
    lar:=lar+7;
    i:=i+1;
    cc:=TRUE;
END;
IF (ch = 'H') THEN
BEGIN
    lll:=lll-2;
    y2:=y2-1;
    cc:=TRUE;
END;
IF (ch = 'P') THEN
BEGIN;
    lll:=lll+2;
    y2:=y2+1;
    cc:=TRUE;
END;
IF lar<10 THEN
BEGIN
    i:=6;
    lar:=72;
END;
IF lar>72 THEN
BEGIN
    lar:=10;
    i:=0;
END;
IF lll<5 THEN
BEGIN
    y2:=k+5;
    IF y2>nint THEN
    BEGIN
        x1:=y2-nint;
        y2:=nint;
        lll:=15-(2*x1);
    END
    ELSE
        lll:=15;
END;
IF (lll>15) OR (y2>nint) THEN
BEGIN
    lll:=5;
    y2:=k;
END;
IF lar=17 THEN lar:=37;
IF lar=30 THEN lar:=10;
END

```

```

ELSE
  IF (ch = #27) THEN
    BEGIN
      k:=k-12;
      ultimo:=kkk;
      kkk:=k+6;
      j:=j-12;
      ok:=TRUE;
      cc:=TRUE;
      escape:=TRUE;
    END
  ELSE
    IF (ch<#44) OR (ch>#122) OR ((ch>#57)
      AND (ch<>#60) AND(ch<#65)) THEN
      BEGIN
        cc:=TRUE;
        lar:=1;
        i:=Y1;
        lll:=sss;
        s:=y2;
        s1:='false';
      END
    ELSE
      BEGIN
        IF ressl=FALSE THEN
          BEGIN
            Write(' ');
            IF (y1=0) THEN Write(' ');
            ressl:=TRUE;
            gotoxy(lar,sss);
          END;
          Write(ch);
          x:=x+1;
          IF (x=4) AND (i>0) THEN cc:=TRUE;
          IF (x=20) AND (i=0) THEN cc:=TRUE;
          sttr:=concat(sttr,ch);
          s1:=sttr;
        END;
      END;
    END;
  UNTIL cc;
  IF s1<>' ' THEN
    BEGIN
      gotoxy(1,sss);
      IF (y1=0) THEN Write(' ');
      IF (y1>0) THEN Write(' ');
      gotoxy(1,sss);
      IF (y1=0) AND (length(s1)<21) AND (s1<>' ') THEN
        BEGIN
          IF (lar=1) AND (sss=lll) AND (s1='false') THEN
            s1:=nom[s];
            nom[s]:=s1;
            Write(nom[s]);

```

```

END;
val(s1,z,result);
IF (y1=1) THEN
BEGIN
  IF (result=0) AND ((abs(z)<=nint) OR (z=101)) THEN
    val(s1,dy1[s],result);
    Write(dy1[s]);
END;
IF (y1=2) THEN
BEGIN
  IF (result=0) AND ((abs(z)<=nint) OR (z=101)) THEN
    val(s1,dy2[s],result);
    Write(dy2[s]);
END;
IF (y1=3) THEN
BEGIN
  IF (result=0) AND ((abs(z)<=nint) OR (z=101)) THEN
    val(s1,dy3[s],result);
    Write(dy3[s]);
END;
IF (y1=4) THEN
BEGIN
  IF (result=0) AND ((abs(z)<=nint) OR (z=101)) THEN
    val(s1,dy4[s],result);
    Write(dy4[s]);
END;
IF (y1=5) THEN
BEGIN
  IF (result=0) AND ((abs(z)<=nint) OR (z=101)) THEN
    val(s1,dy5[s],result);
    Write(dy5[s]);
END;
IF (y1=6) THEN
BEGIN
  IF (result=0) AND (z<>0) AND (abs(z)<=nint) THEN
    val(s1,dx[s],result);
  IF (result=0) AND ((z=0) OR (abs(z)>nint) ) THEN
    dx[s]:= 101;
    Write(dx[s]);
  END;
END;
UNTIL ok;
textbackground(9);
gotoxy(4,17);
Write
(
  textbackground(11);
  k:=k+6;
  gotoxy(1,5);
UNTIL (NOT(escape)) OR (j<0);
UNTIL (j>=nint+1) OR (j<0);
dato:=TRUE;
primero:=FALSE;
);

```



```

IF j<0 THEN primero:=TRUE;
IF j>0 THEN
BEGIN
escape:=FALSE;
IF NOT second THEN aumenta:=FALSE;
    textbackground(9);
    clrscr;
    textcolor(14);
    gotoxy(1,1);
    textbackground(11);
    fsttr:=concat(copy(fsttr,1,34),copy(fr,1,5),chr(209),
        copy(fr,1,11));
    fsttr:=concat(fsttr,chr(209),copy(fr,1,11),chr(209),
        copy(fr,1,11),chr(187));
    dsttr:=concat(copy(dsttr,1,34),copy(dr,1,5),chr(197),
        copy(dr,1,11));
    dsttr:=concat(dsttr,chr(197),copy(dr,1,11),chr(197),
        copy(dr,1,11),chr(182));
    xsttr:=concat(copy(xsttr,1,34),copy(fr,1,5),chr(207),
        copy(fr,1,11));
    xsttr:=concat(xsttr,chr(207),copy(fr,1,11),chr(207),
        copy(fr,1,11),chr(188));
    vsttr:=concat(copy(vsttr,1,34),copy(vr,1,5),chr(179),
        copy(vr,1,11));
    vsttr:=concat(vsttr,chr(179),copy(vr,1,11),chr(179),
        copy(vr,1,11),chr(186));
    WriteLn(fsttr);
    textbackground(12);
    sttr:=concat(' Factor ',chr(179),' p-bit pres',chr(179));
    sttr:=concat(sttr,' Condiciones');
    sl:=concat(' Int. ',chr(179),' Nombre de ');
    sttr:=concat(sl,chr(179),' Sale ',chr(179),sttr);
    WriteLn(concat(chr(186),sttr,chr(186)));
    sttr:=concat('escala n ',chr(179),' ',chr(179));
    sttr:=concat(sttr,' Iniciales ');
    sl:=concat(' # ',chr(179),' la Variable ');
    sttr:=concat(sl,chr(179),' ',chr(179),sttr);
    WriteLn(concat(chr(186),sttr,chr(186)));
    textbackground(11);
    WriteLn(dsttr);
    j:=1;
    sss:=0;
    k:=1;
    kkk:=1;
    IF (nint<nint2) AND (ress=TRUE) AND (NOT aumenta) THEN
    BEGIN
        i:=nint2-nint;
        FOR l:=1 TO i DO
            Read(infile,archi);
        END;
    REPEAT
    REPEAT
        l:=0;

```



BIBLIOTECA

```

111:=0;
  IF j+5>nint THEN
    BEGIN
      sss:=j+5-nint;
      FOR i:=1 TO sss DO
        BEGIN
          gotoxy(1,5);
          textbackground(9);
          delline;
          delline;
          gotoxy(1,15-(i-1)*2);
          WriteLn;
          WriteLn;
          textbackground(11);
        END;
      END;
      gotoxy(1,5);
      str(j,numero);
      lar:=length(nom[j]);
      IF j<9 THEN BEGIN
        sttr:=concat(copy(vsttr,1,3),numero,copy(vsttr,5,5),nom[j]);
        sttr:=concat(sttr,copy(vsttr,10+lar,80));
        WriteLn(sttr);
      END;
      IF j>9 THEN BEGIN
        sttr:=concat(copy(vsttr,1,3),numero,copy(vsttr,6,4),nom[j]);
        sttr:=concat(sttr,copy(vsttr,10+lar,80));
        WriteLn(sttr);
      END;
      WHILE (j+1<=nint) AND (j mod 6<>0) DO
        BEGIN
          str(j+1,numero);
          lar:=length(nom[j+1]);
          WriteLn(dsttr);
          IF j<9 THEN BEGIN
            sttr:=concat(copy(vsttr,1,3),numero,
                          copy(vsttr,5,5),nom[j+1]);
            sttr:=concat(sttr,copy(vsttr,10+lar,80));
            WriteLn(sttr);
          END;
          IF j>8 THEN BEGIN
            sttr:=concat(copy(vsttr,1,3),numero,
                          copy(vsttr,6,4),nom[j+1]);
            sttr:=concat(sttr,copy(vsttr,10+lar,80));
            WriteLn(sttr);
          END;
          j:=j+1;
        END;
      i:=0;
      IF (j mod 6<>0) THEN i:=6-(j mod 6);
      j:=j+1+i;
      IF (j+1 >= nint) OR ((j-1) mod 6=0) THEN WriteLn(xsttr);
      IF (NOT dato) AND res THEN

```



```

FOR i:=1 TO 80 DO
BEGIN
    sal[i]:='N';
    cx[i]:=0;
    cy[i]:=0;
    cinic[i]:=0;
    d[i]:=0;
END;
REPEAT
    lll:=lll+1;
    IF (NOT aumenta) AND (kkk<=nint2) AND (NOT dato2) AND res THEN
    BEGIN
        Read(infile,archi);
        sal[kkk]:=archi.primer[1];
        cx[kkk]:=archi.segundo[1];
        cy[kkk]:=archi.segundo[2];
        cinic[kkk]:=archi.tercero[1];
    END;
    d[1]:=cx[kkk];
    d[2]:=cy[kkk];
    numr:=cinic[kkk];
    s1:=sal[kkk];
    gotoxy(37,3+(lll*2));
    IF (dato2 OR ((kkk<=nint2) AND res)) THEN
    BEGIN
        Write(s1);
        FOR i:=1 TO 2 DO
        BEGIN
            gotoxy(44+(i-1)*12,3+(lll*2));
            Write(d[i]:5);
        END;
        gotoxy(66,3+(lll*2));
        Write(numr:9:3);
    END;
    kkk:=kkk+1;
    UNTIL (lll=6) OR ((kkk>=nint2+1) AND (NOT(aumenta))
        AND (NOT dato2)) OR (kkk=nint+1);
    kkk:=k+6;
    IF kkk=ultimo THEN escape:=FALSE;
    gotoxy(5,17);
    textbackground(9);
    textcolor(15);
    Write(' <ENTER>=Continua    <ESC>=Regresar    <^ ,chr(24),chr(26));
    Write(chr(25),chr(27),'>=Movimiento en Pantalla ^');
    textcolor(14);
    textbackground(11);
    i:=0 ;
    escape:=FALSE;
    y2:=k;
    lll:=5;
    lar:=37;
    ok:=FALSE;
REPEAT

```

```

gotoxy(lar, lll);
cc:=FALSE;
sstr:='';
sl:='';
l:=lar;
sss:=lll;
s:=y2;
yl:=i;
x:=1;
ressl:=FALSE;
REPEAT
  IF keypressed THEN
    BEGIN
      Read(kbd, ch);
      IF (ch = #13) THEN
        BEGIN
          ok:=TRUE;
          cc:=TRUE;
          sstr:='';
        END
      ELSE
        BEGIN
          IF (ch = #27) AND keypressed THEN
            BEGIN
              Read(kbd, ch);
              sstr:='';
              IF (ch = 'K') THEN
                BEGIN
                  lar:=lar-12;
                  i:=i-1;
                  cc:=TRUE;
                END;
              IF (ch = 'M') THEN
                BEGIN;
                  lar:=lar+12;
                  i:=i+1;
                  cc:=TRUE;
                END;
              IF (ch = 'H') THEN
                BEGIN
                  lll:=lll-2;
                  y2:=y2-1;
                  cc:=TRUE;
                END;
              IF (ch = 'P') THEN
                BEGIN;
                  lll:=lll+2;
                  y2:=y2+1;
                  cc:=TRUE;
                END;
              IF (lar<37) AND (lar<>32) THEN
                BEGIN
                  i:=3;

```

```

        lar:=66;
    END;
    IF (lar>66) AND (lar<>68) THEN
    BEGIN
        lar:=37;
        i:=0;
    END;
    IF l11<5 THEN
    BEGIN
        y2:=k+5;
        IF y2>nint THEN
        BEGIN
            x1:=y2-nint;
            y2:=nint;
            l11:=15-(2*x1);
        END
        ELSE
            l11:=15;
        END;
        IF (l11>15) OR (y2>nint) THEN
        BEGIN
            l11:=5;
            y2:=k;
        END;
        IF lar=49 THEN lar:=44;
        IF lar=68 THEN lar:=66;
        IF lar=32 THEN lar:=37;
        IF lar=54 THEN lar:=56;
    END
    ELSE
        IF (ch = #27) THEN
        BEGIN
            k:=k-12;
            ultimo:=kkk;
            kkk:=k+6;
            j:=j-12;
            ok:=TRUE;
            cc:=TRUE;
            escape:=TRUE;
            aumenta:=TRUE;
        END
        ELSE
            IF (ch<#44) OR (ch>#122) OR ((ch>#57)AND (ch<>#60)
                AND (ch<#65)) THEN
            BEGIN
                cc:=TRUE;
                lar:=1;
                i:=Y1;
                l11:=sss;
                s:=y2;
                s1:='x';
            END
            ELSE

```

```

BEGIN
  IF ressl=FALSE THEN
    BEGIN
      IF (i=0) THEN Write(' ')
      ELSE
        IF (i=3) THEN Write(' ')
        ELSE
          Write(' ');
          ressl:=TRUE;
          gotoxy(lar,sss);
        END;
      Write(ch);
      x:=x+1;
      IF (x=2) AND (i=0) THEN cc:=TRUE;
      IF (x=6) AND ((i=1) OR (i=2)) THEN cc:=TRUE;
      IF (x=10) AND (i=3) THEN cc:=TRUE;
      sttr:=concat(sttr,ch);
      sl:=sttr;
    END;
  END;
END;
UNTIL cc;
IF sl<>' ' THEN
  BEGIN
    gotoxy(1,sss);
    IF (y1=0) THEN Write(' ');
    IF (y1>0) THEN Write(' ');
    IF (y1=3) THEN Write(' ');
    gotoxy(1,sss);
    IF (y1=0) AND (s1='s') THEN s1:='S';
    IF (y1=0) AND (s1='n') THEN s1:='N';
    IF (y1=0) AND ((s1='S') OR (s1='N')) THEN sal[s]:=s1;
    IF y1=0 THEN Write(sal[s]);
    IF (y1=1) THEN
      BEGIN
        val(s1,z,result);
        IF (result=0) THEN val(s1,cx[s],result);
        Write(cx[s]:5);
      END;
    IF (y1=2) THEN
      BEGIN
        val(s1,z,result);
        IF (result=0) THEN val(s1,cy[s],result);
        Write(cy[s]:5);
      END;
    IF (y1=3) THEN
      BEGIN
        z:=length(s1);
        val(s1,numr,result);
        IF (result=0) THEN val(s1,cinic[s],result);
        IF cinic[s]>=1.0E+8 THEN z:=9;
        IF (cinic[s]>=1.0E+7) AND (cinic[s]<1.0E+8) THEN z:=8;
        IF (cinic[s]>=1.0E+6) AND (cinic[s]<1.0E+7) THEN z:=7;
      END;
    END;
  END;

```



```

        IF z<7 THEN Write(cinic[s]:9:3);
        IF z=7 THEN Write(cinic[s]:9:2);
        IF z=8 THEN Write(cinic[s]:9:1);
        IF z>8 THEN Write(cinic[s]:9:0);
    END;
    END;
    UNTIL ok;
    textbackground(9);
    gotoxy(4,17);
    Write
(
    textbackground(11);
    k:=k+6;
    gotoxy(1,5);
    UNTIL (NOT escape) OR (j<0);
    UNTIL (j>=nint+1) OR (j<0);
    second:=FALSE;
    dato2:=TRUE;
    IF j<0 THEN second:=TRUE;
END
ELSE
    second:=FALSE;END;END;
    UNTIL (NOT(escape)) OR regresar ;
END;
(*****
(* Este procedimiento escribe los datos de entrada en la impresora. Se le *)
(* pasa como parametro el nombre del archivo que contiene estos datos. El *)
(* formato del reporte se encuentra especificado en DDALST01. *)
(*****)

OVERLAY PROCEDURE datosentimp(filename:workstring);
VAR
    c1,c2,c3,c4:workstring;
    r,s,t,u,v:INTEGER;
BEGIN
    WriteLn(lst);
    Write(lst,'_____');
    WriteLn(lst,'_____');
    c2:=concat('Archivo : ',filename);
    WriteLn(lst,'_____,c2);
    c2:='DATOS DE ENTRADA';
    WriteLn(lst,'_____,c2);
    Write(lst,'_____');
    WriteLn(lst,'_____');
    WriteLn(lst);
    WriteLn(lst,'          Numero de integradores :    ',nint);
    WriteLn(lst,'          S_bit de precision      :    ',nint);
    WriteLn(lst,'          Paso de impresion       :    ',prnt);
    WriteLn(lst,'          Valor Inicial           :    ',xst:0:4);
    WriteLn(lst,'          Valor Final             :    ',xfn:0:4);
    WriteLn(lst);
    Write(lst,' Nombre del Integrador  dy1 dy2 dy3 dy4 dy5  dx   n   ');
    WriteLn (lst,'p      C.Inic      SALE');

```

```

Write(lst,'_____');
WriteLn(lst,'_____');
l:=14;
c3:=
FOR i:=1 TO nint DO
BEGIN
  str(dy1[i],c4);
  v:=length(nom[i]);
  c2:=concat(nom[i],copy(c3,1,22-v),c4);
  v:=length(c4);
  str(dy2[i],c4);
  c2:=concat(c2,copy(c3,1,4-v),c4);
  v:=length(c4);
  str(dy3[i],c4);
  c2:=concat(c2,copy(c3,1,4-v),c4);
  v:=length(c4);
  str(dy4[i],c4);
  c2:=concat(c2,copy(c3,1,4-v),c4);
  v:=length(c4);
  str(dy5[i],c4);
  c2:=concat(c2,copy(c3,1,4-v),c4);
  v:=length(c4);
  str(dx[i],c4);
  c2:=concat(c2,copy(c3,1,4-v),c4);
  v:=length(c4);
  str(cx[i],c4);
  c2:=concat(c2,copy(c3,1,4-v),c4);
  v:=length(c4);
  str(cY[i],c4);
  c2:=concat(c2,copy(c3,1,6-v),C4);
  v:=length(c4);
  str(cinic[i]:0:3,c4);
  c2:=concat(c2,copy(c3,1,6-v),c4);
  v:=length(c4);
  c2:=concat(c2,copy(c3,1,15-v),sal[i]);
  WriteLn(lst,' ',c2);
  l:=l+1;
  IF l=55 THEN
  BEGIN
    l:=1;
    FOR v:=1 TO 15 DO
      WriteLn(lst);
    END;
  END;
REPEAT
UNTIL keypressed;
END;

```



```

(*****
(* Este procedimiento implementa el desplazamiento del cursor de seleccion *)
(* en el menu de ayuda. El parametro "a" retorna el numero de ayuda esco- *)
(* gido por el usuario. *)
(*****

```



```
OVERLAY PROCEDURE cursoayu(VAR a:INTEGER);
VAR
  ch:CHAR;
  c,o:BOOLEAN;
BEGIN
  o:=FALSE;
  l:=a;
  REPEAT
    window(6,5,40,12);
    gotoxy(2,1);
    j:=1;
    IF l=2 THEN
      BEGIN
        textbackground(0);
        textcolor(15);
        Write(' - LISTADO DE ARCHIVOS ');
        window(6,5,40,12);
      END;
    IF l=3 THEN
      BEGIN
        textbackground(0);
        textcolor(15);
        Write(' - DISCO ');
        window(6,5,40,12);
      END;
    IF l=4 THEN
      BEGIN
        textbackground(0);
        textcolor(15);
        Write(' - CAMBIO DE BASE DE IMPRESION ');
        window(6,5,40,12);
      END;
    IF l=5 THEN
      BEGIN
        textbackground(0);
        textcolor(15);
        Write(' - BORRAR ARCHIVO ');
        window(6,5,40,22);
      END;
    IF l=6 THEN
      BEGIN
        textbackground(0);
        textcolor(15);
        Write(' - SALIDA AL MENU PRINCIPAL ');
        window(6,5,40,22);
      END;
    gotoxy(3,1);
    c:=FALSE;
    REPEAT
      IF keypressed THEN
        BEGIN
          Read(kbd,ch);
```

```

IF (ch = #13) THEN
BEGIN
    o:=TRUE;
    c:=TRUE;
END;
IF (ch = #27) AND keypressed THEN
BEGIN
    Read(kbd,ch);
    IF (ch = 'H') THEN
    BEGIN
        l:=l-1;
        c:=TRUE;
    END;
    IF (ch = 'P') THEN
    BEGIN;
        l:=l+1;
        c:=TRUE;
    END;
    IF l<2 THEN l:=6;
    IF l>6 THEN l:=2;
    END;
END;
UNTIL c;
window(6,5,40,12);
gotoxy(2,j);
IF (j=2) AND (NOT o) THEN
BEGIN
    textbackground(15);
    textcolor(1);
    Write(' - LISTADO DE ARCHIVOS ');
END;
IF (j=3) AND (NOT o) THEN
BEGIN
    textbackground(15);
    textcolor(1);
    Write(' - DISCO ');
END;
IF (j=4) AND (NOT o) THEN
BEGIN
    textbackground(15);
    textcolor(1);
    Write(' - CAMBIO DE BASE DE IMPRESION ');
END;
IF (j=5) AND (NOT o) THEN
BEGIN
    textbackground(15);
    textcolor(1);
    Write(' - BORRAR ARCHIVO ');
END;
IF (j=6) AND (NOT o) THEN
BEGIN
    textbackground(15);
    textcolor(1);

```

```

        Write(' - SALIDA AL MENU PRINCIPAL ');
    END;
    UNTIL o;
    a:=1;
    END;

```

```

(*****
(* Este procedimiento dibuja el marco del status del sistema del computa- *)
(* dor correspondiente a la pantalla DDASCR07. *)
(*****

```

```

OVERLAY PROCEDURE msistema;
BEGIN

```

```

    textbackground(1);
    textcolor(15);
    window(46,2,79,10);
    clrscr;
    window(46,2,79,12);
    gotoxy(2,1);
    Write(' ',copy(lin1,1,30),' ');
    gotoxy(2,2);
    Write(' | ');
    gotoxy(2,3);
    Write(' | Mem. Libre : ');
    gotoxy(2,4);
    Write(' | ');
    gotoxy(2,5);
    Write(' | Disco : ');
    gotoxy(2,6);
    Write(' | ');
    gotoxy(2,7);
    Write(' | Var. Base : ');
    gotoxy(2,8);
    Write(' | ');
    gotoxy(2,9);
    Write(' ',copy(lin1,1,30),' ');
    sistema;

```

```

END;

```

```

(*****
(* Este procedimiento dibuja la pantalla general cuya especificacion de di- *)
(* seño se encuentra en DDASCR07. *)
(*****

```

```

OVERLAY PROCEDURE menayu;

```

```

    BEGIN
    window(1,1,80,25);
    textbackground(15);
    textcolor(0);
    clrscr;
    gotoxy(70,23);
    Write(' ');
    gotoxy(70,24);

```



```

Write(' | DDA | ');
gotoxy(70,25);
Write(' | ');

window(2,12,78,22);
textbackground(0);
clrscr;
lin1:='-';
lin2:='=';
FOR i:=1 TO 80 DO
BEGIN
    lin1:=concat(lin1,'-');
    lin2:=concat(lin2,'=');
END;
textcolor(1);
textbackground(15);
window(1,1,80,25);
gotoxy(1,1);
Write(' | ',copy(lin1,1,42),' | ');
gotoxy(1,2);
Write(' | ');
gotoxy(1,3);
Write(' | ');
gotoxy(1,4);
Write(' | ');
gotoxy(1,5);
Write(' | ');
gotoxy(1,6);
Write(' | ');
gotoxy(1,7);
Write(' | ');
gotoxy(1,8);
Write(' | ');
gotoxy(1,9);
Write(' | ');
gotoxy(1,10);
Write(' | ');
gotoxy(1,11);
Write(' | ',copy(lin1,1,42),' | ');
textcolor(15);
window(10,2,36,4);
textbackground(1);
clrscr;
gotoxy(2,1);
Write(' | ',copy(lin1,1,23),' | ');
gotoxy(2,2);
Write(' | MENU | ');
gotoxy(2,3);
Write(' | ',copy(lin1,1,23),' | ');
window(6,5,40,12);
textbackground(15);
textcolor(1);
gotoxy(3,2);

```

```

Write('- LISTADO DE ARCHIVOS      ');
gotoxy(3,3);
Write('- DISCO                      ');
gotoxy(3,4);
Write('- CAMBIO DE BASE DE IMPRESION ');
gotoxy(3,5);
Write('- BORRAR ARCHIVO              ');
gotoxy(3,6);
Write('- SALIDA AL MENU PRINCIPAL    ');
msistema;
END;

(*****)
(* Este procedimiento permite borrar todos los archivos marcados en la op- *)
(* cion de borrar archivos del menu de ayuda. Borrar los archivos del *)
(* directorio actual. *)
(*****)

OVERLAY PROCEDURE borrar;
BEGIN
    ress1:=FALSE;
    l:=1;
    k:=2;
    ii:=1;
    i:=0;
    ress:=FALSE;
    FOR ii:= 1 TO 80 DO
        borra[ii]:='.';
    ii:=1;
    y1:=0;
    REPEAT
        ok:=FALSE;
        REPEAT
            cc:=FALSE;
            textbackground(11);
            textcolor(15);
            gotoxy(k,1);
            sss:=1;
            ress:=FALSE;
            REPEAT
                IF (ddas[ii]=borra[sss]) THEN ress:=TRUE;
                sss:=sss+1
            UNTIL (sss>y1) OR ress;
            IF ress THEN
                BEGIN
                    textbackground(1);
                    textcolor(15);
                END;
            IF NOT ress THEN
                WriteLn(copy(concat(ddas[ii],',',1,14)));
            x2:=ii;
            gotoxy(k,1);
            lll:=1;

```



```

x1:=k;
REPEAT
  IF keypressed THEN
    BEGIN
      Read(kbd,ch);
      IF (ch = #13) THEN
        BEGIN
          ok:=TRUE;
          cc:=TRUE;
        END;
      IF (ch = 'b') OR (ch = 'B') THEN
        BEGIN
          ok:=TRUE;
          cc:=TRUE;
          res1:=TRUE;
          i:=-199;
        END;
      IF (ch = #27) AND keypressed THEN
        BEGIN
          Read(kbd,ch);
          IF (ch = 'H') THEN
            BEGIN
              l:=l-1;
              cc:=TRUE;
              ii:=ii-5;
            END;
          IF (ch = 'P') THEN
            BEGIN
              l:=l+1;
              cc:=TRUE;
              ii:=ii+5;
            END;
          IF (ch = 'K') THEN
            BEGIN
              k:=k-15;
              cc:=TRUE;
              ii:=ii-1;
            END;
          IF (ch = 'M') THEN
            BEGIN
              k:=k+15;
              cc:=TRUE;
              ii:=ii+1;
            END;
          IF l<1 THEN
            BEGIN
              l:=1;
              largo:=k;
              k:=2;

              ii:=ii+5;
              IF ii>5 THEN
                BEGIN

```

```

ii:=ii-5;
kat:=kat-5;
FOR i:=round((int((ii-1)/5)*5)+1)
  TO kat DO
  BEGIN
    gotoxy(k,1);
    IF ddas[i]<>'no more file' THEN
      BEGIN
        sss:=1;
        ress:=FALSE;
        REPEAT
          IF (ddas[i]=borra[sss]) THEN
            ress:=TRUE;
            sss:=sss+1;
        UNTIL (sss>y1) OR ress;
        textbackground(0);
        textcolor(15);
        IF ress THEN
          BEGIN
            textbackground(1);
            textcolor(15);
          END;
        Write(copy(concat(ddas[i],
                           '),1,14));

      END;
      k:=k+15;
      IF k>62 THEN
        BEGIN
          l:=l+1;
          k:=2;
        END;
      END;
      l:=1;
      lll:=lll+1;
      END;
      k:=largo;
    END;
  IF l>11 THEN
    BEGIN
      l:=11;
      ii:=ii-5;
      IF kat<z THEN
        BEGIN
          kat:=kat+5;
          sss:=1;
          window(2,12,78,22);
          textbackground(0);
          textcolor(15);
          gotoxy(75,11);
          Write(' ');
          window(2,12,78,23);
          REPEAT
            hal:=1;

```

```

ress:=FALSE;
REPEAT
  IF (ddas[kat-5+sss]=borra[hal])
  THEN ress:=TRUE;
  hal:=hal+1
UNTIL (hal>y1) OR ress;
textbackground(0);
textcolor(15);
IF ress THEN
BEGIN
  textbackground(1);
  textcolor(15);
END;
gotoxy(2+((sss-1)*15),11);
Write(ddas[kat-5+sss]);
sss:=sss+1;
UNTIL (sss>5) OR (kat-6+sss=z);
ii:=ii+5;
IF ii>z THEN
BEGIN
  ii:=z;
  k:=2+((sss-2)*15);
END;
111:=111-1;
END;
END;
IF k>62 THEN
BEGIN
  k:=2;
  ii:=ii-5;
END;
IF k<2 THEN
BEGIN
  k:=62;
  ii:=ii+5;
END;
END
ELSE IF (ch = #27) THEN BEGIN
  ok:=TRUE;
  cc:=TRUE;
  ress1:=TRUE;
END;
END;
UNTIL cc;
IF NOT ress1 THEN
BEGIN
  gotoxy(x1,111);
  textbackground(0);
  textcolor(15);
  sss:=1;
  ress:=FALSE;
  REPEAT
    IF (ddas[x2]=borra[sss]) THEN ress:=TRUE;

```

```

        sss:=sss+1
    UNTIL (sss>y1) OR res;
    IF (NOT ok) AND (NOT res) THEN
        WriteLn(copy(concat(ddas[x2], ' '),1,14));
    IF ok THEN
        BEGIN
            sss:=1;
            res:=FALSE;
            REPEAT
                IF (ddas[x2]=borra[sss]) THEN res:=TRUE;
                sss:=sss+1
            UNTIL (sss>y1) OR res;
            sss:=sss-1;
            IF res THEN
                borra[sss]:='.'
            ELSE
                BEGIN
                    textcolor(15);
                    textbackground(1);
                    WriteLn(copy(
                        concat(
                            ddas[x2], ' '),1,14));
                    borra[y1]:=ddas[x2];
                    y1:=y1+1;
                END;
            END;
            lll:=1;
            x1:=k;
        END;
    UNTIL ok;

    IF res1 THEN
        BEGIN
            setdrive(disco);
            IF i=-199 THEN

                FOR i:=1 TO 80 DO
                    IF borra[i]<>'.' THEN
                        ddas[i]:=borraentry(borra[i],$20);

                window(2,12,78,22);
                setdrive(disc2);
                textbackground(0);
                textcolor(15);
                clrscr;
                window(2,12,78,22);
                leer;

            END;
        UNTIL res1;
        sss:=5;
    END;

```

```

{ ***** }
{          RUTINAS  DE DISCOS          }
{ ***** }

(*****)
(*  READSECTOR WRITESECTOR Lee o escribe en uno o varios sectores del  *)
(*      disco, teniendo en cuenta los siguiente parametros:            *)
(*          - nbsec = numero de sectores leidos o escritos (0..8)      *)
(*          - trk = numero de la pista (0..39)                          *)
(*          - sec = numero de sector (1..9)                             *)
(*          - side = cara del disco (0 o 1)                             *)
(*          - dev = numero de la unidad (a=0 b=1)                       *)
(*          - segment = la direccion del segmento del buffer que       *)
(*                   debe recibir los datos o a partir del cual deberan *)
(*                   escribirse.                                         *)
(*          - offset =desplazamiento de este buffer                    *)
(*                                                                      *)
(*                                                                      *)
(*      NOTA :  Prudencia! No es aplicable al disco duro              *)
(*****)

PROCEDURE ReadSector(nbsec,trk,sec,side,dev,segment,offset:INTEGER);

TYPE
    result=RECORD
        AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
    END;

VAR
    parms : result;
    index : INTEGER;

BEGIN
    index:=0;
    REPEAT
        parms.AX:=$0200 OR nbsec;
        parms.CX:=trk SHL 8 OR sec;
        parms.DX:=side SHL 8 OR dev;
        parms.ES:= segment;
        parms.BX:= offset;
        index:=index+1; { cuenta el numero de intentos}
        intr ($13,parms);
        IF NOT(odd(parms.flags))
        THEN
            ReadOk:=0
        ELSE
            ReadOk:=Hi(parms.AX); { Contiene el codigo de error}
        IF NOT(ReadOk=0) THEN
            BEGIN
                parms.AX:=$0D00; {reset del controlador}
                MsDos(parms);
            END;
        END;
    UNTIL ReadOk=0;
END;

```



```
FUNCTION DiskType(d:disquette):BYTE;
VAR
```

```
    drive:BYTE;
    buffer : ARRAY [0..511] OF BYTE;
```

```
BEGIN
```

```
    drive:=ord(d);
    ReadSector(1,0,2,0,drive,Seg(buffer),Ofs(buffer));
    {lee el primer sector de la FAT }
```

```
CASE buffer[0] OF      {tipo disquette en el primer octeto}
    $FF : DiskType:=1;
    $FE : DiskType:=2;
    $FD : DiskType:=3;
    $FC : DiskType:=4;
```

```
END;
```

```
END;
```

```
(*****
(*   DEFDRIVE   Nos da un caracter que representa la unidad por defecto   *)
(*****
```

```
FUNCTION DefDrive:disquette;
```

```
TYPE
```

```
    result = RECORD
                AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
            END;
```

```
VAR
```

```
    parms : result;
```

```
BEGIN
```

```
    parms.AX:=$1900;
```

```
    MsDos(parms);
```

```
    DefDrive:=disquette(Lo(parms.Ax));      {ser A si A1=0}
                                           {utilizacion de rtyping}
```

```
END;
```

```
(*****
(*   SETDRIVE   Selecciona la unidad por defecto segun el valor de d(a,b)   *)
(*****
```

```
PROCEDURE SetDrive(d:disquette);
```

```
TYPE
```

```
    result = RECORD
                AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
            END;
```



```

VAR
    parms : result;

BEGIN
    parms.AX:=$E00; {funcion $E}
    parms.DX:=ord(d); {pasa numero de unidad}
    MsDos(parms);

END;

(*****
(* DISKSPACE Nos da el espacio que queda libre en el disco en Kilobytes *)
(*****)

FUNCTION DiskSpace(d:disquette):REAL;

TYPE
    result = RECORD
        AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
    END;

VAR
    parms:result;
    Space:REAL;
    n:BYTE;

BEGIN
    n:=ord(d)+1;
    parms.AX:=$3600; {convencion distinta para esta funcion}
    parms.DX:=n; {n=0 ser la unidad por defecto}
    MsDos(parms); {en este caso escriba DiskSpace(DefDrive)}
    IF parms.AX=$FFFF
    THEN
        Space:=0 {Unidad no valida}
    ELSE
        BEGIN
            Space:=parms.AX * parms.BX; {calcula en dos veces evita el overflow}
            Space:=Space * parms.CX / 1024;
        END;
        diskSpace:=space;
    END;

END;

(*****
(* DISKOK Nos da el numero de Clusters defectuosos inscritos en la FAT *)
(* .Se trata de hecho de los clusters inutilizables marcados *)
(* con un FF7H *)
(*****)

FUNCTION DiskOk(d:disquette):INTEGER;

```

```

VAR
  drive,base,offset,contador,index,i : INTEGER;
  buffer : ARRAY [0..1023] OF BYTE;

BEGIN
  drive:=ord(d);
  base:=Seg(buffer);
  offset:=Ofs(buffer);
  ReadSector(2,0,2,0,drive,base,offset);
  IF (Mem[base:offset]=$FD) OR (Mem[base:offset]=$FC) THEN
    index:=2
  ELSE
    {FAT de 1 o 2 sectores segun el formato}
    index:=1;

  contador:=0;
  FOR i:=0 TO 512*index-1
    DO BEGIN
      IF (Mem[base:offset+i]=$FF) AND ((i mod 6 = 0) OR (i mod 6=3)) THEN
        IF ((Mem[base:offset+i+1]) AND $F0=$70) THEN
          contador:=Succ(contador);

      {si un octeto es $FF y su numero de orden es un multiplo de 3
       o de 6 , verificar que el digito siguiente es 7 }

      IF (Mem[base:offset+i]=$F7) AND ((i mod 6=2) OR (i mod 6=5)) THEN
        IF ((Mem[base:offset+i-1]) AND $0F=$0F) THEN
          contador:=Succ(contador);

      { si un octeto es igual $F7 y su numero de orden es un multiplo
        de 2 o de 5, verificar que el digito precedente es F}

    END;
    DiskOk:=contador;
  END;

  (*****
  (*      VERIFYSWITCH Activa o desactiva el << verify switch >> del DOS      *)
  (*****

PROCEDURE VerifySwitch(eleccion:switch);

TYPE
  result = RECORD
    AX,BX,CX,DX,BP,SI,DI,DS,ES,flags:INTEGER;
  END;

VAR
  parms:result;

BEGIN
  IF eleccion=on THEN parms.AX:=$2E01
  ELSE
    parms.AX:=$2E00;

```

```
        { bit 0 de AX a 1 o a 0 - llamada funcion $2E}  
    parms.DS:=parms.DS AND $FF00;  
    MsDos(parms);  
END;
```

```
(*****)
(* Este procedimiento dibuja el menu principal de acuerdo a lo especificado *)
(* el diseño de pantalla DDDSCR01. *)
(*****)
```

```
PROCEDURE men;
```

```
  BEGIN
```

```
    hal:=44;
```

```
    window(1,1,80,25);
```

```
    textbackground(3);
```

```
    clrscr;
```

```
    window(69,22,77,24);
```

```
    textcolor(12);
```

```
    textbackground(0);
```

```
    clrscr;
```

```
    window(69,22,77,25);
```

```
    gotoxy(1,1);
```

```
    Write('┌───────────┐');
```

```
    gotoxy(1,2);
```

```
    Write('│    DDA    │');
```

```
    gotoxy(1,3);
```

```
    Write('└───────────┘');
```

```
    textcolor(15);
```

```
    window(10,2,36,4);
```

```
    textbackground(9);
```

```
    clrscr;
```

```
    lin1:='-';
```

```
    lin2:='=';
```

```
    FOR i:=1 TO 80 DO
```

```
      BEGIN
```

```
        lin1:=concat(lin1,'-');
```

```
        lin2:=concat(lin2,'=');
```

```
      END;
```

```
    gotoxy(2,1);
```

```
    Write('┌',copy(lin1,1,23),'┐');
```

```
    gotoxy(2,2);
```

```
    Write('│    MENU PRINCIPAL    │');
```

```
    gotoxy(2,3);
```

```
    Write('└',copy(lin1,1,23),'┘');
```

```
    textcolor(1);
```

```
    window(6,5,40,22);
```

```
    textbackground(15);
```

```
    clrscr;
```

```
    gotoxy(3,2);
```

```
    Write('- ENTRADA DE DATOS POR TECLADO');
```

```
    gotoxy(3,5);
```

```
    Write('- ENTRADA DE DATOS POR ARCHIVO');
```

```
    gotoxy(3,8);
```

```
    Write('- COMUNICACION');
```

```
    gotoxy(3,11);
```

```
    Write('- SALIDA DE DATOS');
```

```
    gotoxy(3,14);
```

```
    Write('- AYUDAS');
```

```

    gotoxy(3,17);
    Write('~- SALIDA HACIA EL SISTEMA OP.~');
END;

```

```

(*****
(* Este procedimiento habilita el servicio de interrupcion de imprimir pan- *)
(* talla, usado en este programa para imprimir las graficas.                *)
(*****

```

```

PROCEDURE pintscreen;
BEGIN
    inline($55/$cd/$05/$5d);
END;

```

```

(*****
(* Este procedimiento dibuja el cuadro de ayuda que se muestra por cada op- *)
(* cion del menu principal (DDDSCR01) al mover el cursor.                  *)
(*****

```

```

PROCEDURE ventana(b:INTEGER);
BEGIN
    window(hal,b-round(largo/2),hal+33,b+round(largo/2)-1);
    textbackground(color);
    textcolor(15);
    clrscr;
    window(hal,b-round(largo/2),hal+34,b+round(largo/2));
    gotoxy(1,1);
    Write('┌',copy(lin2,1,32),'┐');
    FOR i:=2 TO largo-1 DO
    BEGIN
        gotoxy(1,i);
        Write('│');
        gotoxy(34,i);
        Write('│');
    END;
    gotoxy(1,largo);
    Write('└',copy(lin2,1,32),'┘');
END;

```

```

(*****
(* Este procedimiento implementa el movimiento del cursor de las opciones *)
(* del menu principal, retorna en "a" el numero de opcion seleccionado      *)
(* por el usuario al presionar ENTER.                                         *)
(*****

```

```

OVERLAY PROCEDURE curso(VAR a:INTEGER);
VAR
    ch:CHAR;
    c,o:BOOLEAN;
BEGIN
    o:=FALSE;
    l:=a;
    REPEAT

```

```

window(6,5,40,22);
gotoxy(2,1);
j:=1;
IF l=2 THEN
BEGIN
    textbackground(0);
    textcolor(15);
    Write(' - ENTRADA DE DATOS POR TECLADO ');
    ventana(l+2);
    window(6,5,40,22);
END;
IF l=5 THEN
BEGIN
    textbackground(0);
    textcolor(15);
    Write(' - ENTRADA DE DATOS POR ARCHIVO ');
    ventana(l+2);
    window(6,5,40,22);
END;
IF l=8 THEN
BEGIN
    textbackground(0);
    textcolor(15);
    Write(' - COMUNICACION ');
    ventana(l+2);
    window(6,5,40,22);
END;
IF l=11 THEN
BEGIN
    textbackground(0);
    textcolor(15);
    Write(' - SALIDA DE DATOS ');
    ventana(l+2);
    window(6,5,40,22);
END;
IF l=14 THEN
BEGIN
    textbackground(0);
    textcolor(15);
    Write(' - AYUDAS ');
    ventana(l+2);
    window(6,5,40,22);
END;
IF l=17 THEN
BEGIN
    textbackground(0);
    textcolor(15);
    Write(' - SALIDA HACIA EL SISTEMA OP. ');
    ventana(l+2);
    window(6,5,40,22);
END;
gotoxy(3,1);
c:=FALSE;

```

```

REPEAT
  IF keypressed THEN
    BEGIN
      Read(kbd,ch);
      IF (ch = #13) THEN
        BEGIN
          o:=TRUE;
          c:=TRUE;
        END;
      IF (ch = #27) AND keypressed THEN
        BEGIN
          Read(kbd,ch);
          IF (ch = 'H') THEN
            BEGIN
              l:=l-3;
              c:=TRUE;
            END;
          IF (ch = 'P') THEN
            BEGIN;
              l:=l+3;
              c:=TRUE;
            END;
          IF l<2 THEN l:=17;
          IF l>17 THEN l:=2;
        END;
      END;
    UNTIL c;
    textbackground(3);
    window(44,j-1,77,j+4);
    clrscr;
    window(6,5,40,22);
    gotoxy(2,j);
    IF (j=2) AND (NOT o) THEN
      BEGIN
        textbackground(15);
        textcolor(1);
        Write(' - ENTRADA DE DATOS POR TECLADO ');
      END;
    IF (j=5) AND (NOT o) THEN
      BEGIN
        textbackground(15);
        textcolor(1);
        Write(' - ENTRADA DE DATOS POR ARCHIVO ');
      END;
    IF (j=8) AND (NOT o) THEN
      BEGIN
        textbackground(15);
        textcolor(1);
        Write(' - COMUNICACION ');
      END;
    IF (j=11) AND (NOT o) THEN
      BEGIN
        textbackground(15);

```



```

        textcolor(1);
        Write(' - SALIDA DE DATOS                ');
    END;
    IF (j=14) AND (NOT o) THEN
    BEGIN
        textbackground(15);
        textcolor(1);
        Write(' - AYUDAS                ');
    END;
    IF (j=17) AND (NOT o) THEN
    BEGIN
        textbackground(15);
        textcolor(1);
        Write(' - SALIDA HACIA EL SISTEMA OP.  ');
    END;
    UNTIL o;
    a:=1;
    END;

(*****
(* Este modulo asigna el disco seleccionado por el usuario al nombre de un *)
(* archivo para determinar si este existe o no. En caso de que no exista *)
(* este procedimiento mostrara el mensaje de error correspondiente.      *)
(*****)

PROCEDURE dassign(infi :workstring;i,j:INTEGER);
    BEGIN
        IF disco=A THEN infi:='a:'+infi;
        IF disco=B THEN infi:='b:'+infi;
        IF disco=C THEN infi:='c:'+infi;
        assign(infile,infi);
        {$i-} reset (infile){$i+};
        ok:=(ioresult = 0);
        IF (NOT ok) THEN
            BEGIN
                gotoxy(i,j);
                Write('Archivo no existente');
            END;
        END;
    END;

(*****
(* Este modulo permite grabar los datos de entrada leidos por teclado o de *)
(* un archivo al disco y directorio actual. Utiliza la estructura de datos *)
(* "archi" para facilitar su obtencion. Ademas crea un archivo simultanea- *)
(* mente con los mismo datos pero en formato de secuencia de caracteres    *)
(* para facilidad de la comunicacion.                                       *)
(*****)

PROCEDURE graba(infilename:workstring);
    VAR
        textfile:TEXT;
    BEGIN
        setdrive(disco);

```

```

assign(infile,infilename);
rewrite(infile);
archi.segundo[1]:=nint;           { Numero de Integradores }
archi.segundo[2]:=nite;          { Numero de Iteraciones }
archi.tercero[1]:=xst;           { Valor Inicial }
archi.tercero[2]:=xfn;           { Valor Final }
archi.segundo[3]:=prnt;          { Paso de impresion }
Write(infile,archi);
FOR i:=1 TO nint DO
BEGIN
    archi.segundo[1]:=dy1[i];     { Entradas dy }
    archi.segundo[2]:=dy2[i];
    archi.segundo[3]:=dy3[i];
    archi.segundo[4]:=dy4[i];
    archi.segundo[5]:=dy5[i];
    archi.segundo[6]:=dx[i];      { Entrada dx }
    archi.primerio:=nom[i];       { Nombre del Integrador }
    Write(infile,archi);
END;
FOR i:=1 TO nint DO
BEGIN
    archi.primerio[1]:=sal[i];     { salida de datos }
    archi.segundo[1]:=cx[i];       { Factor de Escala n }
    archi.segundo[2]:=cy[i];       { P_bit de precision }
    archi.tercero[1]:=cinic[i];    { Condicion Inicial }
    Write(infile,archi);
END;
close(infile);
lar:=8;
IF length(infilename)<8 THEN lar:=length(infilename);
sss:=0;
FOR i:=1 TO lar DO
    IF infilename[i]='.' THEN sss:=i;
numero:=concat(copy(infilename,1,sss),'dde');
assign(texfile,numero);
rewrite(texfile);
str(nint,sttr);                   { Numero de integradores }
Write(texfile,sttr);
WriteLn(texfile);
i:=1;
FOR j:=1 TO nite DO
i:=2*i;
numr:=int(i*(xfn-xst)+prnt);
str(numr:0:0,sttr);               { Fin de iteraciones }
Write(texfile,sttr);
WriteLn(texfile);
str(prnt,sttr);                   { Paso de Impresion }
Write(texfile,sttr);
WriteLn(texfile);
FOR i:=1 TO nint DO               { Entradas dy }
BEGIN
    str(dy1[i],s1);
    str(dy2[i],s2);

```

```

    str(dy3[i],s3);
    str(dy4[i],s4);
    str(dy5[i],s5);
    sttr:=concat(s1,' ',s2,' ',s3,' ',s4,' ',s5);
    Write(texfile,sttr);
    WriteLn(texfile);
END;
str(dx[1],s1);           { Entrada dx           }
s5:=sal[1];              { salida de datos      }
str(cx[1],s2);           { Factor de escala n    }
str(cy[1],s3);           { P_bit de precision   }

k:=1;
FOR i:=1 TO cy[1] DO
k:=k*2;
numr:=int(cinic[1]*k);   { Yi=cond.inic. * conc.dy}
str(numr:0:0,s4);
FOR i:=2 TO nint DO
BEGIN
    str(dx[i],sttr);
    s1:=concat(s1,' ',sttr);
    str(cx[i],sttr);
    s2:=concat(s2,' ',sttr);
    str(cy[i],sttr);
    s3:=concat(s3,' ',sttr);
    k:=1;
    FOR j:=1 TO cy[i] DO
    k:=k*2;
    numr:=int(cinic[i]*k);
    str(numr:0:0,sttr);
    s4:=concat(s4,' ',sttr);
    s5:=concat(s5,' ',sal[i]);
END;
Write(texfile,s1);
WriteLn(texfile);
Write(texfile,s5);
WriteLn(texfile);
Write(texfile,s2);
WriteLn(texfile);
Write(texfile,s3);
WriteLn(texfile);
Write(texfile,s4);
WriteLn(texfile);
Write(texfile,'F');
WriteLn(texfile);
close(texfile);
setdrive(disc2);
END;
```

```

(* **** *)
(* Este modulo imprime un reporte con formato especificado en DDALST01 en *)
(* la pantalla. Los datos que se imprimen son los de entrada. *)
(* **** *)
```

```

PROCEDURE datosent(filename:workstring);
VAR
  c1,c2,c3,c4:workstring;
  r,s,t,u,v:INTEGER;

BEGIN
  textbackground(9);
  textcolor(0);
  window(1,1,80,25);
  clrscr;
  c1:='';
  FOR i:=1 TO 80 DO
    c1:=concat(c1,'D');
  gotoxy(25,1);
  WriteLn(copy(c1,1,30));
  textcolor(15);
  c2:=concat('Archivo : ',filename);
  r:=round(length(c2)/2);
  gotoxy(40-r,2);
  WriteLn(c2);
  c2:='DATOS DE ENTRADA';
  gotoxy(32,3);
  WriteLn(c2);
  textcolor(0);
  gotoxy(25,4);
  WriteLn(copy(c1,1,30));
  textcolor(15);
  WriteLn;
  WriteLn('          Numero de integradores : ',nint);
  WriteLn('          S_bit de precision      : ',nite);
  WriteLn('          Paso de impresion        : ',prnt);
  WriteLn('          Valor Inicial             : ',xst:0:4);
  WriteLn('          Valor Final               : ',xfn:0:4);
  WriteLn;
  Write(' Nombre del Integrador  dy1 dy2 dy3 dy4 dy5  dx  n  ');
  WriteLn('p      C.Inic      SALE');
  textcolor(0);
  WriteLn(c1);
  textcolor(15);
  l:=15;
  FOR i:=1 TO nint DO
    BEGIN
      gotoxy(4,1);
      Write(nom[i]);
      gotoxy(26,1);
      Write(dy1[i]);
      gotoxy(30,1);
      Write(dy2[i]);
      gotoxy(34,1);
      Write(dy3[i]);
      gotoxy(38,1);
      Write(dy4[i]);
    
```

```

        gotoxy(42,1);
        Write(dy5[i]);
        gotoxy(46,1);
        Write(dx[i]);
        gotoxy(50,1);
        Write(cx[i]);
        gotoxy(56,1);
        Write(cy[i]);
        gotoxy(62,1);
        Write(cinic[i]:0:3);
        gotoxy(77,1);
        Write(sal[i]);
        l:=l+1;
        IF l=24 THEN
        BEGIN
            l:=15;
            Read(sttr);
            window(1,1,80,25);
            clrscr;
            window(1,1,80,25);
        END;
    END;
    REPEAT
    UNTIL keypressed;
END;

```

```

(*****
(* Este modulo imprime un reporte con formato especificado en DDALST02 en *)
(* la pantalla. Los datos que se imprimen son los de salida. *)
(*****

```

```

PROCEDURE listar(resultado:workstring);
VAR
    AbortChar : CHAR;
    sr,dr:workstring;
    texto:TEXT[$1000];
    k,a1,a8,a2,a10,a9,a3,a4,a5,a6,a7,numr:INTEGER;
    r1,r2,r3,r4,nite2:REAL;
BEGIN
    IF disco=A THEN resultado:='a:'+resultado;
    IF disco=B THEN resultado:='b:'+resultado;
    IF disco=C THEN resultado:='c:'+resultado;

    assign(texto,resultado);
    {$i-} reset(texto) {$i+};
    ok:=(ioresult=0);
    IF (ok) THEN
    BEGIN
        close (texto);
        i:=1;
        FOR j:=1 TO nite DO
            i:=2*i;

```

```

nite2:=i;
numr:=round(i*(xfn-xst)+prnt);
r:=chr(196);
HFOR i:=1 TO 80 DO
dr:=concat(dr,chr(196));
window(1,1,80,25);
textbackground(1);
clrscr;
a5:=1;
l1l:=1;
a2:=0;
j:=0;
a6:=0;
a10:=0;
k:=0;
FOR a1:=1 TO nint DO BEGIN
  IF (sal[a1]='S') THEN BEGIN
    j:=j+1;
    IF (pan[j]=TRUE) THEN a2:=a2+1;
  END;
END;
a9:=j;
REPEAT
textbackground(14);
textcolor(0);
window(2,2,79,4);
clrscr;
window(2,2,80,7);
gotoxy(10,1);
WriteLn('                                ',resultado);
WriteLn(' Paso ');
WriteLn;
Write('_____');
WriteLn('_____');
  assign(texto,resultado);
  {$i-} reset(texto) {$i+};
IF nint<a5+6 THEN
  a4:=nint-a5+2
ELSE
  a4:=6;
a10:=a6+a10;
k:=a10;
a8:=k;
a6:=0;
gotoxy(10,2);
Write(base);
gotoxy(18,2);
FOR i:=a5 TO nint+1 DO
  IF (sal[i]='S') AND (a6<a4) THEN
  BEGIN
    k:=k+1;
    IF pan[k]=TRUE THEN
    BEGIN

```

```

        IF i<10 THEN Write(' ',i,' ');
        IF i>=10 THEN Write(' ',i,' ');
        a6:=a6+1;
        a3:=i;
    END;
END;
WriteLn;
k:=a8;
a6:=0;
gotoxy(18,3);
textcolor(15);
FOR i:=a5 TO nint+1 DO
    IF (sal[i]='S') AND (a6<a4) THEN
    BEGIN
        k:=k+1;
        IF pan[k]=TRUE THEN
        BEGIN
            sttr:=copy(concat(nom[i],' '),1,8);
            Write(' ',sttr);
            a6:=a6+1;
        END;
    END;
END;
WriteLn;
textbackground(7);
textcolor(1);
window(2,6,79,24);
clrscr;
k:=a6;
j:=6;
i := 1;
AbortChar := 0;
WHILE (AbortChar <> #27) AND (i <= (numr div (prnt+1))) DO
BEGIN
    ReadLn(texto,sttr);
    j:=j+1;
    r1:=(((i-1)*(prnt+1))/nite2)+xst;
    Write((i-1)*(prnt+1):6,' ',r1:10:3);
    k:=1;
    l:=1;
    s:=a5;
    REPEAT
        IF sal[s]='S' THEN
        BEGIN
            IF (pan[k+a5-1]) AND (l<=a6) THEN
            BEGIN
                sr:=copy(sttr,((k+a5-1)*5)-4,5);
                val(sr,r1,result);
                IF r1>=32768.0 THEN
                    r1:=r1-65536.0;
                r2:=1;
                IF cy[s]>0 THEN FOR a7:=1 TO cy[s] DO
                    r2:=2*r2;
                    r1:=r1/r2;

```



```

        Write(r1:10:3);
        l:=l+1;
    END;
    k:=k+1;
    END;
    s:=s+1;
    UNTIL (l=a4+1) OR (s>nint);
    WriteLn;
    IF (j = 24) THEN
    BEGIN
        AbortChar := 0;
        REPEAT
            IF Keypressed THEN
                read(kbd,AbortChar);
            UNTIL AbortChar <> #0;
            j:=6;
            clrscr;
        END;
        i := i + 1;
    END;
    a5:=a3+1;
    FOR i:=1 TO a1 DO
    Write(' ');
    close(texto);
    lll:=lll+1;
    IF AbortChar <> #27 THEN
    BEGIN
        AbortChar := #0;
        REPEAT
            IF Keypressed THEN
                read(kbd,AbortChar);
            UNTIL AbortChar <> #0;
        END;
    UNTIL (AbortChar = #27) OR ( a5>nint) OR
        (a10+a6=a2) OR ((a5=nint) AND (pan[a9]=FALSE));
    END;
END;

```



BIBLIOTECA

```

(*****
(* Este modulo imprime un reporte con formato especificado en DDALST02 en *)
(* la impresora. Los datos que se imprimen son los de salida. *)
(*****

```

```

PROCEDURE listarimp(resultado:workstring);
VAR
    sr,dr:workstring;
    texto:TEXT[$1000];
    k,a1,a8,a2,a10,a9,a3,a4,a5,a6,a7,numr:INTEGER;
    r1,r2,r3,r4,nite2:REAL;
BEGIN
    i:=1;
    setdrive(disco);
    FOR j:=1 TO nite DO

```

```

i:=2*i;
nite2:=i;
numr:=round(i*(xfn-xst)+prnt);
dr:=chr(196);
FOR i:=1 TO 80 DO
dr:=concat(dr,chr(196));
a5:=1;
l1l:=1;
a2:=0;
j:=0;
a6:=0;
a10:=0;
k:=0;
FOR a1:=1 TO nint DO BEGIN
  IF (sal[a1]='S') THEN BEGIN
    j:=j+1;
    IF (imp[j]=TRUE) THEN a2:=a2+1;
  END;
END;
a9:=j;
REPEAT
{$i-} WriteLn(lst) {$i+};
ok:=(ioresult=0);
IF NOT ok THEN BEGIN
  gotoxy(3,5);
  Write('ERROR DE IMPRESORA');
  REPEAT UNTIL keypressed;
  gotoxy(3,5);
  Write(' ');
END;
UNTIL (ok);

REPEAT
  gotoxy(3,5);
  Write('Impresora ...');
  REPEAT UNTIL keypressed;
  gotoxy(3,5);
  Write(' ');
  WriteLn(lst);
  WriteLn(lst);
  WriteLn(lst);
  WriteLn(lst,' ',resultado);
  WriteLn(lst);
  Write(lst,' Paso ');
  assign(texto,resultado);
  {$i-} reset(texto) {$i+};
  IF nint<a5+6 THEN
    a4:=nint-a5+2
  ELSE
    a4:=6;
  a10:=a6+a10;
  k:=a10;
  a8:=k;

```



```

        val(sr,r1,result);
        IF r1>=32768.0 THEN
            r1:=r1-65536.0;
            r2:=1;
            IF cy[s]>0 THEN      FOR a7:=1 TO cy[s] DO
                r2:=2*r2;
                r1:=r1/r2;
            Write(lst,r1:10:3);
            l:=l+1;
        END;
        k:=k+1;
    END;
    s:=s+1;
UNTIL (l=a4+1) OR (s>nint);
WriteLn(lst);
IF (j = 60) THEN
BEGIN
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    WriteLn(lst);
    j:=6;
END;
END;
a5:=a3+1;
FOR i:=1 TO a1 DO
    Write(lst,' ');
close(texto);
lll:=lll+1;
UNTIL ( a5>nint) OR (a10+a6=a2) OR ((a5=nint) AND (imp[a9]=FALSE));
setdrive(disc2);
END;

(*****
(* Este procedimiento convierte un numero representado en cadena al dato      *)
(* numerico decimal correspondiente.                                          *)
(*****

PROCEDURE leenum(sr:workstring ; VAR r1:REAL;VAR nn:INTEGER);
BEGIN
    sttr:=copy(sr,(nn*5)-4,5);
        val(sttr,r1,result);
        IF r1>=32768.0 THEN
            r1:=r1-65536.0;
        END;

```

```
(*****
(* Este procedimiento dibuja las graficas a partir de los datos de salida *)
(* en la pantalla de alta resolucio. Permite graficar maximo de graficas *)
(* en el mismo plano, coloca marcas de distincion en para diferenciar una *)
(* grafica de otra. *)
(*****)
```

```
PROCEDURE grafica(resultado:workstring;ni:INTEGER;xt,xn:REAL;xc,yc:INTEGER;
          nt:INTEGER);
```

```
{i graph.p }
```

```
VAR
```

```
dsttr:STRING[80];
```

```
texto:TEXT;
```

```
aa3,nite2,borx1,borx2,bory1,bory2,
```

```
com,ecu,yo,aa1,aa2,b1,a3,a4,b2,l:INTEGER;
```

```
r1,r2,s,a1,a2,r3,r4,be,bf,mayor,menor:REAL;
```

```
x1,y1,x2,y2:REAL;
```

```
BEGIN
```

```
setdrive(disco);
```

```
assign(texto,resultado);
```

```
{ $i- } reset(texto) { $i+ };
```

```
ok:=(ioresult=0);
```

```
IF (NOT ok) THEN
```

```
BEGIN
```

```
gotoxy(10,23);
```

```
ventana(16);
```

```
gotoxy(7,5);
```

```
Write('Archivo no existente');
```

```
delay(2000);
```

```
END;
```

```
IF ok THEN
```

```
BEGIN
```

```
textbackground(0);
```

```
textcolor(14);
```

```
window(1,1,80,25);
```

```
hires;
```

```
hirescolor(15);
```

```
graphbackground(14);
```

```
borx1:=115;
```

```
borx2:=615;
```

```
bory1:=18;
```

```
IF graf>=2 THEN
```

```
borx2:=555;
```

```
IF graf=3 THEN
```

```
borx1:=155;
```

```
draw(borx1,18,borx2,18,1);
```

```
draw(borx1,18,borx1,179,1);
```

```
draw(borx1,179,borx2,179,1);
```

```
draw(borx2,18,borx2,179,1);
```

```
gotoxy(63,25);
```

```
Write(base);
```

```
com:=0;
```

```

ecu:=1;
yo:=1;
k:=0;
REPEAT
  a3:=0;
  com:=com+1;
  a4:=1;
  FOR i:=com TO nint+1 DO
    IF (sal[i]='S') AND (a4=1) THEN
      BEGIN
        k:=k+1;
        IF (((gpan[k]) AND (sat=3)) OR
            ((gimp[k]) AND (sat=5))) THEN
          BEGIN
            a4:=a4+1;
            a3:=i;
            ecu:=i;
          END;
        END;
      numero:=nom[a3];
      IF graf>=0 THEN
        BEGIN
          i:=length(numero);

          IF graf=1 THEN
            FOR j:=1 TO i DO
              BEGIN
                gotoxy(2,2+(j-1));
                Write(numero[j]);
              END;
            numero:=concat(numero,' ','Vs. ',base);
            i:=length(numero);
            IF yo=1 THEN gotoxy(2,1);
            IF yo=3 THEN gotoxy(10,2);
            IF yo=2 THEN gotoxy(74-i,1);
            IF graf=1 THEN gotoxy(40-round(i/2),1);
            Write(numero);
            IF graf>1 THEN
              BEGIN
                IF yo=1 THEN Write(' ( x )');
                IF yo=3 THEN
                  BEGIN
                    Write(' (   )');
                    i:=i*8+99;
                    draw(i,9,i+4,15,1);
                    draw(i,9,i-4,15,1);
                    draw(i-4,15,i+4,15,1);
                  END;
                IF yo=2 THEN
                  BEGIN
                    Write(' (   )');
                    i:=610;
                    draw(i,1,i-4,4,1);

```

```

        draw(i,1,i+4,4,1);
        draw(i+4,4,i,7,1);
        draw(i-4,4,i,7,1);
    END;
END;

```

```

END;
mayor:=1;
com:=ecu;
FOR i:=1 TO cx[a3] DO
    mayor:=mayor*2;
menor:=-mayor;
ReadLn(texto,sttr);
leenum(sttr,s,k);

```

{**PARA SACAR EL MAYOR Y MENOR NUMERO A GRAFICAR**}

```

a1:=s;
a2:=s;
nite2:=1;
FOR i:=1 TO nite DO
    nite2 :=nite2*2;
nite2:=round((xfn-xst)*nite2);
FOR i:=1 TO nite2-1 DO
BEGIN
    ReadLn(texto,sttr);
    leenum(sttr,s,k);
    IF s>a2 THEN a2:=s;
    IF s<a1 THEN a1:=s;
END;
be:=1;
FOR i := 1 TO cy[a3] DO
be:=2*be;
aa3:=a3;
IF (a1<>a2) THEN BEGIN
    mayor:=a2/be;
    menor:=a1/be;
END;
reset(texto);

```

{**PARA GRAFICAR VALORES EN LOS DOS EJES, DIVIDIDO EN SEIS PARTES**}

```

r1:=(xfn-xst)/5;
r2:=(mayor-menor)/5;
r3:=xst;
r4:=menor;
gotoxy(6,24);
borx1:=6;
borx2:=13;
bory2:=12;
IF ((graf>=2) AND (yo=1)) THEN
BEGIN

```



```

    borx1:=1;
    bory2:=11;
    IF graf>2 THEN bory2:=10;
END;
IF graf=3 THEN borx2:=17;
IF ((graf>=2) AND (yo=2)) THEN
BEGIN
    borx1:=71;
    bory2:=11;
    IF graf>2 THEN bory2:=10;
END;
IF ((graf>2) AND (yo=3)) THEN
BEGIN
    borx1:=10;
    bory2:=10;
END;
FOR i:=1 TO 6 DO
BEGIN
    str(r3:0:3, sttr);
    str(r4:8:3, dsttr);
    r3:=r3+r1;
    r4:=r4+r2;
    gotoxy(borx2+(i-1)*bory2, 24);
    Write(sttr);
    gotoxy(borx1, 23-(i-1)*4);
    Write(dsttr);
END;
borx1:=115;
borx2:=615;
bory1:=50;
IF graf>=2 THEN
BEGIN
    bory1:=44;
    borx2:=555;
END;
IF graf=3 THEN
BEGIN
    bory1:=40;
    borx1:=155;
END;
FOR i:=1 TO 9 DO
BEGIN
    draw(borx1+bory1+(i-1)*bory1, 181,
        borx1+bory1+(i-1)*bory1, 177, 1);
    draw(borx1-2, 35+(i-1)*16, borx1+2, 35+(i-1)*16, 1);
END;
borx1:=501;
IF graf=2 THEN borx1:=441;
IF graf=3 THEN borx1:=401;
r1:=(xfn-xst)/nite2;
r3:=(xfn-xst)/borx1;
r4:=(mayor-menor)/161;
x1:=xst;

```

```

aa1:=115;
borx1:=115;
IF graf>2 THEN
BEGIN
    aa1:=155;
    borx1:=155;
END;

```

{**GRAFICA LAS LINEAS DE CADA GRAFICO**}

```

ReadLn(texto,sttr);
leenum(sttr,s,k);
b1:=179-round(((s/be)-menor)/r4);
FOR i:=1 TO (nite2 div (prnt+1)) DO
BEGIN
    gotoxy(20,24);
    x1:=x1+r1;
    aa2:=round(((x1-xst)*(prnt+1))/r3)+borx1;
    ReadLn(texto,sttr);
    leenum(sttr,s,k);
    b2:=179-round(((s/be)-menor)/r4);
    draw(aa1,b1,aa2,b2,1);
    IF ((graf>1) AND (i=1)) THEN
    BEGIN
        IF yo=1 THEN
        BEGIN
            draw(aa1-4,b1-2,aa1+4,b1+2,1);
            draw(aa1-4,b1+2,aa1+4,b1-2,1);
        END;
        IF yo=3 THEN
        BEGIN
            draw(aa1,b1-3,aa1+4,b1+3,1);
            draw(aa1,b1-3,aa1-4,b1+3,1);
            draw(aa1-4,b1+3,aa1+4,b1+3,1);
        END;
        IF yo=2 THEN
        BEGIN
            draw(aa1,b1-3,aa1-4,b1,1);
            draw(aa1,b1-3,aa1+4,b1,1);
            draw(aa1+4,b1,aa1,b1+3,1);
            draw(aa1-4,b1,aa1,b1+3,1);
        END;
    END;
    aa1:=aa2;
    b1:=b2;
END;

IF graf>1 THEN
BEGIN
    IF yo=1 THEN
    BEGIN
        draw(aa1-4,b1-2,aa1+4,b1+2,1);
    END;

```

```

        draw(aa1-4,b1+2,aa1+4,b1-2,1);
    END;
    IF yo=3 THEN
    BEGIN
        draw(aa1,b1-3,aa1+4,b1+3,1);
        draw(aa1,b1-3,aa1-4,b1+3,1);
        draw(aa1-4,b1+3,aa1+4,b1+3,1);
    END;
    IF yo=2 THEN
    BEGIN
        draw(aa1,b1-3,aa1-4,b1,1);
        draw(aa1,b1-3,aa1+4,b1,1);
        draw(aa1+4,b1,aa1,b1+3,1);
        draw(aa1-4,b1,aa1,b1+3,1);
    END;
    END;
    yo:=yo+1;
    reset(texto);
    UNTIL yo>graf;
    close(texto);
    setdrive(disc2);
    IF sat=5 THEN
    BEGIN
        pintscreen;
    END;
    REPEAT UNTIL keypressed;
    imprime:=FALSE;
    textmode;
    END ;
END;

```

```
(*****
(*)          INICIALIZACION DEL 8250 UART          *)
(*****)
```

```
PROCEDURE init2;
```

```
BEGIN
```

```
inline($55/      {    PUSH    BP      }
      $BA/$FB/$03/ {    MOV     DX,3FBH }
      $B0/$80/     {    MOV     AL,80H  }
      $EE/         {    OUT      DX,AL   }
      $BA/$F8/$03/ {    MOV     DX,3F8H  }
      $B0/$60/     {    MOV     AL,60H  }
      $EE/         {    OUT      DX,AL   }
      $BA/$F9/$03/ {    MOV     DX,3F9H  }
      $B0/$00/     {    MOV     AL,0    }
      $EE/         {    OUT      DX,AL   }
      $BA/$FB/$03/ {    MOV     DX,3FBH  }
      $B0/$0B/     {    MOV     AL,0BH  }
      $EE/         {    OUT      DX,AL   }
      $BA/$FC/$03/ {    MOV     DX,3FCH  }
      $B0/$03/     {    MOV     AL,03H  }
      $EE/         {    OUT      DX,AL   }
      $BA/$F9/$03/ {    MOV     DX,3F9H  }
      $B0/$00/     {    MOV     AL,0    }
      $EE/         {    OUT      DX,AL   }
      $5D);        {    POP      BP      }
```

```
END;
```

```
(*****
(*)          FUNCION PARA RECIBIR UN CARACTER          *)
(*****)
```

```
FUNCTION reci2:INTEGER;
```

```
BEGIN
```

```
inline($55/      {    PUSH    BP      }
      $8B/$EC/     {    MOV     BP,SP   }
      $83/$C5/$08/ {    ADD     BP,8    }
      $1E/         {    PUSH    DS    }
      $BA/$FD/$03/ {    MOV     DX,3FDH  }
      $B9/$FF/$FF/ {    MOV     CX,4000H  }
      $49/         { LAZ:  DEC     CX    }
      $74/$10/     {    JZ      NCA    }
      $EC/         {    IN       AL,DX   }
      $A8/$1E/     {    TEST    AL,1EH  }
      $75/$10/     {    JNZ     ERR    }
      $A8/$01/     {    TEST    AL,01H  }
      $74/$F4/     {    JZ      LAZ    }
      $BA/$F8/$03/ {    MOV     DX,3F8H  }
      $EC/         {    IN       AL,DX   }
      $EB/$08/$90/ {    JMP     MUE    }
      $B0/$FF/     { NCA:  MOV     AL,0FFH  }
      $EB/$03/$90/ {    JMP     MUE    }
```



```

$B0/$00/      { ERR: MOV    AL,0      }
$B4/$00/      { MUE: MOV    AH,0      }
$89/$46/$00/  {          MOV    BP,AX   }
$1F/          {          POP    DS      }
$5D);         {          POP    BP      }

END;

```

```

(*****
(*)          PROCEDIMIENTO PARA TRANSMITIR UN CARACTER          (*)
(*****

```

```

PROCEDURE transmit(mx:INTEGER);
BEGIN

```

```

inline ($55/      {          PUSH    BP      }
      $8B/$EC/      {          MOV    BP,SP   }
      $83/$C5/$08/  {          ADD    BP,8     }
      $1E/          {          PUSH    DS      }
      $BA/$FD/$03/  { LAZ: MOV    DX,3FDH }
      $EC/          {          IN      AL,DX   }
      $A8/$20/      {          TEST   AL,20H   }
      $74/$F8/      {          JZ     LAZ      }
      $BA/$F8/$03/  {          MOV    DX,3F8H }
      $8A/$46/$00/  {          MOV    AL,BP   }
      $EE/          {          OUT    DX,AL   }
      $1F/          {          POP    DS      }
      $5D);         {          POP    BP      }

```

```

END;

```

```

(*****
(*) PROCEDIMIENTO PARA COMUNICACION DE LA COMPUTADORA IBM-PC CON LA TRS-80 *)
(*) TRANSMITE EL ARCHIVO CON EXTENSION ".DDE" QUE CONTIENE LOS DATOS DE      *)
(*) ENTRADA Y RECIBE EL ARCHIVO CON EXTENSION ".DDS" QUE SON LOS RESULTADOS*)
(*) DE LAS VARIABLES DE LOS INTEGRADORES DIGITALES                          *)
(*****

```

```

PROCEDURE comunica(txrigo:workstring);

```

```

VAR

```

```

sttr,sttr2:STRING[80];
textfile:TEXT[$5000]; {Buffer de memoria}
x:REAL;
ppar,i,j,k,l,r,num,cont,band:INTEGER;
ch:CHAR;
arreglo:ARRAY[1..255] of CHAR;

```

```

BEGIN

```

```

    init2;

```

```

    i:=0;

```

```

    REPEAT

```

```

        i:=i+1;

```

```

    UNTIL txrigo[i]='.';

```

```

sstr2:=concat(copy(txrigo,1,i),'dds');

{ Transmision del archivo con extension ".DDE"}

band:=0;
setdrive(disco);
assign(textfile,txrigo);
reset(textfile);
cont := 0;
ppar:=0;
num:= ord('T');
transmit(num);
REPEAT {handshake}
  r:=reci2;
UNTIL (chr(r) = 'R') OR (r = $0FF);
IF r <> $0FF THEN
  BEGIN

    REPEAT
      ReadLn(textfile,sstr);
      i:=length(sstr);
      FOR j:=1 TO i DO
        BEGIN
          num := ord(sstr[j]);
          ppar:=ppar XOR num;
          delay(25);
          transmit(num); {transmision de caracteres del archivo}
          cont := cont + 1;
        END;
      num := ord('L');
      ppar:=ppar XOR num; {palabra de paridad}
      delay(25);
      transmit(num);

      UNTIL sstr[1] = 'F'; {caracter de fin de archivo}
      delay(25);
      transmit(ppar);
      num:=ord('Y'); {caracter de fin de transmision}
      delay(25);
      transmit(num);

    END;

gotoxy(5,5);
WriteLn('caracteres transmitidos ',cont);

close(textfile);
{ Recepcion del archivo con extension ".DDS" }

IF r = $0FF THEN band:= 1;
IF r <> $0FF THEN
  BEGIN
    assign(textfile,sstr2);

```



```

rewrite(textfile);
x:=0;
ppar:=0;
REPEAT
  k:=0;
  REPEAT {handshake}
    k:=k+1;
    r:=reci2;
    IF r = 89 THEN
      BEGIN
        flush(textfile); {graba a disco}
        k:=2;
      END;

  UNTIL (chr(r) = 'T') OR (k = 2);
  IF k = 2 THEN band:=1;
  IF k <> 2 THEN {si existe cable}
    BEGIN
      num:=ord('R');
      transmit(num);
      j :=reci2;
      IF (j <> 26) AND (j > 9) THEN {si es caracter ASCII}
        BEGIN
          Write(textfile,chr(j));
          ppar:=ppar XOR j; {palabra de paridad}
          x:=x+1
        END;
    END;

  REPEAT
    num:=reci2; {recepcion de caracteres del archivo}

    IF (num > 9) AND (num <> $0FF) THEN {si es caracter ASCII}

      BEGIN

        x:=x+1;
        ch :=chr(num);
        ppar:=ppar XOR num;
        Write(textfile,ch);
        IF (num=10) OR (num=26) THEN
          BEGIN
            ppar:=ppar + 1;
            REPEAT
              j:=reci2;
            UNTIL (j <> 0);
            ppar:=ppar XOR j;
            IF (ppar<>0) THEN band:=2;{control de la palabra
              de paridad}
          END;
        END;

      END;

    UNTIL (num = 10) OR (num = 26) OR (num = $0FF);

```



```

        END;      {fin de transmision por fin de linea o fin de archivo}

UNTIL (ch=chr(26)) OR (k=2) ; {fin de transmision por fin de archivo
                             o ausencia de transmision}
gotoxy(5,6);

IF k <> 2 THEN
    BEGIN
        WriteLn('caracteres recibidos ',x:4:0)
    END
ELSE
    BEGIN
        IF band = 2 THEN WriteLn('error en la comunicacion');
    END;

    close(textfile); {fin de comunicacion}
    setdrive(disc2);
END;
gotoxy(5,6);
IF band = 1 THEN WriteLn('no hay cable');
REPEAT UNTIL keypressed;
END; {comunica}

```

C.2 LISTADO DEL PROGRAMA EN ASSEMBLER DEL TRS-80
COCO II PARA EL CONTROL DE LOS INTEGRADORES
DIGITALES Y COMUNICACION DE LA MAQUINA DDA

LISTADO DEL PROGRAMA EN ASSEMBLER DE LA TRS-80 COCOII PARA EL CONTROL DE LOS INTEGRADORES DIGITALES DEL ANALIZADOR DIFERENCIAL DIGITAL.

Definición de Variables

DIREC.	CONTENIDO	NOMBRE	TIPO	# BYTES	COMENTARIO
2E2D	3030		FDB	\$3030	
2E2F	3030		FDB	\$3030	
2E31	30		FCB	\$30	
2E32		DATO	RMB	\$5	Dirección que se guardan cinco caracteres numéricos, que es cada dato.
2E37		TABLA	RMB	300	Dirección de memoria que se guardan los caracteres enviados por la PC
2F63		TEMP	RMB	\$1	
2F64		ASCI	RMB	\$50	Dirección donde se almacena los caracteres ascii (5) listos para ser enviados.
2FB4		VALOR	RMB	600	Dirección que se guardan los valores

				binarios para los integradores.
320C	NBIT	RMB	1	Número de bit a ser enviados.
320D	ERROR	RMB	1	Byte que indica si un valor ha sido mal receptado.
320E	RSIN	RMB	1	Dato a ser Recibido.
320F	PART	RMB	1	Setea bit de paridad.
3210	PPAR	RMB	1	Palabra de paridad.
3211	RSOUT	RMB	1	Caracter que recepta de la PC.
3212	NDBUF	RMB	2	Número de datos del Buffer.
3214	DTEM	RMB	2	Contador de datos emitidos.
3216	NDASAL	RMB	1	Número de datos de salida.
3217	TEMPO	RMB	2	Dirección que se guarda en número binario ya transformado
3219	M	RMB	1	Número de Integradores
321A	FIN	RMB	2	Número de Iteraciones
321C	PRINT	RMB	1	Contador de pasos.
321D	PRNT	RMB	1	Número de pasos a emitir un resultado.

321E	SS	RMB	100	Dirección donde se almacenan las conexiones dy entre los integradores.
3282	XS	RMB	20	Dirección donde se almacenan las conexiones dx entre los integradores.
3296	CY	RMB	20	Dirección donde se almacena 'S' o 'N' que indica si queremos la respuesta de dicho integrador.
32AA	SN	RMB	20	Dirección donde se almacena los números de bits (n) de precisión de cada variable (Y).
32BE	SY	RMB	20	Dirección donde se almacena los números de bits (p) de precisión de los diferenciales de la variable (dY).
32D2	R	RMB	40	Dirección donde se almacena el valor integrando de la varia-

					ble.
32FA		Z	RMB	20	Dirección donde se almacena el valor di- ferencial de la res- puesta de la variable (dz).
330E		YY	RMB	40	Dirección donde se almacena las condi- ciones iniciales de los integradores.
3336		DA	RMB	20	Dirección donde se almacena cuantas co- necciones dy tiene cada integrador.
334A	0000	ICTR	FDB	0	Contador de iteracio- nes.
334C		I	RMB	1	Contador General.
334D		J	RMB	1	Contador General.
334E		DY	RMB	20	Dirección donde se almacena el valor a- cumulado de los dife- renciales dy.
3362		POS	RMB	1	Posición relativa en un arreglo de dos di- mensiones.
3363		SIGN	RMB	1	Indicador de signo.
3364		J1	RMB	1	Contador General.

3365	XI	RMB	20	Dirección donde se almacena los valores de los diferenciales dx.
3379	Q	RMB	1	Valor relativo que indica las conecciones de cada integrador.
337A	AA	RMB	1	Parámetro de entrada dx.
337B	BB	RMB	1	Parámetro de entrada dy.
337C	CCY	RMB	2	Parámetro de entrada -salida representa el valor actualizado de la variable Y.
337E	YTEMP	RMB	2	
3380	DD	RMB	2	Parámetro de entrada -salida representa el valor actualizado del integrando R.
3382	E	RMB	1	Parámetro de entrada - salida dz.
3383	F	RMB	1	Parámetro de entrada n.
3384	G	RMB	1	Parámetro de entrada p.

3385		L	RMB	1	Longitud en bits del registro R.
3386		RESTA	RMB	2	
3388		DIVD	RMB	\$3	
338B		RS	RMB	\$2	
338D	2710	K	FDB	10000	Potencias de 10.
338F	03E8		FDB	1000	
3391	0064		FDB	100	
3393	000A		FDB	10	
3395	0001		FDB	1	
	0000	LAST	SET	0	Dirección para fin de tabla.
	0002	COUNT	SET	LAST+2	Contador decimal.
	0003	SAVEA	SET	COUNT+1	Dirección para almacenar A.
	0004	NBYTES	SET	SAVEA+1	Número de bytes para variables locales.

Proceso de Recepción de Información para Interconexión de Integradores, escalamiento y Valores Iniciales.
--

DIREC.	CONTENIDO	ETIQUETA	INSTRUCCION	COMENTARIO
3397	10CE 4000	START	LDS #\$4000	Def. Puntero de Pila
339B	86 02	RECIB	LDA #\$02	Inicialización de
339D	B7 FF20		STA \$FF20	puertos.
33A0	8E 39B1		LDX #LLAMA	Definición de inte-

33A3	BF	010D		STX	\$010D	rrupción.
33A6	7F	3210		CLR	PPAR	Encero palabra de paridad.
33A9	108E	2E37		LDY	#TABLA	Puntero a TABLA.
33AD	BD	3957	SIGR	JSR	RCBCA	Recibo caracter.
33B0	B6	3211		LDA	RSOUT	
33B3	81	54		CMPA	#\$54	Hasta que sea " T "
33B5	26	F6		BNE	SIGR	
33B7	86	52		LDA	#\$52	
33B9	BD	38D9		JSR	FIJPAR	Fijo Paridad de "R"
33BC	86	52		LDA	#\$52	
33BE	BD	3865		JSR	TRANSM	Transmito " R "
33C1	BD	3902	REC2	JSR	RCBC2	Recibo caracter.
33C4	B6	3211		LDA	RSOUT	Si no existe cable
33C7	4D			TSTA		vuelvo a receptar.
33C8	1027	FFCF		LBEQ	RECIB	
33CC	81	59		CMPA	#\$59	Si es "Y", fin de
33CE	27	0A		BEQ	SALREC	archivo, salga.
33D0	A7	A0		STA	,Y+	Guardo caracter
33D2	B8	3210		EORA	PPAR	Almaceno palabra de
33D5	B7	3210		EORA	PPAR	paridad
33D8	20	E7		BRA	REC2	Continuo recibiendo.
33DA	12		SALREC	NOP		

Conversión de caracteres " ASCII " a Binario

33DB	8E	2E37	ST3	LDX	#TABLA	Puntero a TABLA
33DE	108E	2FB4		LDY	#VALOR	Puntero a VALOR
33E2	7F	3363		CLR	SIGN	Encero signo
33E5	A6	84	INTAB	LDA	,X	Si caracter es "F" se
33E7	81	46		CMPA	#\$46	terminó la tabla
33E9	27	68		BEQ	FINTAB	
33EB	CE	2E32		LDU	#DATO	Puntero a Dato
33EE	A6	84	WHI2	LDA	,X	Si caracter es "L"
33F0	81	4C		CMPA	#\$4C	fin de linea.
33F2	27	1D		BEQ	FINLAZ	
33F4	81	20		CMPA	#\$20	Si caracter es "b"
33F6	27	19		BEQ	FINLAZ	otro dato
33F8	81	53		CMPA	#\$53	Si caracter es "S" o
33FA	27	4F		BEQ	NOCON	"N" no convierto.
33FC	81	4E		CMPA	#\$4E	
33FE	27	4B		BEQ	NOCON	
3400	81	2D		CMPA	#\$2D	Si caracter es "-"
3402	26	07		BNE	NOMEN	valor negativo.
3404	86	FF		LDA	#\$FF	Seteo signo a -1
3406	B7	3363		STA	SIGN	
3409	30	01		LEAX	1,X	Incremento puntero X
340B	A6	80	NOMEN	LDA	,X+	
340D	A7	C0		STA	,U+	Guardo caracter en

340F	20	DD		BRA	WHI2	DATO
3411	30	01	FINLAZ	LEAX	1,X	
3413	A6	1E		LDA	-2,X	
3415	81	53		CMPA	#\$53	Si es "S" o "N" no
3417	27	38		BEQ	NOCON	convierto.
3419	81	4E		CMPA	#\$4E	
341B	27	34		BEQ	NOCON2	
341D	34	10		PSHS	X	
341F	34	20		PSHS	Y	
3421	33	5B		LEAU	-5,U	Actualizo puntero pa-
3423	1F	31		TFR	U,X	ra Subrutina
3425	17	02AC		LBSR	CVDTB	Convierto ASCII a Bi-
3428	35	20		PULS	Y	nario
342A	35	10		PULS	X	
342C	FD	3217		STD	TEMPO	
342F	B6	3363		LDA	SIGN	Si es negativo ob-
3432	81	FF		CMPA	#\$FF	tengo su complemento
3434	26	0E		BNE	NEGTEM	a 2
3436	FC	3217		LDD	TEMPO	
3439	43			COMA		
343A	53			COMB		
343B	C3	0001		ADDD	#1	
343E	FD	3217		STD	TEMPO	
3441	7F	3363		CLR	SIGN	Encero signo
3444	FC	3217	NEGTEM	LDD	TEMPO	Almaceno Binario en
3447	ED	A1		STD	,Y++	"VALOR"
3449	20	9A		BRA	INTAB	

344B	1F	89	NOCON	TFR	A,B	Si no hubo conversión
344D	ED	A1		STD	,Y++	Almaceno "S" o "N"
344F	30	01		LEAX	1,X	
3451	20	92	NOCON2	BRA	INTAB	

Inicialización de valores de Variables y Registros de los Integradores Digitales
--

3453	108E	2FB4	FINTAB	LDY	#VALOR	
3457	EC	A1		LDD	,Y++	
3459	F7	3219		STB	M	Numero de Integrado-
345C	EC	A1		LDD	,Y++	res
345E	FD	321A		STD	FIN	Fin de Simulación
3461	EC	A1		LDD	,Y++	
3463	5C			INCB		
3464	F7	321D		STB	PRNT	Paso de Impresión
3467	C6	01		LDB	#1	
3469	F7	321C		STB	PRINT	Contador de paso de
						impresión
346C	8E	321E		LDX	#SS	Conecciones de Inte-
346F	B6	3219		LDA	M	gradores a entrada dy
3472	B7	334C		STA	I	
3475	86	05	CONG	LDA	#5	
3477	B7	334D		STA	J	
347A	EC	A1	SSR	LDD	,Y++	
347C	E7	80		STB	,X+	
347E	7A	334D		DEC	J	

3481	26	F7		BNE	SSR	
3483	7A	334C		DEC	I	
3486	26	ED		BNE	CONG	
3488	8E	3282		LDX	#XS	Conección de Inte-
348B	B6	3219		LDA	M	gradores a entrada dx
348E	B7	334C		STA	I	
3491	EC	A1	MXS	LDD	,Y++	
3493	E7	80		STB	,X+	
3495	7A	334C		DEC	I	
3498	26	F7		BNE	MXS	
349A	8E	3296		LDX	#CY	Condición de salida.
349D	B6	3219		LDA	M	
34A0	7F	334D		CLR	J	
34A3	B7	334C		STA	I	
34A6	EC	A1	MCY	LDD	,Y++	
34A8	E7	80		STB	,X+	
34AA	C1	53		CMPB	#\$53	
34AC	26	03		BNE	NCY	
34AE	7C	334D		INC	J	
34B1	7A	334C	NCY	DEC	I	
34B4	26	F0		BNE	MCY	
34B6	B6	334D		LDA	J	Contador de Datos de
34B9	B7	3216		STA	NDASAL	Salida
34BC	8E	32AA		LDX	#SN	Factor de Escala n
34BF	B6	3219		LDA	M	
34C2	B7	334C		STA	I	
34C5	EC	A1	MSN	LDD	,Y++	

34C7	E7	80		STB	,X+	
34C9	7A	334C		DEC	I	
34CC	26	F7		BNE	MSN	
34CE	8E	32BE		LDX	#SY	Valor p - bit preci-
34D1	B6	3219		LDA	M	sión de dy
34D4	B7	334C		STA	I	
34D7	EC	A1	MSY	LDD	,Y++	
34D9	E7	80		STB	,X+	
34DB	7A	334C		DEC	I	
34DE	26	F7		BNE	MSY	
34E0	8E	330E		LDX	#YY	Inicialización de re-
34E3	B6	3219		LDA	M	gistros Y
34E6	B7	334C		STA	I	
34E9	EC	A1	MTY	LDD	,Y++	
34EB	ED	81		STD	,X++	
34ED	7A	334C		DEC	I	
34F0	26	F7		BNE	MTY	
34F2	8E	32D2		LDX	#R	Inicialización de re-
34F5	B6	3219		LDA	M	gistros R
34F8	B7	334C		STA	I	
34FB	CC	0000	MR	LDD	#0	
34FE	ED	81		STD	,X++	
3500	7A	334C		DEC	I	
3503	26	F6		BNE	MR	
3505	8E	32FA		LDX	#Z	Inicialización de dz
3508	C6	01		LDB	#1	
350A	E7	8813		STB	19,X	

350D	B6	3219		LDA	M
3510	B7	334C		STA	I
3513	C6	00	MZ	LDB	#0
3515	E7	80		STB	,X+
3517	7A	334C		DEC	I
351A	26	F7		BNE	MZ

Definición del número de conecciones a dy

351C	CC	0000		LDD	#0	Inicialización de
351F	FD	334A		STD	ICTR	contador de Iteraciones
3522	FD	3212		STD	NDBUF	Inicialización de
3525	B6	3216		LDA	NDASAL	contador de datos en
3528	C6	06		LDB	#6	Buffer
352A	3D			MUL		
352B	FD	3214		STD	DTEM	Números de datos emi-
352E	7F	334C	ST2	CLR	I	tidos por iteración
3531	8E	321E		LDX	#SS	Puntero a SS
3534	108E	3336		LDY	#DA	Puntero a DA
3538	B6	334C	SUDY	LDA	I	
353B	B7	3362		STA	POS	
353E	C6	05		LDB	#5	
3540	E7	A6		STB	A,Y	
3542	7F	334D		CLR	J	Contador de Columna
3545	B6	334C		LDA	I	Contador de Fila

3548	C6	05		LDB	#5	
354A	3D			MUL		
354B	F7	3362		STB	POS	Posición en la tabla
354E	F6	334D	NDY	LDB	J	SS
3551	FB	3362		ADDB	POS	
3554	E6	85		LDB	B,X	
3556	5D			TSTB		Si no es 0
3557	27	0C		BEQ	RES1	
3559	7C	334D		INC	J	Incremento contador
355C	F6	334D		LDB	J	de columna
355F	C1	05		CMPB	#5	
3561	27	0A		BEQ	FIJO	
3563	20	E9		BRA	NDY	
3565	B6	334C	RES1	LDA	I	
3568	F6	334D		LDB	J	Guardo contador de
356B	E7	A6		STB	A,Y	columna en contador
356D	B6	334C	FIJO	LDA	I	de número de conex-
3770	4C			INCA		ciones dy
3571	B7	334C		STA	I	Si contador de filas
3574	B1	3219		CMPA	M	es número de integra-
3577	26	BF		BNE	SUDY	dores sale.

Proceso de Transmisión de Datos de Salida
y actualización de incrementos dx, dy

3579	7A	321C	SALGA	DEC	PRINT	Si contador de paso
357C	26	0C		BNE	SEGUIR	de impresión es igual

357E	B6	321D		LDA	PRNT	al número de paso de
3581	B7	321C		STA	PRINT	impresión, se inicia-
						liza transmisión de
						datos
3584	17	031E		LBSR	CONVER	Subrutina de prepa-
						ración de datos para
						transmisión.
3587	17	01F1	ST4	LBSR	SALIDA	Subrutina de trans-
						misión de datos
358A	FC	334A	SEGUIR	LDD	ICTR	Incremento contador
358D	C3	0001		ADDD	#1	de Iteraciones
3590	FD	334A		STD	ICTR	
3593	8E	3336		LDX	#DA	Puntero a DA
3596	108E	334E		LDY	#DY	Puntero a DY
359A	CE	321E		LDU	#SS	Puntero a SS
359D	7F	334C		CLR	I	Contador de filas
35A0	7F	334D		CLR	J	Contador de columnas
35A3	B6	334C	ADDY	LDA	I	
35A6	E6	86		LDB	A,X	
35A8	F7	3364		STB	J1	Contador de número de
35AB	6F	A6		CLR	A,Y	conecciones a dy
35AD	5D			TSTB		Si es cero no sume
35AE	27	4C		BEQ	NOSUM	
35B0	B6	334C		LDA	I	
35B3	C6	05		LDB	#5	
35B5	3D			MUL		
35B6	F7	3362		STB	POS	Posición en la tabla

35B9	7F	334D		CLR	J	SS.
35B6	F6	334D	LADY	LDB	J	
35BF	FB	3362		ADDB	POS	
35C2	E6	C5		LDB	B,U	Chequeo signo de la
35C4	1D			SEX		conección
35C5	4D			TSTA		
36C6	27	04		BEQ	NOABS	Si es negativo obten-
35C8	53			COMB		go el valor absoluto
35C9	5C			INCB		
35CA	20	01		BRA	SIGNDY	
35CC	4C		NOABS	INCA		
35CD	B7	3363	SIGNDY	STA	SIGN	Almaceno el signo
35D0	F7	3379		STB	Q	Almaceno en Q el nú-
35D3	B6	3379		LDA	Q	mero del integrador
35D6	4A			DECA		
35D7	81	63		CMPA	#99	Si es 101 escojo la
35D9	2D	02		BLT	Q20	variable independien-
35DB	86	13		LDA	#19	te
35DD	34	10	Q20	PSHS	X	
35DF	8E	32FA		LDX	#Z	Obtengo dz
35E2	E6	86		LDB	A,X	
35E4	B6	3363		LDA	SIGN	
35E7	3D			MUL		Multiplico por el
35E8	35	10		PULS	X	signo.
35EA	B6	334C		LDA	I	
35ED	EB	A6		ADDB	A,Y	Lo acumulo en dy
35EF	E7	A6		STB	A,Y	

35F1	7C	334D		INC	J	
35F4	B6	334D		LDA	J	Hasta que termine el
35F7	B1	3364		CMPA	J1	contador de coneccio-
35FA	26	C0		BNE	LADY	nes a dy.
35FC	B6	334C	NOSUM	LDA	I	Contador de fila.
35FF	34	10		PSHS	X	
3601	8E	3282		LDX	#XS	Puntero a XS.
3604	E6	86		LDB	A,X	Cargo número de co-
3606	1D			SEX		nección dx.
3607	35	10		PULS	X	
3609	4D			TSTA		Chequeo signo.
360A	27	02		BEQ	POST	
360C	53			COMB		Si es negativo ob-
360D	5C			INCB		tengo valor absoluto.
360E	B7	3363	POST	STA	SIGN	Almaceno en Q el nú-
3611	F7	3379		STB	Q	mero del integrador.
3614	34	10		PSHS	X	
3616	C1	63		CMPB	#99	Si es 101 es variable
3618	2E	1E		BGT	DX1	independiente.
361A	F6	3379		LDB	Q	
361D	5A			DECB		
361E	8E	32FA		LDX	#Z	
3621	7D	3363		TST	SIGN	Si el signo es nega-
3624	27	06		BEQ	IGUAL	tivo.
3626	A6	85		LDA	B,X	Obtengo valor absolu-
3628	43			COMA		to de la conexión.
3629	4C			INCA		

362A	A7	85		STA	B,X	
362C	A6	85	IGUAL	LDA	B,X	
362E	F6	334C		LDB	I	
3631	8E	3365		LDX	#XI	Almaceno en dx el dz
3634	A7	85		STA	B,X	correspondiente.
3636	20	14		BRA	CLAZO	
3638	8E	3365	DX1	LDX	#XI	Si la variable es in-
363B	F6	334C		LDB	I	dependiente, dx es el
363E	86	01		LDA	#1	incremento 1.
3640	A7	85		STA	B,X	
3642	7D	3363		TST	SIGN	
3645	27	05		BEQ	CLAZO	Si es negativo, dx es el incremento -1.
3647	B6	3363		LDA	SIGN	
364A	A7	85		STA	B,X	
364C	35	10	CLAZO	PULS	X	
364E	7C	334C		INC	I	Si contador de filas
3651	B6	334C		LDA	I	es número de integra-
3654	B1	3219		CMPA	M	dores finaliza la ac-
3657	1026	FF48		LBNE	ADDY	tualización de incre- mentos.

Proceso de Integración en la máquina DDA
--

365B	7F	334C		CLR	I	Contador de integra-
365E	8E	3365	DDA	LDX	#XI	dores.

3661	F6	334C	LDB	I	
3664	A6	85	LDA	B,X	Entrada del incremen-
3666	B7	337A	STA	AA	to dx.
3669	8E	334E	LDX	#DY	Entrada del incremen-
366C	A6	85	LDA	B,X	to dy.
366E	B7	337B	STA	BB	
3671	8E	330E	LDX	#YY	Entrada del registro
3674	58		ASLB		Y.
3675	EE	85	LDU	B,X	
3677	FF	337C	STU	CCY	
367A	8E	32D2	LDX	#R	Entrada del registro
367D	EE	85	LDU	B,X	R.
367F	FF	3380	STU	DD	
3682	57		ASRB		
3683	8E	32FA	LDX	#Z	Entrada del incremen-
3686	A6	85	LDA	B,X	to dz.
3688	B7	3382	STA	E	
368B	8E	32AA	LDX	#SN	Entrada del factor de
368E	A6	85	LDA	B,X	escalamiento n.
3690	B7	3383	STA	F	
3693	8E	32BE	LDX	#SY	Entrada de p - bit de
3696	A6	85	LDA	B,X	precisión del dy.
3698	B7	3384	STA	G	
369B	17	0069	LBSR	MODTRZ	Ejecución de la inte-
369E	F6	334C	LDB	I	gración de un inte-
36A1	58		ASLB		gradador digital.
36A2	8E	330E	LDX	#YY	Actualización del re-

36A5	FE	337C	LDU	CCY	gistro Y.
36A8	EF	85	STU	B,X	
36AA	8E	32D2	LDX	#R	Actualización del re-
36AD	FE	3380	LDU	DD	gistro R.
36B0	EF	85	STU	B,X	
36B2	57		ASRB		
36B3	8E	32FA	LDX	#Z	Actualización del in-
36B6	B6	3382	LDA	E	cremento dz.
36B9	A7	85	STA	B,X	
36BB	7C	334C	INC	I	Paso del proceso al
36BE	B6	334C	LDA	I	siguiente integrador.
36C1	B1	3219	COMPA	M	Fin del proceso a los
36C4	26	98	BNE	DDA	integradores.
36C6	FC	334A	LDD	ICTR	
36C9	10B3	321A	COMPD	FIN	Si llego a fin de
36CD	1025	FFA8	LBLO	SALGA	proceso de simulación
36D1	16	FCC7	LBRA	RECIB	Vuelve a inicio de
					Programa.

Convertir dato ASCII (5 bytes) en dirección X al número binario en registro D
--

36D4	108E	0005	CVDTB	LDY	#5	Número de bytes a convertir.
36D8	4F		CLRA			Encero A.
36D9	5F		CLRB			Encero B.
36DA	34	02	PSHS	A		Guardo A.

36DC	34	04	C1	PSHS	B	Guardo B.
36DE	E6	80		LDB	,X+	Cargo número binario.
36E0	C0	30		SUBB	#\$30	Convierto ASCII.
36E2	EB	E0		ADDB	,S+	Sumo con anterior
36E4	89	00		ADCA	#0	número.
36E6	24	02		BCC	C3	Chequeo si mayor a 0.
36E8	6C	E4		INC	,S	Incremento dato bina- rio.
36EA	31	3F	C3	LEAY	-1,Y	Decremento puntero y.
36EC	1027	0009		LBEQ	C2	Si es cero salgo.
36F0	17	0009		LBSR	MUL10	Multiplico por 10 bi- nario.
36F3	24	E7		BCC	C1	Si es mayor a 0 con- tinuo proceso.
36F5	6C	E4		INC	,S	Incremento dato bi- nario.
36F7	20	E3		BRA	C1	Continuo proceso.
36F9	66	E0	C2	ROR	,S+	Divido para 2 dato
36FB	39			RTS		binario.

Subrutina multiplica por 10 número binario en D

36FC	34	06	MUL10	PSHS	D	Guardo D.
36FE	58			ASLB		Multiplico por 2.
36FF	49			ROLA		
3700	58			ASLB		Multiplico por 2.

3701	49		ROLA		
3702	E3	E1	ADDD	,S++	Multiplico por 4.
3704	58		ASLB		Multiplico por 2.
3705	49		ROLA		
3706	39		RTS		

Subrutina proceso de integración en un integrador digital

3707	B6	3383	MODTRZ	LDA	F	Cargo n
370A	BB	3384		ADDA	G	Cargo p
370D	B7	3385		STA	L	Sumo n y p y obtengo longitud de registro.
3710	F6	367B		LDB	BB	Cargo dy en registro
3713	1D			SEX		D.
3714	F3	337C		ADDD	CCY	Sumo dy al registro Y
3717	FD	337C		STD	CCY	
371A	F6	337B		LDB	BB	Divido para 2 dy.
371D	57			ASRB		
371E	1D			SEX		
371F	F3	337C		ADDD	CCY	Sumo al registro Y
3722	FD	337E		STD	YTEMP	(Integración trapezoidal).
3725	B6	337A		LDA	AA	Chequeo signo dx.
3728	4D			TSTA		
3729	2B	0D		BMI	NEG	Si es negativo.
372B	27	1A		BEQ	RABS	Si es cero

372D	FC	337E		LDD	YTEMP	Sumo al Y actual y
3730	F3	3380		ADDD	DD	obtengo Y*
3733	FD	3380		STD	DD	
3736	20	0F		BRA	RABS	
3738	FC	337E	NEG	LDD	YTEMP	Resto al Y actual y
373B	FD	3386		STD	RESTA	obtengo Y*
373E	FC	3380		LDD	DD	
3741	B3	3386		SUBD	RESTA	
3744	FD	3380		STD	DD	
3747	B6	3385	RABS	LDA	L	Cargo longitud de
374A	B7	334D		STA	J	registro R
374D	FC	3380		LDD	DD	
3750	47		RDL	ASRA		Obtengo valor máximo
3751	56			RORB		del registro R.
3752	7A	334D		DEC	J	
3755	26	F9		BNE	RDL	
3757	F7	3382		STB	E	Si es mayor dz = 1
375A	5D			TSTB		Si es cero retorno
375B	27	1D		BEQ	RETOR	
375D	B6	3385		LDA	L	Si es negativo ac-
3760	B7	334D		STA	J	tualizo valor de R.
3763	F6	3382		LDB	E	
3766	1D			SEX		
3767	58		LDR	LSLB		Valor de R* es valor
3768	49			ROLA		máximo de R menos R.
3769	7A	334D		DEC	J	
376C	26	F9		BNE	LDR	

376E	FD	3386	STD	RESTA
3771	FC	3380	LDD	DD
3774	B3	3386	SUBD	RESTA
3777	FD	3380	STD	DD
377A	39	RETOR	RTS	

Subrutina de transmisión de datos

377B	4F		SALIDA	CLRA	Inicializo puertos
377C	B7	FF20		STA	\$FF20
377F	8E	39B1		LDX	#LLAMA Definición interrup-
3782	BF	010D		STX	\$010D ción.
3785	86	54	SIGT2	LDA	#\$54 Cargo " T "
3787	BD	38D9		JSR	FIJPAR Fijo paridad
378A	86	54		LDA	#\$54
378C	BD	3865		JSR	TRANSM Transmito " T "
378F	86	05		LDA	#5 Contador de espera
3791	B7	334C		STA	I
3794	86	02	SIGT	LDA	#\$02 Fijo nivel alto en
3796	B7	FF20		STA	\$FF20 puerto de salida.
3799	BD	3902		JSR	RCBC2 Recibo carácter.
379C	B6	3211		LDA	RSOUT
379F	81	52		CMPA	#\$52 Hasta que sea " R "
37A1	27	08		BEQ	SALT
37A3	7A	334C		DEC	I Decremento número de
37A6	26	EC		BNE	SIGT espera.

37A8	16	FBEC		LBRA	START	Regresa al inicio por no recepción.
37AB	7F	3210	SALT	CLR	PPAR	Encero palabra de pa- ridad.
37AE	8E	2F64		LDX	#ASCI	Puntero a caracteres a ser transmitidos.
37B1	B6	3216		LDA	NDASAL	Cargo número de datos
37B4	B7	334D		STA	J	de salida.
37B7	86	05	SITRA	LDA	#5	Cargo contador de ca-
37B9	B7	334C		STA	I	racteres por dato.
37BC	A6	84	NUMTR	LDA	,X	Cargo carácter a transmitir.
37BE	BD	38D9		JSR	FIJPAR	Fijo paridad
37C1	A6	84		LDA	,X	
37C3	B8	3210		EORA	PPAR	Actualizo palabra de
37C6	B7	3210		STA	PPAR	paridad
37C9	A6	80		LDA	,X+	
37CB	BD	3865		JSR	TRANSM	Transmito carácter
37CE	7A	334C		DEC	I	Decremento contador
37D1	26	E9		BNE	NUMTR	Si no es cero sigo transmitiendo.
37D3	7A	334D		DEC	J	Decremento contador de datos.
37D6	27	02		BEQ	NOTRA	Si es cero dejo de transmitir.
37D8	20	DD		BRA	SITRA	
37DA	FC	334A	NOTRA	LDD	ICTR	Chequeo si es fin del

37DD	FB	321D		ADDB	PRNT	proceso de Simulación
37E0	89	00		ADCA	#0	
37E2	10B3	321A		CPMD	FIN	
37E6	25	22		BLO	NOFIN	
37E8	86	1A		LDA	#\$1A	Cargo fin de archivo
37EA	BD	38D9		JSR	FIJPAR	Fijo paridad
37ED	86	1A		LDA	#\$1A	
37EF	B8	3210		EORA	PPAR	Actualizo palabra de
37F2	B7	3210		STA	PPAR	paridad.
37F5	86	1A		LDA	#\$1A	
37F7	BD	3865		JSR	TRANSM	Transmito fin de ar- chivo.
37FA	B6	3210		LDA	PPAR	Cargo palabra de pa-
37FD	4C			INCA		ridad.
37FE	BD	38D9		JSR	FIJPAR	Fijo Paridad.
3801	B6	3210		LDA	PPAR	
3804	4C			INCA		
3805	BD	3865		JSR	TRANSM	Transmito palabra de paridad.
3808	20	5A		BRA	FINTRA	Fin de Transmisión
380A	86	0D	NOFIN	LDA	#\$0D	Cargo fin de líneas
380C	BD	38D9		JSR	FIJPAR	Fijo Paridad.
380F	86	0D		LDA	#\$0D	
3811	B8	3210		EORA	PPAR	Actualizo palabra de
3814	B7	3210		STA	PPAR	paridad.
3817	86	0D		LDA	#\$0D	
3819	BD	3865		JSR	TRANSM	Transmito fin de lí-

				nea.
381C	86	0A	LDA	#\$0A Cargo fin de línea
381E	BD	38D9	JSR	FIJPAR Fijo Paridad
3821	86	0A	LDA	#\$0A
3823	B8	3210	EORA	PPAR Actualizo palabra de
3826	B7	3210	STA	PPAR paridad.
3829	86	0A	LDA	#\$0A
382B	BD	3865	JSR	TRANSM Transmito fin de lí-
				nea.
382E	B6	3210	LDA	PPAR Cargo palabra de pa-
3831	4C		INCA	ridad.
3832	BD	38D9	JSR	FIJPAR Fijo Paridad.
3835	B6	3210	LDA	PPAR
3838	4C		INCA	
3839	BD	3865	JSR	TRANSM Transmito palabra de
				paridad.
383C	FC	3212	LDD	NDBUF Cargo contador de nú-
383F	83	0001	SUBD	#1 mero de datos en el
3842	FD	3212	STD	NDBUF buffer.
3845	FC	3214	LDD	DTEM Incremento número de
3848	F3	3212	ADDD	NDBUF datos en buffer con
				número de datos emi-
				tidos.
384B	1083	5000	CMPD	#\$5000 Comparo con máximo
384F	25	13	BLO	FINTRA número de datos en
				Buffer.
3851	86	59	LDA	#\$59 Cargo carácter " Y ".

3853	BD	38D9	JSR	FIJPAR	
3856	86	59	LDA	#\$59	Transmito carácter
3858	BD	3865	JSR	TRANSM	"Y" código de espera.
385B	BD	38F5	JSR	DELAY	Retardo de tiempo.
385E	CC	0000	LDD	#0	
3861	FD	3212	STD	NDBUF	Encero número de da- tos en Buffer.
3864	39	FINTRA	RTS		Fin de transmisión.

Subrutina para la transmisión de un carácter
--

3865	49	TRANSM	ROLA		Primer bit eliminado.	
3866	B7	3211	STA	RSOUT	Guardo caracter a transmitir en RSOUT.	
3869	C6	09	LDB	#\$09	Seteo de tiempo de espera.	
386B	86	00	LDA	#\$00	Bit de parada.	
386D	B7	FF20	STA	\$\$\$FF20	Transmitir bit.	
3870	BD	39A0	JSR	ESPR	Tiempo de espera.	
3873	86	08	LDA	#\$08	Contador de número de	
3875	B7	320C	STA	NBIT	bits.	
3878	B6	3211	ENBIT	LDA	RSOUT	Conversión paralelo serie.
387B	B7	FF20	STA	\$\$\$FF20	Transmito bit.	
387E	BD	39A0	JSR	ESPR	Tiempo de espera.	
3881	46		RORA		Roto bits.	

3882	B7	3211	STA	RSOUT	Actualizo RSOUT.
3885	7A	320C	DEC	NBIT	Decremento contador de bit
3888	26	EE	BNE	ENBIT	Si no es cero sigo transmitiendo.
388A	B6	320F	LDA	PART	
388D	B7	FF20	STA	\$FF20	Transmito bit de pa- ridad.
3890	BD	39A0	JSR	ESPR	Tiempo de espera.
3893	86	02	LDA	#\$02	
3895	B7	FF20	STA	\$FF20	Transmito bit de pa- ridad.
3898	BD	39A0	JSR	ESPR	Tiempo de espera.
389B	FC	3212	LDD	NDBUF	Incremento número de datos enviados al Buffer.
389E	C3	0001	ADDD	#1	
38A1	FD	3212	STD	NDBUF	
38A4	39		RTS		

Subrutina de preparación de datos en dirección de memoria ASCI para transmisión.

38A5	8E	2F64	CONVER	LDX	#ASCI	Puntero a ASCI.
38A8	108E	330E		LDY	#YY	Puntero a registros Y
38AC	7F	334C		CLR	I	Encera contador de

						datos.
38AF	CE	3296	YYPUB	LDU	#CY	Puntero a condición
38B2	B6	334C		LDA	I	de salida.
38B5	E6	C6		LDB	A,U	
38B7	C1	53		CMPB	#\$53	Chequeo si es "S".
38B9	26	12		BNE	PYY	
38BB	48			ASLA		Cargo a registro D el
38BC	EE	A6		LDU	A,Y	registro Y del inte-
38BE	1F	30		TFR	U,D	grador digital.
38C0	34	20		PSHS	Y	
38C2	34	10		PSHS	X	
38C4	17	00F9		LBSR	CVBTD	Convierto binario en
38C7	35	10		PULS	X	caracteres ASCII (5
38C9	35	20		PULS	Y	bytes).
38CB	30	05		LEAX	5,X	Sumo 5 bytes a X.
38CD	7C	334C	PYY	INC	I	Incremento contador.
38D0	B6	334C		LDA	I	Hasta que sea número
38D3	B1	3219		CMPA	M	de integradores.
38D6	26	D7		BNE	YYPUB	Fin de preparación
38D8	39			RTS		

Subrutina que fija el bit de paridad impar para un byte que se va a transmitir.
--

38D9	7F	320F	FIJPAR	CLR	PART	Encero byte de paridad
38DC	C6	08		LDB	#\$08	Número de bit a fijar

					paridad.
38DE	44		ROTE	LSRA	Chequeo si bit es 1
38DF	24	03		BCC	IPAR
38E1	7C	320F		INC	PART Si lo es, incremento
38E4	46		IPAR	RORA	byte de paridad.
38E5	5A			DECB	Hasta que se chequeen
38E6	26	F6		BNE	ROTE todos los bits.
38E8	7C	320F		INC	PART Incremento para pari-
38EB	B6	320F		LDA	PART dad impar.
38EE	84	01		ANDA	#\$01 Enmascaro el bit 1.
38F0	49			ROLA	
38F1	B7	320F		STA	PART Almaceno en PART so-
38F4	39			RTS	lamente el bit de pa-
					ridad y retorno.

Subrutina de Retardo de tiempo

38F5	8E	FFFF	DELAY	LDX	#\$FFFF	Máximo valor de X
38F8	C6	06	DELA3	LDB	#6	Número de veces del
38FA	5A		DELA2	DECB		valor máximo.
38FB	26	FD		BNE	DELA2	
38FD	30	1F		LEAX	-1,X	
38FF	26	F7		BNE	DELA3	Fin de retardo.
3901	39			RTS		

Subrutina de Recepción de carácter, sensa si existe cable

3902	8E	4000	RCBC2	LDX	#\$0000	Máximo retardo de
3905	30	1F	BITS	LEAX	-1,X	tiempo.
3907	27	4A		BEQ	NOCAR	Si se termina no hay recepción.
3909	B6	FF22		LDA	\$\$FF22	Inicialización de puerto de entrada.
390C	84	01		ANDA	#\$01	Máscara para el bit 1
390E	26	F5		BNE	BITS	Chequea bit de inicio
3910	C6	04		LDB	#\$04	Mitad de retardo para el muestreo de bit.
3912	BD	39A0		JSR	ESPR	
3915	86	08		LDA	#\$08	
3917	B7	320C		STA	NBIT	Contador de bit igual a 8.
391A	4F			CLRA		
391B	7F	320F		CLR	PART	Encero bit de paridad
391E	C6	09		LDB	#\$09	Retardo de muestreo de bits.
3920	BD	39A0	RCBIT2	JSR	ESPR	
3923	74	320E		LSR	RSIN	Chequea bit de llegada
3926	24	03		BCC	CPAR2	Si es 1 incrementa contador de paridad.
3928	7C	320F		INC	PART	
392B	46		CPAR2	RORA		
392C	7A	320C		DEC	NBIT	Decrementa contador de bit.

392F	26	EF		BNE	RCBIT2	Hasta que sea cero
3931	B7	3211		STA	RSOUT	Almaceno carácter re-
3934	BD	39A0		JSR	ESPR	cibido.
3937	4F			CLRA		
3938	74	320F		LSR	PART	Cargo bit de paridad
393B	46			RORA		del carácter recibido
393C	B7	320F		STA	PART	
393F	4F			CLRA		
3940	74	320E		LSR	RSIN	Cargo bit de paridad
3943	46			RORA		receptado.
3944	B8	320F		EORA	PART	Si no son iguales
3947	26	05		BNE	NERRO2	existe error.
3949	86	FF		LDA	#\$FF	
394B	B7	320D		STA	ERR0R	Código de error.
394E	BD	39A0	NERRO2	JSR	ESPR	Recepto bit de parada
3951	20	03		BRA	FCAR	Terminó Recepción.
3953	7F	3211	NOCAR	CLR	RSOUT	Si no existe llegada
3956	39		FCAR	RTS		de carácter encero
						RSOUT.

Subrutina de Recepción de carácter

3957	B6	FF22	RCBCA	LDA	\$FF22	Inicialización de puerto de entrada.
395A	84	01		ANDA	#\$01	Máscara para el bit 1
395C	26	F9		BNE	RCBCA	Chequea bit de inicio

395E	C6	04		LDB	#\$04	Mitad de retardo para
3960	BD	39A0		JSR	ESPR	muestreo de bit.
3963	86	08		LDA	#\$08	Contador de bit igual
3965	B7	320C		STA	NBIT	a 8.
3968	4F			CLRA		
3969	7F	320F		CLR	PART	Encero bit de paridad
396C	C6	09		LDB	#\$09	Retardo de muestreo
396E	BD	39A0	RCBIT	JSR	ESPR	de bit.
3971	74	320E		LSR	RSIN	Chequea bit de llega- da
3974	24	03		BCC	CPAR	Si es 1 incrementa
3976	7C	320F		INC	PART	contador de paridad
3979	46		CPAR	RORA		
397A	7A	320C		DEC	NBIT	Decrementa contador de bit.
397D	26	EF		BNE	RCBIT	hasta que sea cero
397F	B7	3211		STA	RSOUT	Almaceno carácter re-
3982	BD	39A0		JSR	ESPR	cibido.
3985	4F			CLRA		
3986	74	320F		LSR	PART	Cargo bit de paridad
3989	46			RORA		del carácter recibido
398A	B7	320F		STA	PART	
398D	4F			CLRA		
398E	74	320E		LSR	RSIN	Cargo bit de paridad
3991	46			RORA		receptado.
3992	B8	320F		EORA	PART	Si no son iguales
3995	26	05		BNE	NERRO	existe error.

3997	86	FF		LDA	#\$FF	Código de error.
3999	B7	320D		STA	ERROR	
399C	BD	39A0	NERRO	JSR	ESPR	Recepto bit de parada
399F	39			RTS		Fin de Recepción.

Subrutina de retardo de tiempo para muestreo de bits,
depende del valor de Registro B.

39A0	34	06	ESPR	PSHS	D	Guardo D.
39A2	86	37	DCRB	LDA	#\$37	Máscara para la inte-
39A4	B7	FF01		STA	\$\$\$FF01	rrupción.
39A7	3C	EF	HIRQ	CWAI	#\$EF	Habilito interrupción
39A9	12			NOP		Espero interrupción.
39AA	12			NOP		
39AB	5A			DECB		Decremento contador
						de interrupciones.
39AC	26	F4		BNE	DCRB	Hasta que sea 0
39AE	35	06		PULS	D	Recupero D
39B0	39			RTS		Fin de Retardo.

Subrutina de Interrupción IRQ

39B1	B6	FF00	LLAMA	LDA	\$\$\$FF00	
39B4	86	36		LDA	#\$36	Deshabilito interrup-
39B6	B7	FF01		STA	\$\$\$FF01	ción.
39B9	B6	FF22		LDA	\$\$\$FF22	Muestrea RS232 IN

39BC	B7	320E	STA	RSIN	Guardo bit recibido
39BF	3B		RTI		en RSIN

Subrutina de conversión de número binario en Registro D
en dato ASCII (5 bytes) en dirección X

39C0	32	7C	CVBTD	LEAS	-NBYTES,S	Localización para
39C2	31	8D F9D1		LEAY	K+10,PCR	variables locales
39C6	10AF	60		STY	LAST,S	
39C9	31	8D F9C0		LEAY	K,PCR	Puntero Y a cons
39CD	6F	62	CVB1	CLR	COUNT,S	tantes.
39CF	A3	A4	CVB2	SUBD	,Y	
39D1	25	04		BLO	CVB3	
39D3	6C	62		INC	COUNT,S	
39D5	29	F8		BRA	CVB2	
39D7	E3	A1	CVB3	ADDD	,Y++	Recupera D,puntean-
39D9	A7	63		STA	SAVEA,S	do Y a la siguiente constante.
39DB	A6	62		LDA	COUNT,S	Coeficiente decimal en A.
39DD	8B	30		ADDA	#\$30	Conversión a ASCII
39DF	A7	80		STA	,X+	Salida al siguiente
39E1	A6	63		LDA	SAVEA,S	dígito ASCII.
39E3	10AC	60		CMPLY	LAST,S	
39E6	26	E5		BNE	CVB1	
39E8	35	A6		PULS	D,Y,PC	Retorne , Fin de conversión.

MISSING END

00001 TOTAL ERRORS

AA	337A	
ADDY	35A3	
ASCI	2F64	
BB	337B	
BITS	3905	
C1	36DC	
C2	36F9	
C3	36EA	
CCY	337C	
CLAZO	364C	
CONG	3475	
CONVER	38A5	
COUNT	0002	S
CPAR	3979	
CPAR2	392B	
CVB1	39CD	
CVB2	39CF	
CVB3	39D7	
CVBTD	39C0	
CVDTB	36D4	
CY	3296	
DA	3336	
DATO	2E32	
DCRB	39A2	

DD	3380
DDA	365E
DELA2	38FA
DELA3	38F8
DELAY	38F5
DIVD	3388
DTEM	3214
DX1	3638
DY	334E
E	3382
ENBIT	3878
ERROR	320D
ESPR	39A0
F	3383
FCAR	3956
FIJO	356D
FIJPAR	38D9
FIN	321A
FINLAZ	3411
FINTAB	3453
FINTRA	3864
G	3384
HIRQ	39A7
I	334C
ICTR	334A
IGUAL	362C
INTAB	33E5

IPAR	38E4	
J	334D	
J1	3364	
K	338D	
L	3385	
LADY	35BC	
LAST	0000	S
LDR	3767	
LLAMA	39B1	
M	3219	
MCY	34A6	
MODTRZ	3707	
MR	34FB	
MSN	34C5	
MSY	34D7	
MTY	34E9	
MUL10	36FC	
MXS	3491	
MZ	3513	
NBIT	320C	
NBYTES	0004	S
NCY	34B1	
NDASAL	3216	
NDBUF	3212	
NDY	354E	
NEG	3738	
NEGTEM	3444	

NERRO	399C
NERRO2	394E
NOABS	35CC
NOCAR	3953
NOCON	344B
NOCON2	3451
NOFIN	380A
NOMEN	340B
NOSUM	35FC
NOTRA	37DA
NUMTR	37BC
PART	320F
POS	3362
POST	360E
PPAR	3210
PRINT	321C
PRNT	321D
POS	3362
POST	360E
PYY	38CD
Q	3379
Q20	35DD
RSIN	320E
RSOUT	3211
R	32D2
RESTA	3386
RS	338B

RECIB	339B	
REC2	33C1	
RES1	3565	
RABS	3747	
RDL	3750	
RETOR	377A	
ROTE	38DE	
RCBC2	3902	
RCBIT2	3920	
RCBCA	3957	
RCBIT	396E	
SS	321E	
SN	32AA	
SY	32BE	
SIGN	3363	
SAVEA	0003	S
START	3397	
SIGR	33AD	
SALREC	33DA	
ST3	33DB	
SSR	347A	
ST2	352E	
SUDY	3538	
SALGA	3579	
ST4	3587	
SEGUIR	358A	
SIGNDY	35CD	

DELIBER

SALIDA	377B
SIGT2	3785
SIGT	3794
SALT	37AB
SITRA	37B7
TABLA	2E37
TEMP	2F63
TEMPO	3217
TRANSM	3865
VALOR	2FB4
WHI2	33EE
XS	3282
XI	3365
YY	330E
YTEMP	337E
YYPUB	38AF
Z	32FA

BIBLIOGRAFIA

1. BERGAMINI, EDUARDO W. A class of backward - Difference Digital Differential Analyzers for Incremental computer realizations. Technical report No 3606-7. Prepared under U.S. ARMY, U.S. NAVY and U.S. AIR FORCE. 1973
2. CHIA, PEIR YU. Digital filter Design Technique and the realization of transfer and Immitance functions by using Digital Elements. Scientific Report No 22 National Aeronautics. 1967
3. DAWOUD S. H. DAWOUD Y NADIA Z EL-ARABY. Parallel Digital Differential Analyzer with arbitrary stores Interconnections. IEEE Transactions on Computer. Vol C-22 N-1 Enero 73 Pgs 41-46. 1973
4. HAKKALA, LAURY Y OJALA, LEO. Discretization Error Analysis in Linar DDA connections. IEEE Transactions on Computer. Vol C-23 N-9 Septiembre 74 pags 932-941. 1974
5. HERRERO, JOSE LUIS. Synthesis of filters. Englewood Cliffs. Prentice Hall. 1966
6. JOHNSON, DAVID E. Introduction to filter theory. Prentice Hall Electrical Engineering Series. 1976
7. MC. GHEE, ROBERT B Y NILSEN, RAGNAR N. The extended resolution Digital Differential Analyzer. A New

- computing Structure for solving Differential Equations. IEEE Transactions on Computer Vol C-19 N-1, enero 79, pags 1-9. 1979
8. MONROE, A.J. Digital Processes for Sampled Data Systems. pp 40-43. 1971
 9. ODUM, O. HOWARD T. Y ODUM, ELISABETH C. Minimodels and Simulations Exercises. Center for Wetlands University of Florida Gainesville. 1989
 10. PARASURAMAN B. Digital Incremental Computation using Automatic Programming Thecniques. Scientific Report No 40 National Aeronautics and Space Administration. 1972
 11. PETERSON, A.M. Simulation of Digital Integrators. 1972
 12. PHILIPPOT, P. Turbo Pascal para IBM - PC y compatibles. Procedimiento y funciones. Colección Informática de Gestión
 13. SCHULZ, E.J. Y PARASURAMAN, B. The Digital Incremental Computer. A New computing structure for the Numerical Solution of Differential Equation. Scientific Reports No 36. National Aeronautics and Space Administrations. 1971
 14. SCHULZ, E.J. Numerical Solution of Differential Equations. Technical Report No 3606-6 Prepared under U.S. ARMY, U.S. NAVY and U.S. AIR FORCE. 1971
 15. SIZER, T.R.H. C.ENG. M.I.E.E. The Digital Differential Analyser. Chapman and Hall Ltd. 1968
 16. WILLEN, DAVID C Y KRANTZ, JEFREY. 8088 Assembler Language Programming the IBM - PC. Howard W. Sams &

Co. Inc. 1983

17. WHITE, DONALD R.J. Electrical Filters, Synthesis Design and Applications. Don White Consultants Inc. 1980
18. TRS-80 Color Computer Technical Reference Manual. Radio Shack. Tandy Corporation. 1982
19. EDTASM+ Color Computer Editor Assembler with Zbug Tandy Corporation. 1982
20. TURBO PASCAL Reference Manual.