



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 D E D I C A T O R I A

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.

A G R A D E C I M I E N T O

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

No	Pag.
CAPITULO III	
CONECCION Y PRUEBAS	
3.1 DIAGRAMA CIRCUITAL DE LA CONECCION 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 conexiones 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 periódicamente 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:

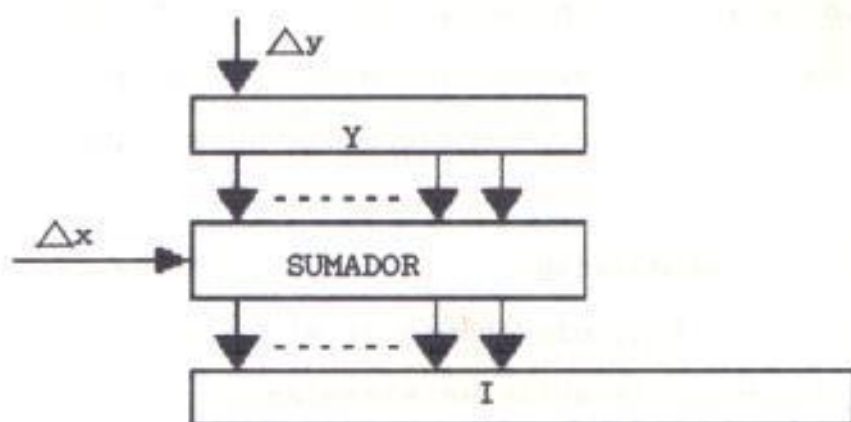


FIG. 1.1 INTEGRADOR DIGITAL

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_1 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.

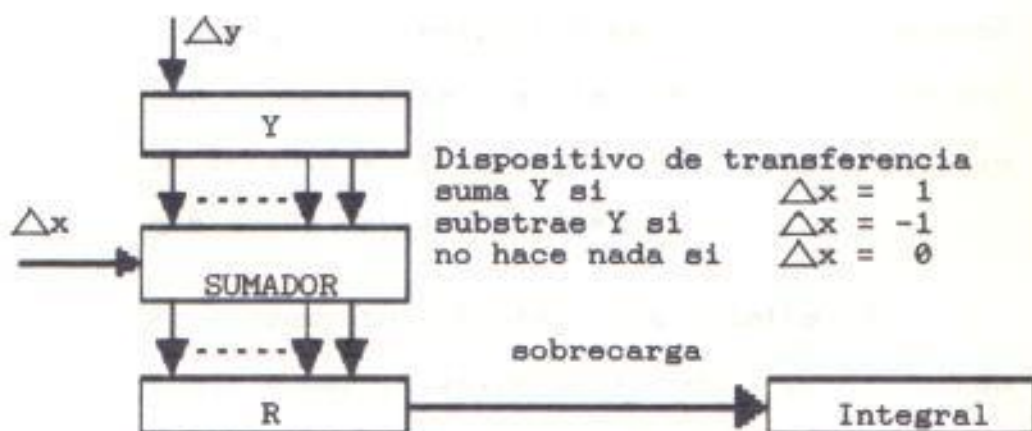


FIG. 1.2 REPRESENTACION DE LA SOBRECARGA EN EL INTEGRADOR DIGITAL.

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:

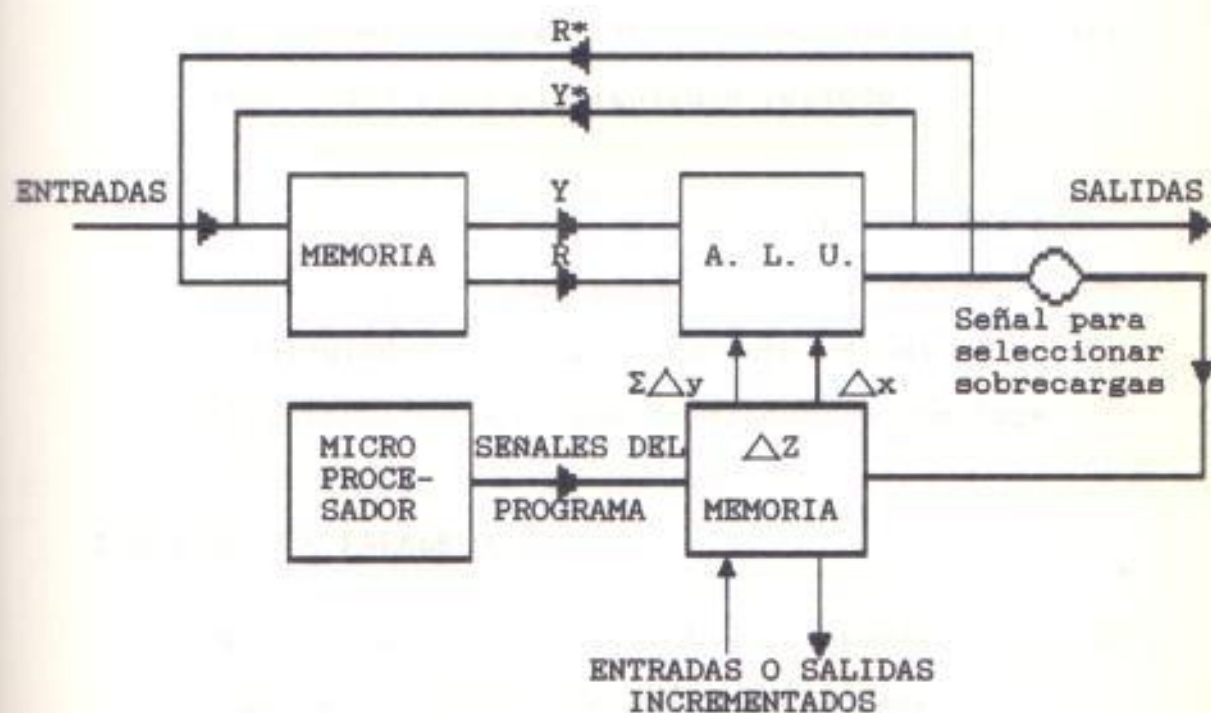


FIG. 1.3 DIAGRAMA DE BLOQUES DEL DDA SERIE

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:

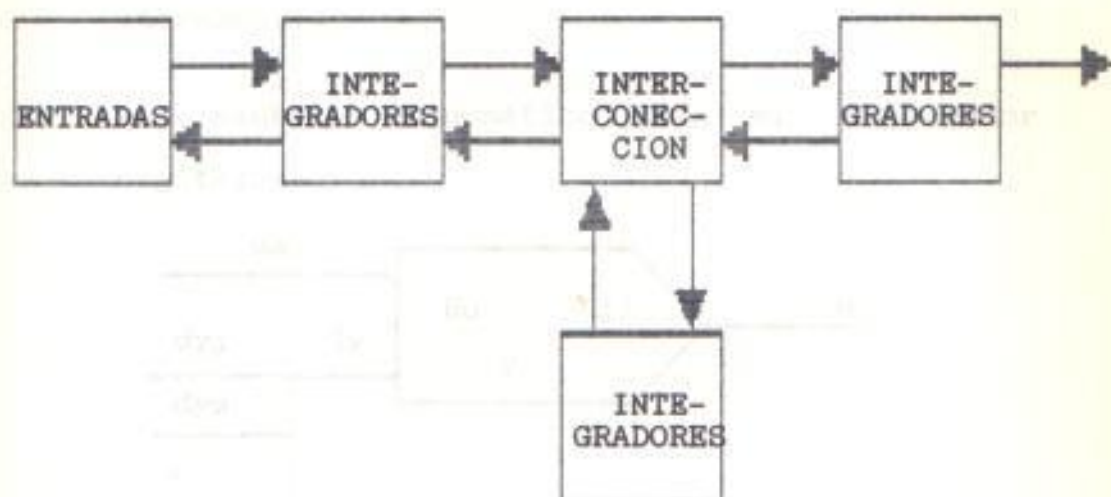


FIG. 1.4 DIAGRAMA DE BLOQUES DEL DDA PARALELO

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:

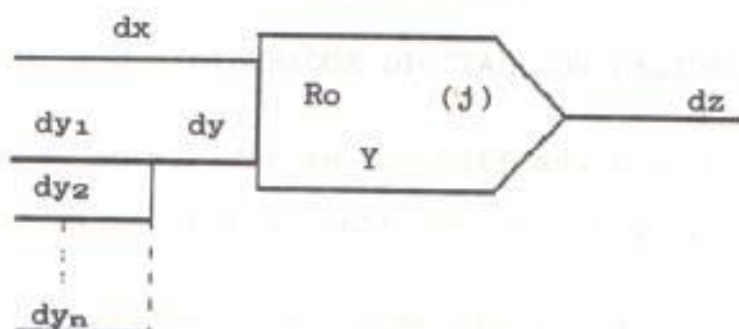


FIG. 1.5 SIMBOLO DEL INTEGRADOR DIGITAL

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:

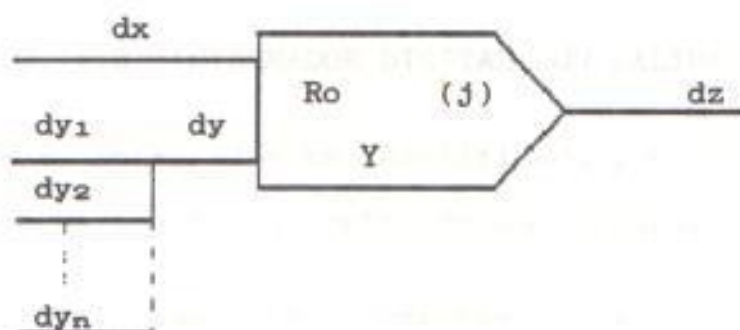


FIG. 1.5 SIMBOLO DEL INTEGRADOR DIGITAL

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:

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

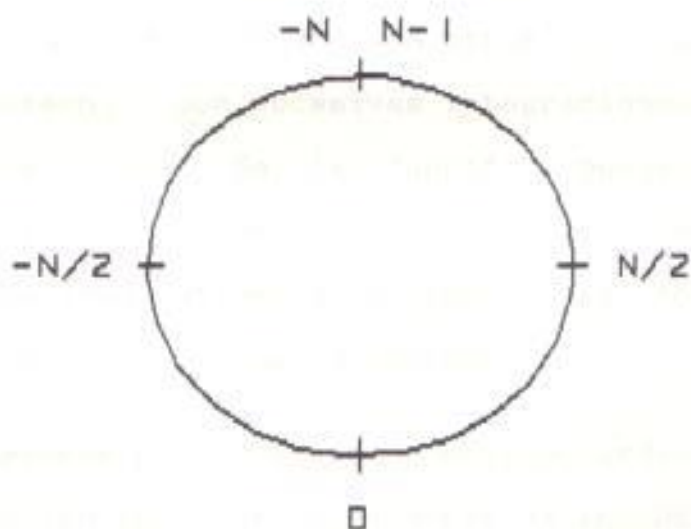


FIG. 1.7 REPRESENTACION CIRCULAR DE NUMEROS BINARIOS EN COMPLEMENTO A DOS.

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 $dv/dt(0) = 0$

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.

1.4.3 ESCALAMIENTO DE LOS INTEGRADORES DIGITALES.⁶

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 dy 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:

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

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^{-z} o sea:

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

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:

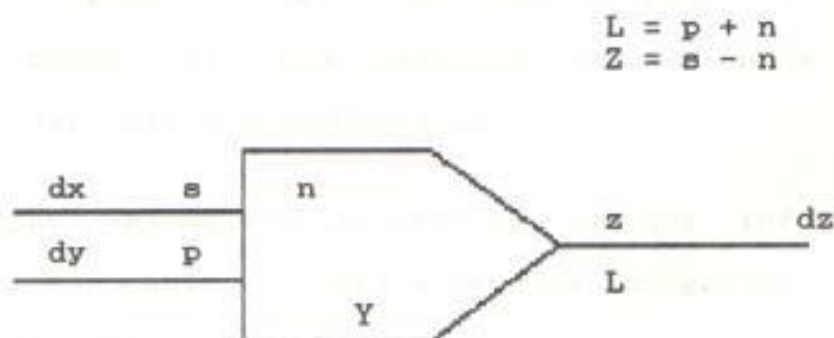


FIG. 1.9 SIMBOLO DEL INTEGRADOR DIGITAL CON PARAMETROS DE ESCALAMIENTOS.

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 más corto integrador debe ser mayor que 0 y el más grande no puede sobrepasar la máxima longitud del registro Y . En nuestro caso que manejamos 16 bits, será 15, puesto que el bit más 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 :

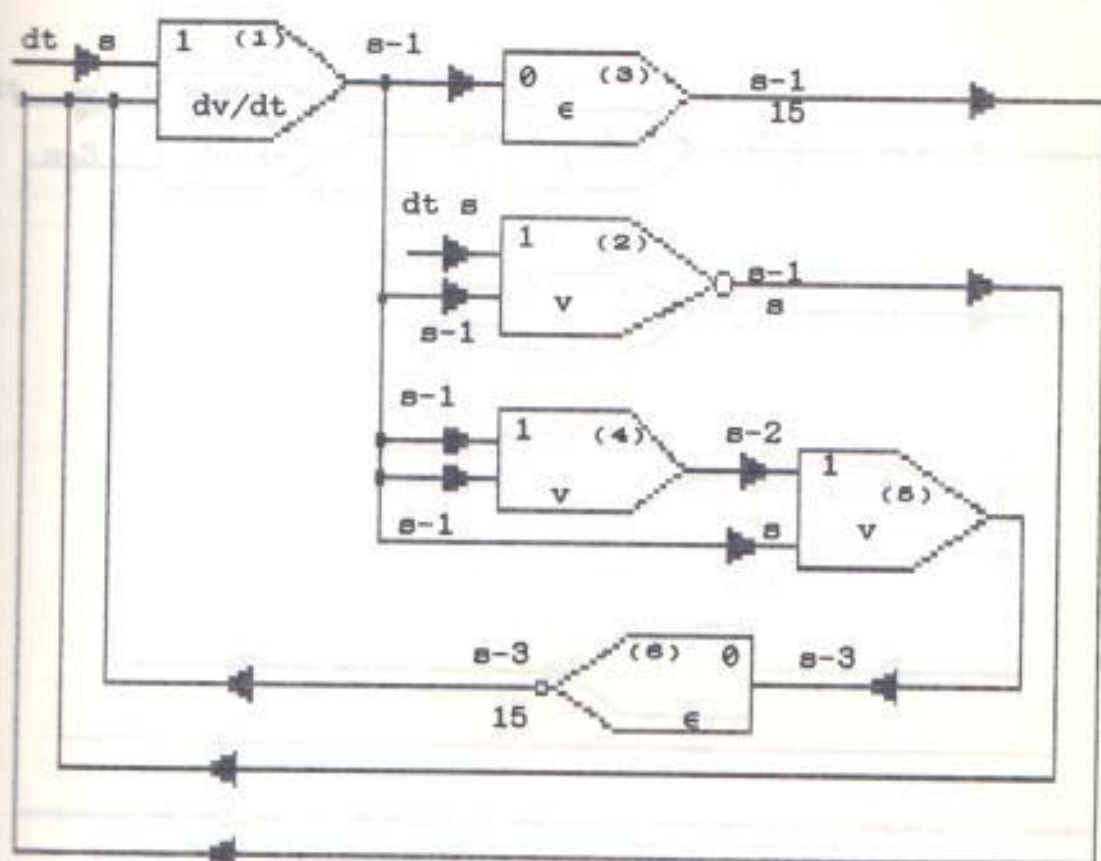


FIG. 1.10 DIAGRAMA CON ESCALAMIENTO PRIMITIVO.

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:

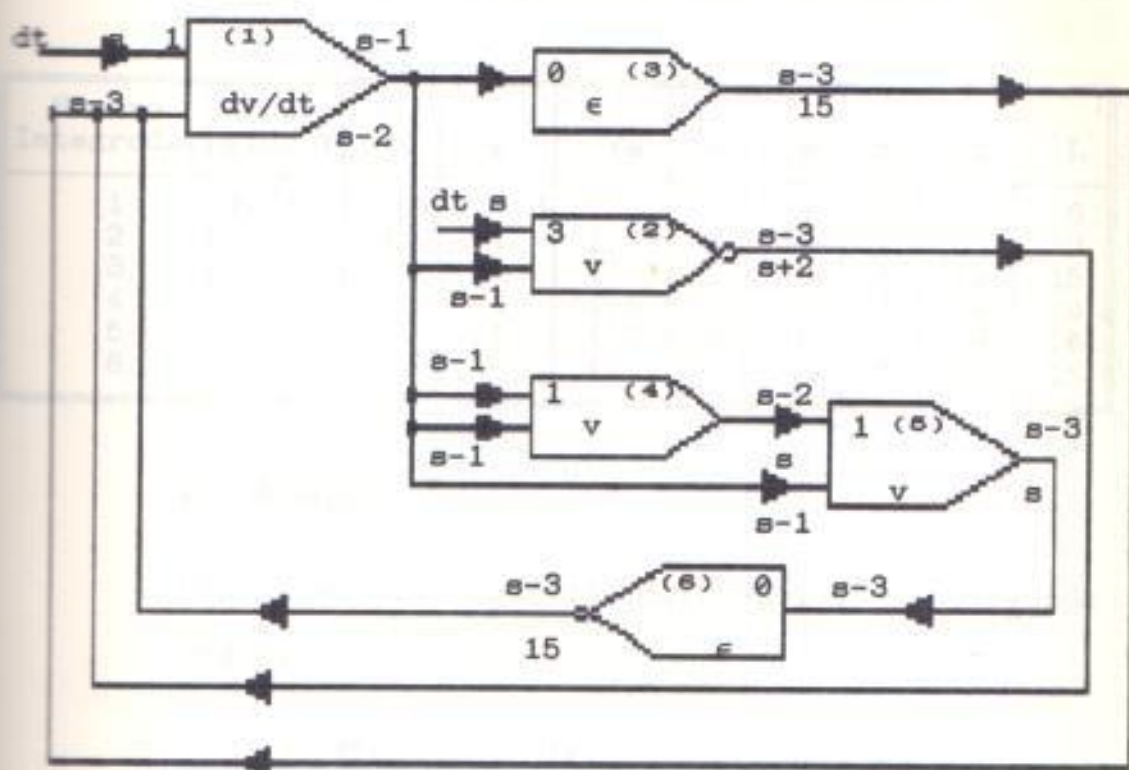


FIG. 1.11 DIAGRAMA DE LA SIMULACION DE LA ECUACION DE VAN DER POL CON ESCALAMIENTO COMPLETO.

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	e	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

2^{do} 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,

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

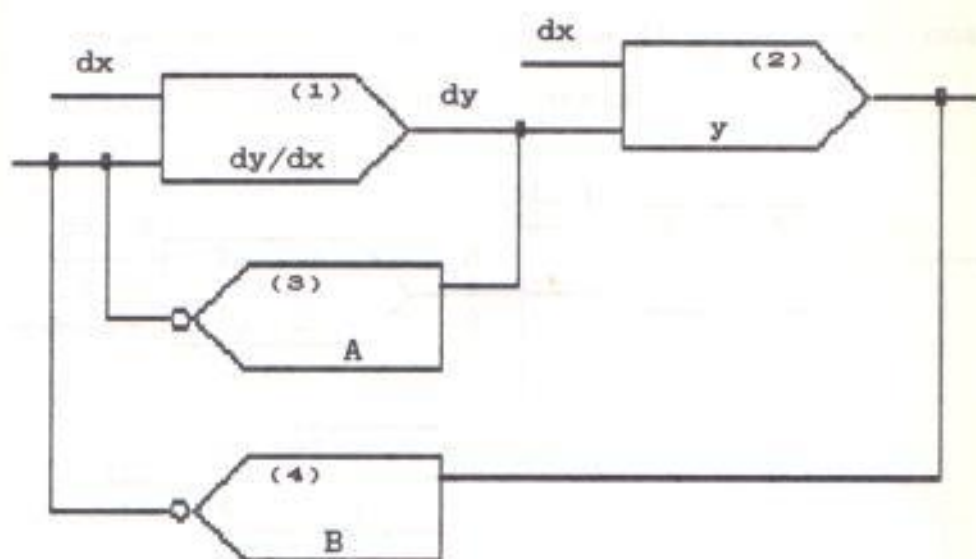


FIG. 1.12 DIAGRAMA DE LA SIMULACION DE LA ECUACION DIFERENCIAL DE SEGUNDO ORDEN.

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:

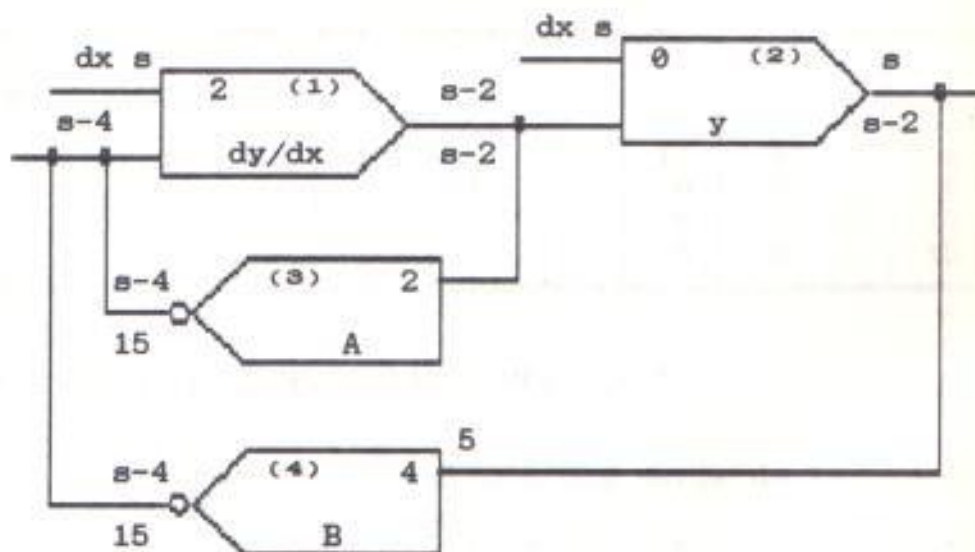


FIG. 1.13 ESCALAMIENTO DE LA ECUACION DIFERENCIAL DE SEGUNDO ORDEN.

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

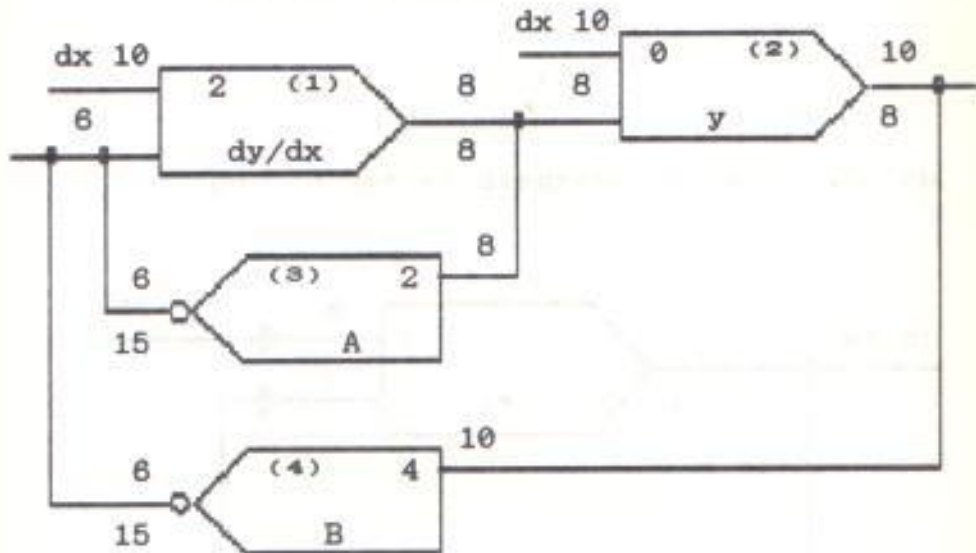


FIG. 1.14 DIAGRAMA COMPLETO DE LA SIMULACION DE LA ECUACION DIFERENCIAL DE SEGUNDO ORDEN.

La tabla del informe de las conexiones 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 :

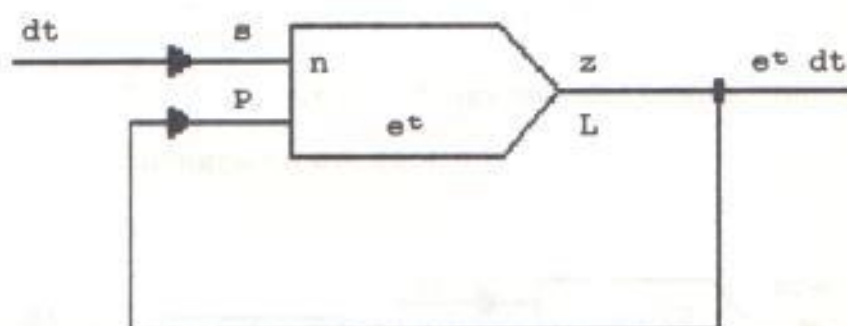


FIG. 1.15 LAZO EXPONENCIAL

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 definan 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:

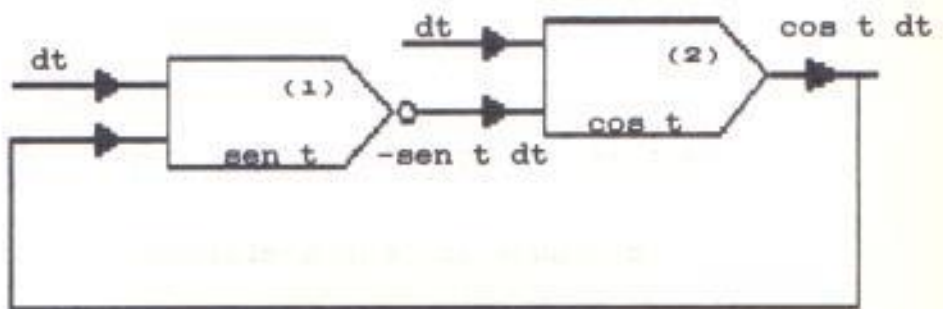


FIG. 1.16 LAZO SEÑO - COSENO

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).

$$z_1 + n_1 = z_2 + n_2$$

$$p_1 + n_1 = p_2 + n_2$$

y como $z_1 = p_2$ y $z_2 = p_1$

concluimos que $n_1 = n_2 = n$

por lo que si $z_1 = s - n$

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 :

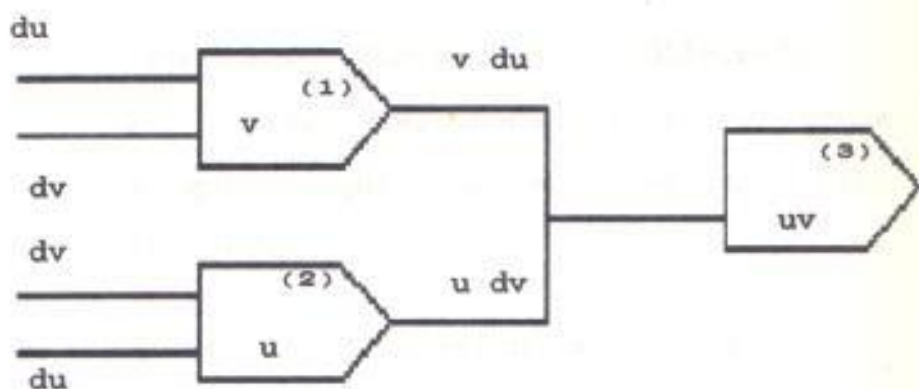


FIG. 1.17 EL PRODUCTO.

Observamos que :

$$p_3 = z_1 = z_2$$

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

$$z_1 = s_1 - L_1 + p_1$$

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

Si escogemos que $L_1 = L_2 = L$

entonces , $s_1 + p_1 = s_2 + p_2$

1.4.4.d Generación de x^2

La ecuación diferencial para este diagrama es:

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

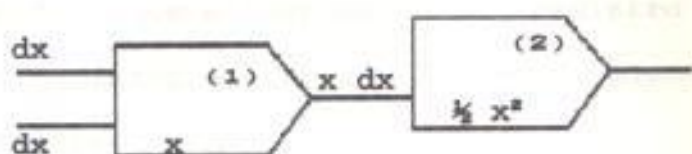


FIG. 1.18 GENERACION DE POLINOMIOS.

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

Si $2 \cdot 2 = 1 = 2 - p^2$ entonces $p^2 = z_1 - 1$

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:

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

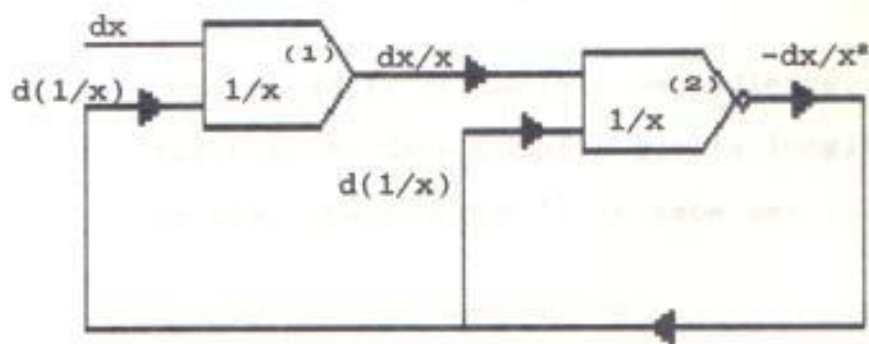


FIG. 1.19 LA FUNCION RECIPROCA.

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 $x^{1/2}$

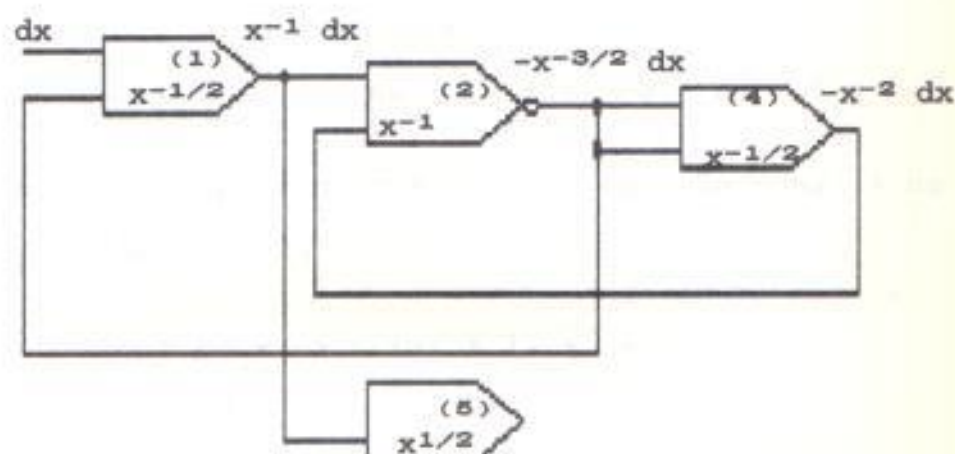


FIG. 1.20 LA RAIZ CUADRADA.

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 las 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 desfase 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 conexiones 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 rápidamente.

Error por truncación puede estar definido matemáticamente, después 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^{-16} unidades; un límite aproximado puede ser establecido para el error:

$$\begin{aligned} \text{total de error} &\approx 2/3 \text{ de 1 incremento} \\ &\approx 2/3 * 2^{-16} * 100\% \\ &\approx 0.002\% \end{aligned}$$

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^o

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:

$$dz_1 = 0 \quad R_{i+1} = R_i + Y_1$$

2) sobrecarga positiva, en este caso:

$$dz_1 = 1 \quad R_{i+1} = R_i + Y_1 - 2^n$$

3) sobrecarga negativa, en este caso:

$$dz_1 = -1 \quad R_{i+1} = R_i + Y_1 + 2^n$$

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

$$dz_1 = 1/2^n * (Y_1 + R_i - R_{i+1})$$

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 :

$$Z_N - Z_0 = \frac{1}{2^n \Delta t} \int_{t_0}^{t_N} y \cdot \Delta t - \frac{1}{2 \cdot 2^n} (y_N - y_0) + \frac{R_N}{2^n} + \frac{1}{2^n} (\alpha_0 + \alpha_1 + \dots + \alpha_N)$$

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 más 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.

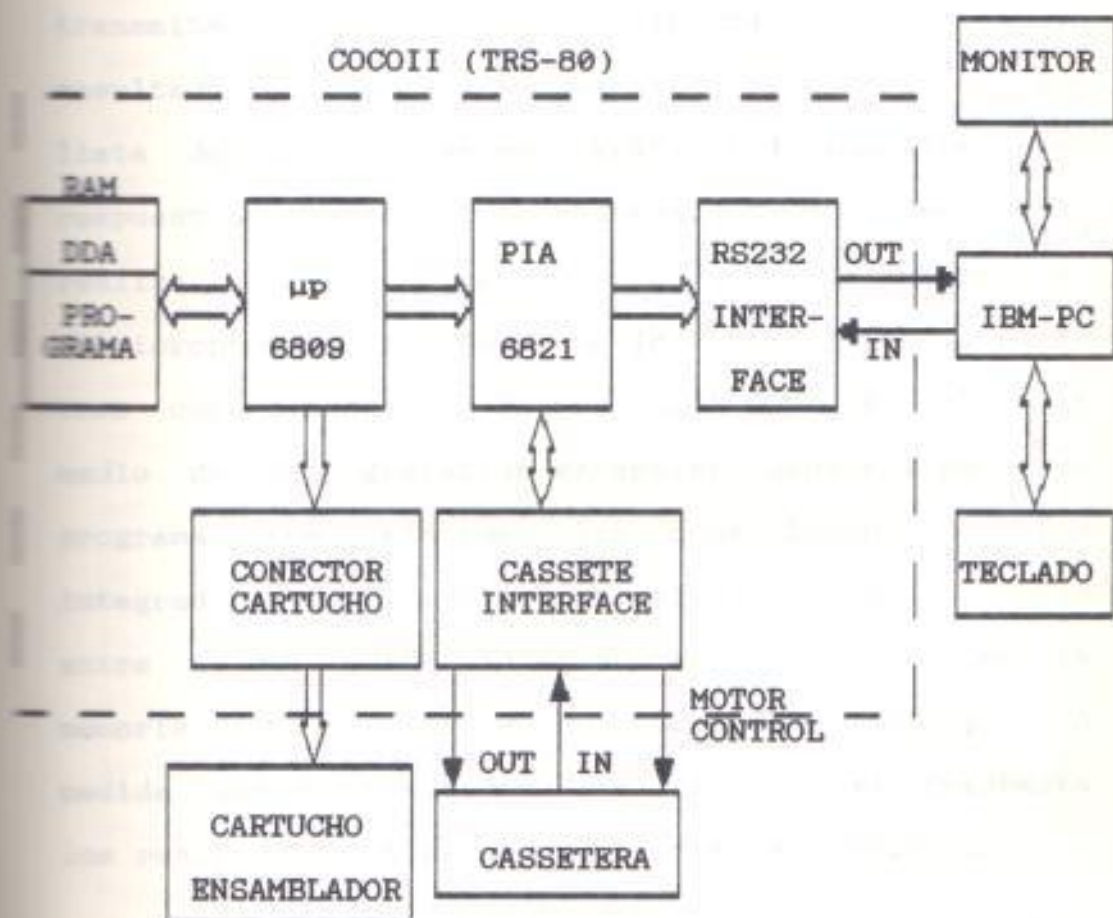
REFERENCIAS

1. PARASURAMAN B. Digital Incremental Computation using Automatic Programming Techniques. 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.

Director DDA
 Director DDA
 Director DDA
 Director DDA

Director Tablas de Operaciones
 Director de Operaciones

Subdirector de Operaciones
 Subdirector de Operaciones
 Subdirector de Operaciones
 Subdirector de Operaciones

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

SEUDOCODIGO

Asunto: DDA

Autor: RPP

Página: 1

Clasificación: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

```

IN
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

Aspcto: 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;
2: BEGIN
Colocar extension '.DDS' a NombArch
Mostrar ('1: Por Pantalla');
Mostrar ('2: Por Impresora');

```


PDD-002A

SEUDOCODIGO

Proyecto: 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;

```

PDD-002A

SEUDOCODIGO

Objeto: 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 lineas

ELSE

Mostrar parte baja de DDASCR02 para 6 lineas;

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

END;

3: BEGIN

Mostrar Pantalla DDASCR03;

IF Nint <= 6 THEN

Mostrar parte baja DDASCR03 para Nint lineas

ELSE

Mostrar parte baja de DDASCR03 para 6 lineas;

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;

HASTA Tecla = 4;

PDD-002A

SEUDOCODIGO

Objeto: 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;
  IniciarCadenas (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

Objeto: 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

```

EscribirCentrado (LST, 'DATOS DE ENTRADA');
WriteIn(LST);
EscribirCentrado (LST, 'Archivo: ', NombArch);
WriteIn(LST);
WriteIn (LST, 'Numero de integradores : ', Nite);
WriteIn (LST, 'S_bit de precision : ', S_bit);
WriteIn (LST, 'Paso de impresion : ', PasoImp);
WriteIn (LST, 'Valor Inicial : ', ValIni);
WriteIn (LST, 'Valor Final : ', ValFin);
WriteIn(LST);
WriteIn (LST, 'Nombre del Integrador dy1 dy2 dy3 dy4 dy5 dx',
        ' n p CondIni');
NumLinea = 14;
FOR i = 1 TO Nint DO
  BEGIN
    WriteIn (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

Aspcto: DDA

Autor: RPP

Página: 8

Identificación: Graficar

Versión: 1.0

Fecha: 10-OCT-90

```

Pasar disco;
IF Arch no existe THEN EXIT;
Habilitar pantalla para alta resolucio;n;
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;

```

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 (str);
          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 str;
                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;
            Pasarse 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;
```


PDD-002A

SKUDOCODIGO

Objeto: DDA

Autor: RPP

Página: 11

Identificación: ListarImp

Versión: 1.0

Fecha: 10-OCT-90

```

IF NombArch no existe THEN EXIT;
Leer archivo (NombArch);
Habilitar formato pantalla DDDLST02;
Nume := 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 (Nume 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;
            END;
          Pasarse a la siguiente Salida;
        UNTIL Salidas mostradas <= SalidasAMostrar;
        WriteLn (LST);
        NumLinea = NumLinea + 1;
      END;
    END;
  END;

```

PDD-002A

SEUDOCODIGO

~~Objeto:~~ DDA

Autor: RPP

Página: 12

~~Modificación:~~ 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

~~Identificación:~~ 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 = chr($FF) OR k = 2 THEN
```

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

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


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}

 reciba2 (ch) {recibo caracter}

 MISTA QUE ch = 'T';

 transm ('R'); {transmito caracter}

REPITO

 reciba (ch); {recibo caracteres}

 almaceno ch en TABLA;

 lejo palabra de paridad del grupo de datos;

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

 lejo (palabra de paridad);

 lejo palabra de paridad;

TRANSFORMACION DE CARACTERES ASCII A BINARIO *)

SIEMPREAS no sea EOF

REPITO

 lejo caracteres de TABLA;

 almaceno en DATO;

 MISTA 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;

 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]; {conexiones dy de integradores}

PDD-002A

SKUDOCODIGO

Objeto: DDA

Autor: RPP

Página: 2

Clasificación: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

```

    incremento j;
    BASTA QUE j = 5;
    almaceno en XS[i];
    incremento i;
    BASTA 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;
    BASTA 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;
    BASTA 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;
        BASTA QUE j = 5 o SS[i,j] = 0;
    incremento i;
    BASTA QUE i = M; {ultimo integrador}

```

SEUDOCODIGO

PDD-002A

Respecto: DDA

Autor: RPP

Página: 3

Identificación: 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}

DEBIDO

incremento print;

SI print = 0 ENTONCES

 conver; {procedimiento de conversión de binario a caracter}

 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}

DEBIDO

 inicializo j:=0; {contador de columnas}

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

 DEBIDO

 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 conecciones 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}

* NECESIDAD DE LOS INTEGRADORES DIGITALES EN LA MAQUINA DDA *

inicializo i:=0; {contador de integradores}

DEBIDO

 actualizamos los parametros de la subrutina que procesa cada integrador)

SEUDOCODIGO

Título: DDA

Autor: RPP

Página: 4

Categorización: PRINCIPAL

Versión: 1.0

Fecha: 10-OCT-90

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

matrz(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

Aspecto: 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}

INICIO

chequeo bit 0 de la palabra FF22;

MIENTA QUE bit 0 = 0;

muestreo la señal de entrada luego de 833.3 μ s*B}

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

cogr(B); {subrutina de retardo}

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

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

INICIO

muestreo la señal de entrada luego de 833.3 μ s*B}

B:=9;

cogr(B);

SI bit recibido = 1 ENTONCES incremento part;

decremento nbit;

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

muestreo la señal de entrada luego de 833.3 μ s*B}

B:=4;

cogr(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: ESPR

Versión: 1.0 Fecha: 10-OCT-90

PROCEDIMIENTO ESPR 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}

{mascara la direccion FF01H con 00110111B ;

{espero por llegada de interrupcion}

CALL EPH; {llamada de interrupcion}

decremento B;

MIRA 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}

{mascara la direccion FF01H con 00110110B;

{enciende bit; {bit 0 de FF22H}

{enciende bit en RSIN;

FIN {subrutina llama}

PDD-002A

SEUDOCODIGO

Asunto: 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 $833.3\mu s * B$ }

esper bit 1 de FF20H; {bit de inicio}

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

B:= 9;

 transmito bit; {roto RSOUT para transmitir cada bit a traves
 del bit 1 de FF20H}

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

decremento nbit;

SI nbit = 0; {fin de transmision de caracter}

transmito bit de paridad;

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

esper bit 1 de FF20H; {bit de parada}

esper(B);

decremento ndbuf; {contador de datos en buffer}

FIN {procedimiento transm}

PDD-002A

SEUDOCODIGO

Aspecto: DDA

Autor: RPP

Página: 9

Especificación: SALIDA

Versión: 1.0

Fecha: 10-OCT-90

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

PROCEDIMIENTO salida;

inicializo i:=-5; {contador de retardo}

trans ('T'); {handshake}

SI

decremento i;

RECIBO RECIBO2(ch); {recibo caracter}

SI (ch = 'R' O ch = 'R');

SI NO 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

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

decremento i;

SI (i = 0) ENTONCES {fin de transmision de un dato}

decremento j;

SI (j = 0) ENTONCES {fin de transmision de todos los datos}

SI (ctr + PRNT) > FIN ENTONCES

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

SI

transm(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}

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

delay; {subrutina de retardo}

encero ndbuf; {buffer vacio}

FIN-SI;

FIN-SI;

FIN {procedimiento salida}

PDD-002A

SEUDOCODIGO

Autor: DDA

Autor: RPP Página:10

Modificación: MODTRZ

Versión: 1.0 Fecha: 10-OCT-90

PROCEDIMIENTO MODTRZ PROCESO DE INTEGRACION DE UN INTEGRADOR
(CONTINUA *)

```

PROCEDIMIENTO modtrz(aa,bb,ccy,dd,e,f,g);
  max ← 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}
  dz ← dd + sgn(aa)* (ccy + bb/2); {integracion trapezoidal}
  max ← 2 * l; {maximo valor del registro R}
  SI dz > max ENTONCES
    dz ← -1; {seteo dz}
    resto en dd valor max; {actualizo en R el residuo}
  SINO SI dz = max ENTONCES
    dz ← 0; {encero dz}
  SINO
    dz ← -1; {seteo dz}
  FIN-SI;
FIN {procedimiento modtrz}

```

PDD-002A

SEUDOCODIGO

Título: DDA

Autor: RPP Página: 11

Identificación: CONVER

Versión: 1.0 Fecha: 10-OCT-90

PROCEDIMIENTO CONVER PARA CONVERTIR DATOS BINARIOS DE LOS
 "BITSTROS 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}

REPITO

SI CY[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

La IBM es una de las compañías más importantes del mundo en el campo de la informática. Su historia comienza en 1911 con la fundación de la International Business Machines Corporation (IBM).

La IBM es una de las compañías más importantes del mundo en el campo de la informática.

La IBM es una de las compañías más importantes del mundo en el campo de la informática.

La IBM es una de las compañías más importantes del mundo en el campo de la informática.

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.

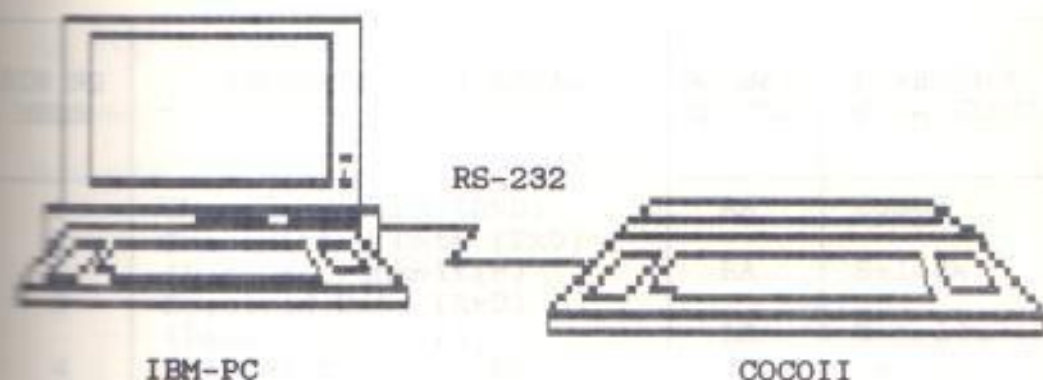


FIGURA 3.1 CONECCION ENTRE CPU Y TRS-80

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.

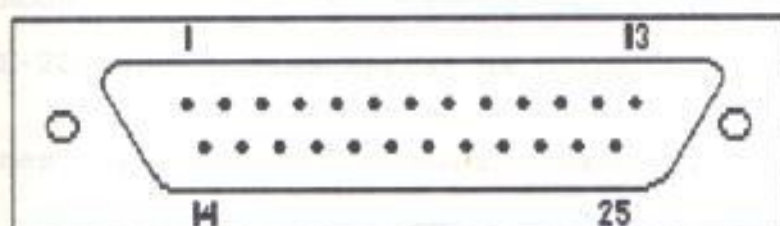


FIG. 3.2 CONECTOR DB25

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 ^o 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:

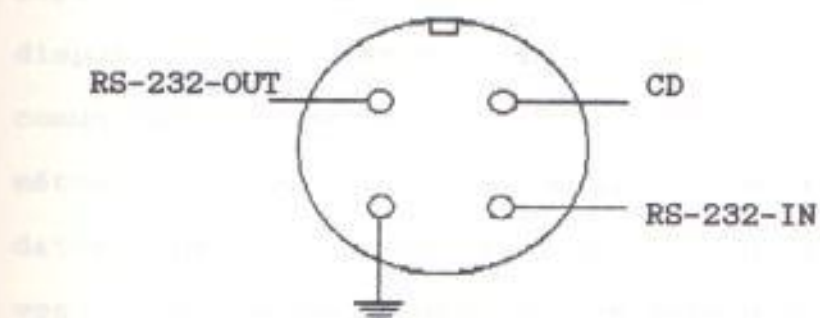


FIG. 3.3 CONECTOR DIN.

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

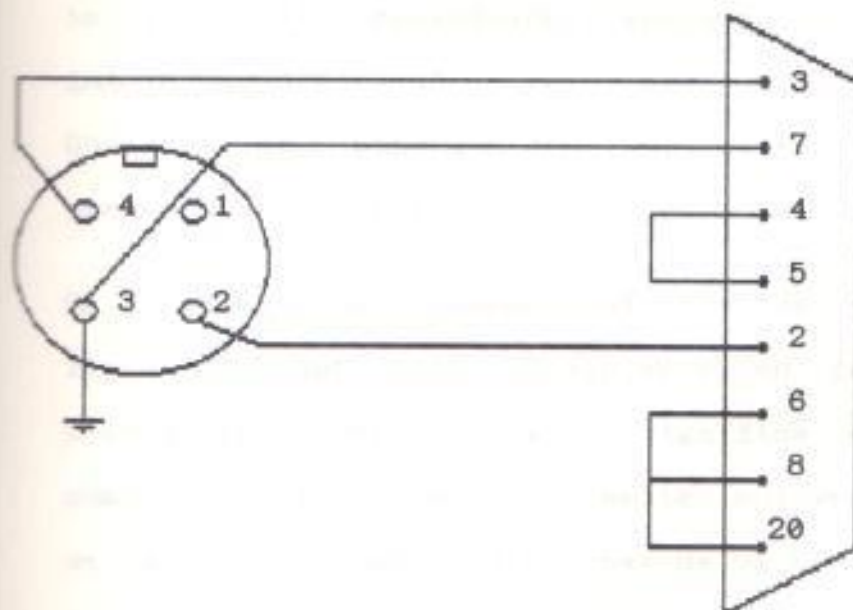


FIG. 3.4 CABLEADO

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, evitando así 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 receipta 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 receiptado 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 línea

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 línea 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 línea; éste determina la longitud del carácter, 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	<table> <tr> <td>0</td> <td>Deshabilita IRQ</td> </tr> <tr> <td>1</td> <td>Habilita IRQ</td> </tr> </table>	0	Deshabilita IRQ	1	Habilita IRQ
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:

- Guardamos el caracter en la dirección de memoria RSOUT.
- A través del bit 1 del registro FF20 enviamos bit de inicio.
- Habilitamos la interrupción IRQ por el bit 0 del registro FF01.
- 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 receptor 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 receptor 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 accesan 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 accesamos 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.

Habiamos 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 teniamos

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 conexiones
- 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 conexiones 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.

TECNICAS 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:

$$\begin{aligned} |H(j\omega)| &= A & 0 < \omega < \omega_c \\ |H(j\omega)| &= 0 & \omega > \omega_c \end{aligned}$$



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

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

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

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 $\omega_0 = 1$

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

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

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

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

$$e^{j2n} = -1 = e^{j(2k-1)\pi} \quad ; \quad \text{para } k=0, 1, 2, \dots, 2n-1$$

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

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

asi,

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

la parte real

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

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

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:

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

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

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

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

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

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

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

$$\begin{aligned} C_n(-js) &= \cos n(u + jv) \\ &= \cos nu \cosh nv - j \sen nu \sinh nv = \pm j/\epsilon \end{aligned}$$

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

$$\text{como} \quad s_k = \sigma_k + j\omega_k$$

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

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

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:

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

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

$$n \approx 2.61$$

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:

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

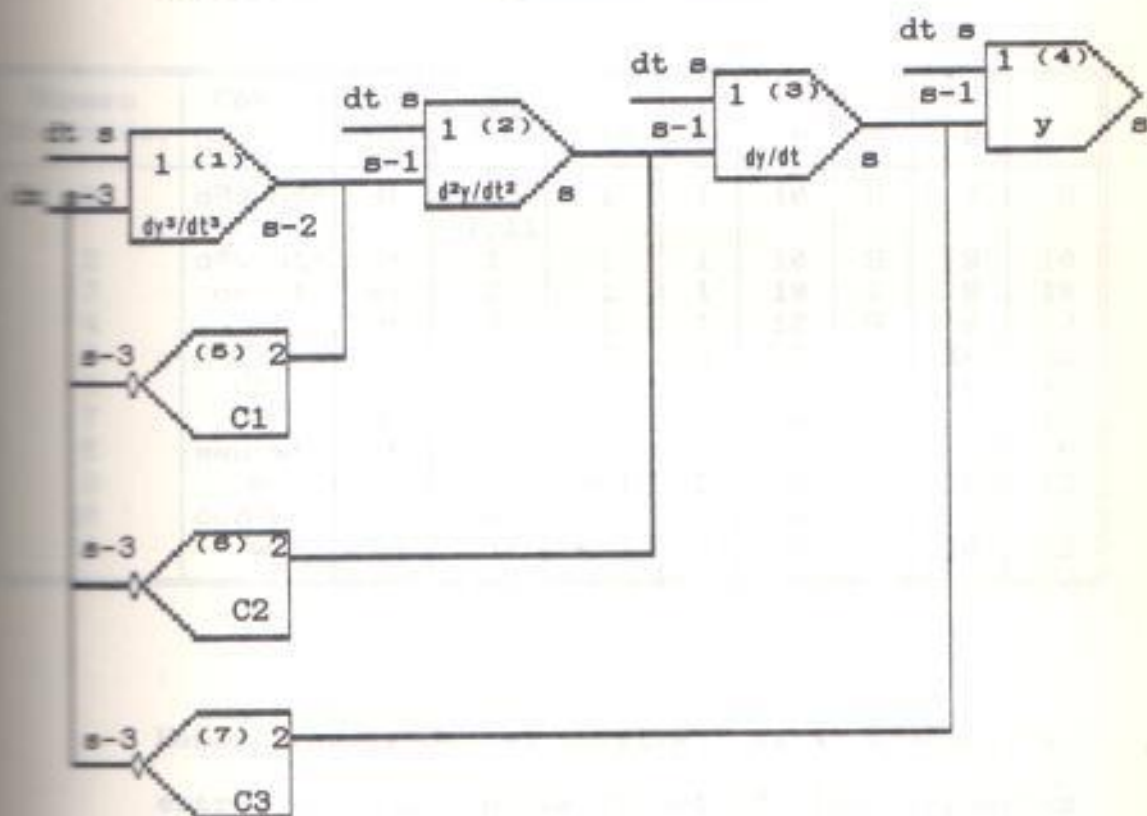
$$\frac{d^3y}{dt^3} + 2 \frac{d^2y}{dt^2} + 2 \frac{dy}{dt} + y = x$$

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:



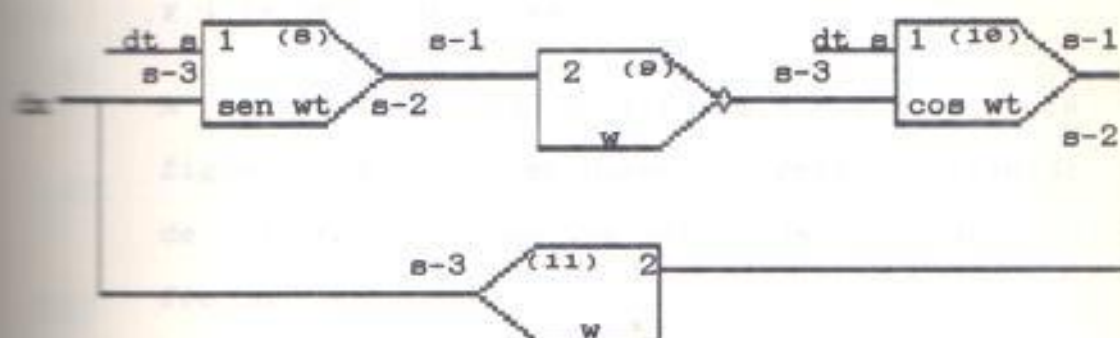


FIG. 4.3 DIAGRAMA DE SIMULACION DE UN FILTRO BUTTERWORTH TERCER ORDEN

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:

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

si $u = \omega^3$ y $v = 1 + u^2$

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

FIG.

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

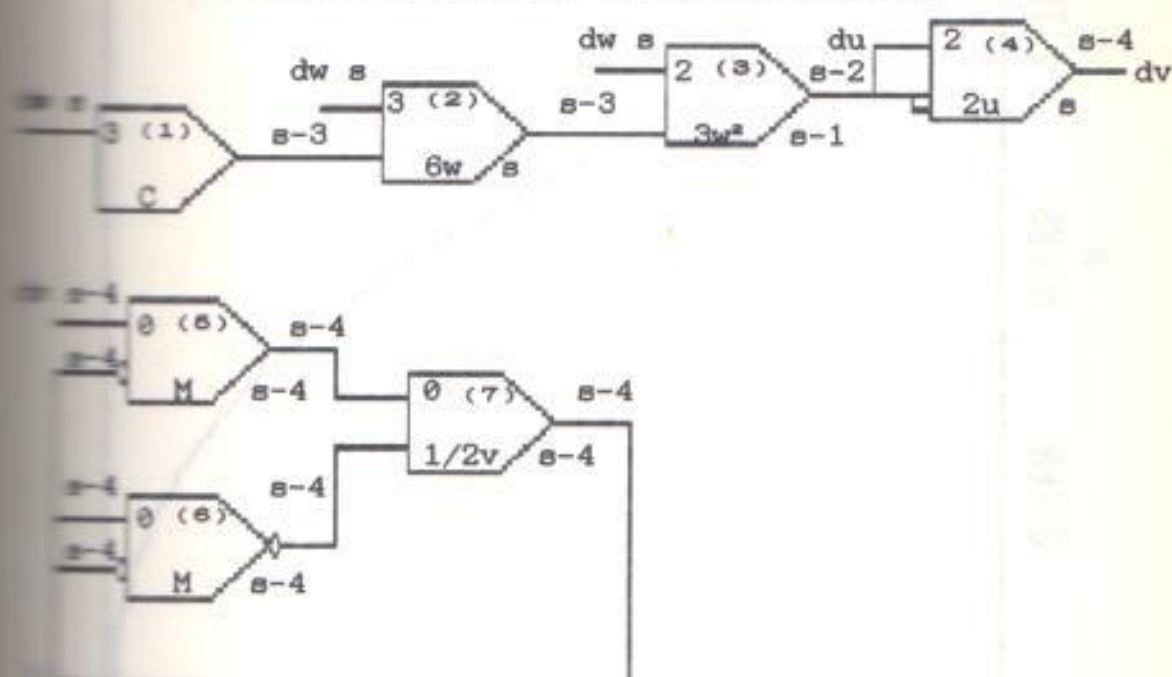


FIG. 4.6 DIAGRAMA DE LA RESPUESTA DE MAGNITUD DEL FILTRO DE BUTTERWORTH DE TERCER ORDEN.

La respuesta de magnitud la obtenemos del integrador 5.

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

Número 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

3.2 EJEMPLO 2 *

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

- El rizado no exceda el 7.8% de su valor máximo 1.
- 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:

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

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

y

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

$$\sigma_1 = - \operatorname{sen} \frac{\pi}{8} \operatorname{senh} (0.4) = - 0.157$$

$$\sigma_2 = - \operatorname{sen} \frac{3\pi}{8} \operatorname{senh} (0.4) = - 0.379$$

$$\sigma_3 = - \operatorname{sen} \frac{5\pi}{8} \operatorname{senh} (0.4) = - 0.379$$

$$\sigma_4 = - \operatorname{sen} \frac{7\pi}{8} \operatorname{senh} (0.4) = - 0.157$$

y

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

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

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

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

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

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:

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

entonces:

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

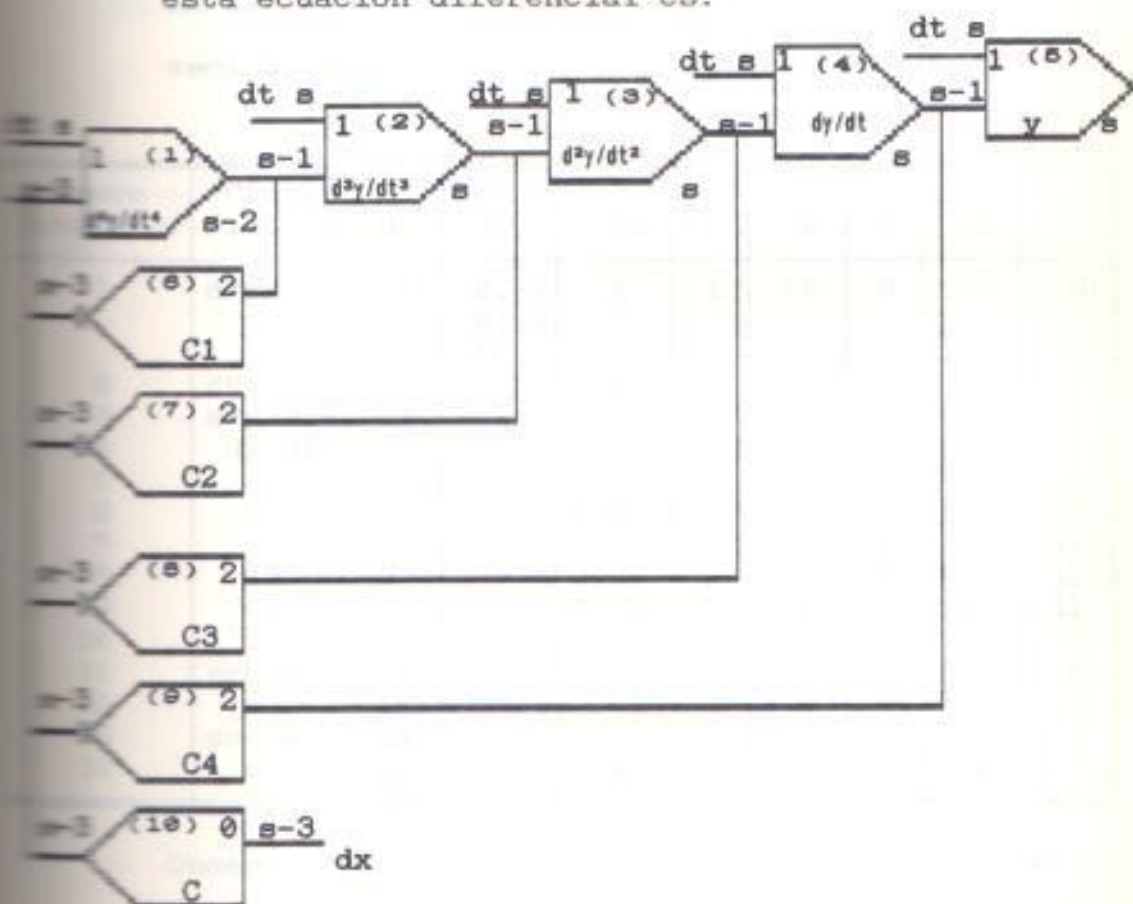
Pasando a la ecuación diferencial en tiempo tenemos:

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

diferenciando y despejando la mas alta derivada:

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

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



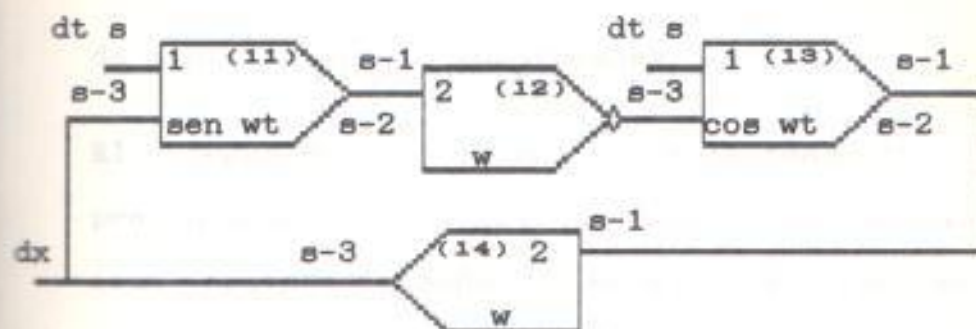


FIG. 4.8 DIAGRAMA DE SIMULACION DEL FILTRO CHEBYSHEV DE CUARTO ORDEN.

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:

Numero Integrador	Contenido Y	dx	dy	Y_m	n	s	z	p	L
1	d^4y/dt^4	dt	-6,-7 -8,-9 10	1	1	10	9	7	8
2	d^3y/dt^3	dt	1	1	1	10	9	9	10
3	d^2y/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:

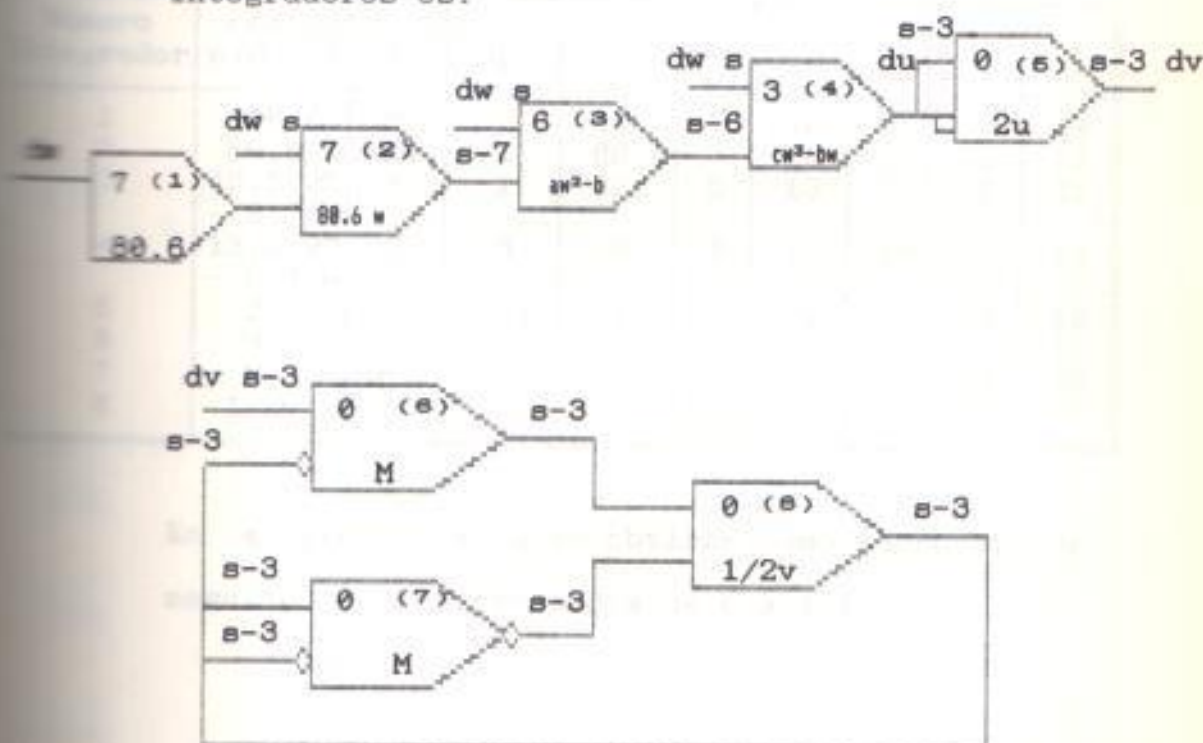


FIG. 4.9 DIAGRAMA DE SIMULACION DE LA RESPUESTA DE MAGNITUD DEL FILTRO CHEBYSHEV DE CUARTO ORDEN

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

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

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

$$\text{Si } u = 3.36 w^4 - 3.36w^2 + 0.42 \text{ y } v = 1 + u^2$$

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.



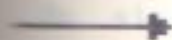
FIG. 4.12. RESPUESTA DE MAGNITUD EN LA FRECUENCIA DE 0 A 1.1.

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

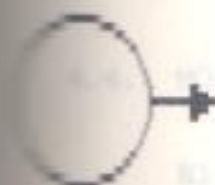
Esta información la obtuve de la obra "Computer
Minimodels and Simulation Exercises" de Howard T.
Odum. El basa sus ideas sobre la relación entre
economía y ecología. Odum sostiene "que el
desarrollo económico no debe conducirnos al deterioro
de la naturaleza y que una economía sana se sustenta
siempre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

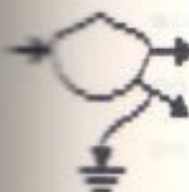


FUENTE DE ENERGIA.- Energía la que es acompañada por
cada uno de los recursos usados por los
ecosistemas tales como el sol, vientos,
lluvias, etc.



ALMACENAJE.- Un lugar donde la energía es almacenada.

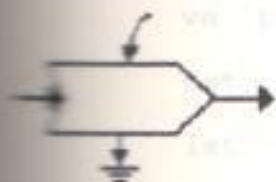
Ejemplos son los recursos tales como
biomasa forestales, materia orgánica,
suelos, etc.



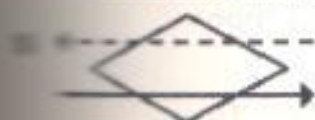
DESHECHOS.- Energía que es disipada.



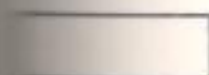
INTERACCION.- Procesos los cuales combinan diferentes tipos de flujos de energía o flujo de materia.



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



CAJA.- Símbolo misceláneo 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 $I_0 = 100$ y la proporción de la energía que se utiliza es $K_0 = 0.38$.

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:

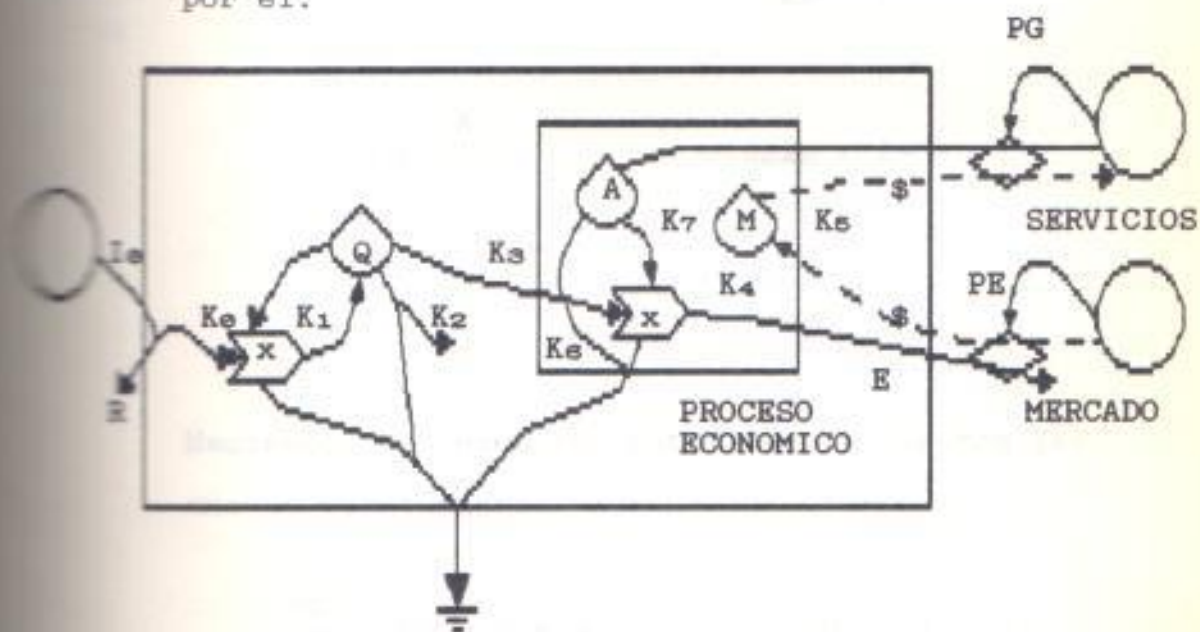


FIG. 4.13 DIAGRAMA DE SIMULACION DEL MODELO ECONUSE

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:

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

$$\frac{dm}{dt} = C_7 * a * q - C_6 * m$$

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

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

$$e = C_7 * a * q$$

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

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

$$\left[\frac{dm}{dt} \right] = C_7 a dq + C_7 q da - C_6 dm$$

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

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

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

$$du = C_9 dq$$

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

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

FIG. 1. Diagrama de conexiones de los integradores para la simulación de la dinámica de un sistema de control de un motor de corriente continua.

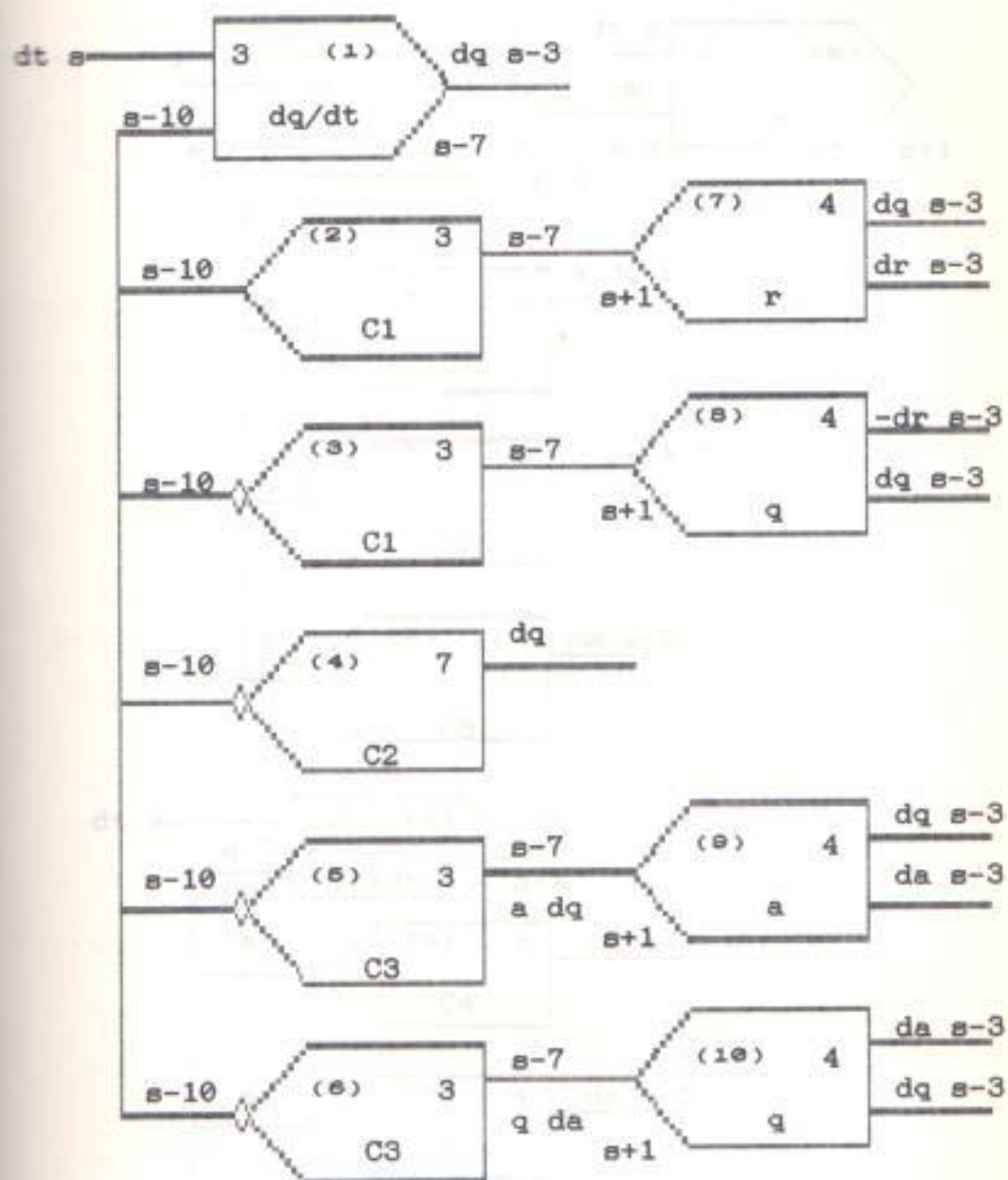


FIG. 4.14 DIAGRAMA DE CONECCION DE INTEGRADORES DIGITALES DE LA SIMULACION DEL MODELO DE ECONUSE (PRIMERA PARTE)

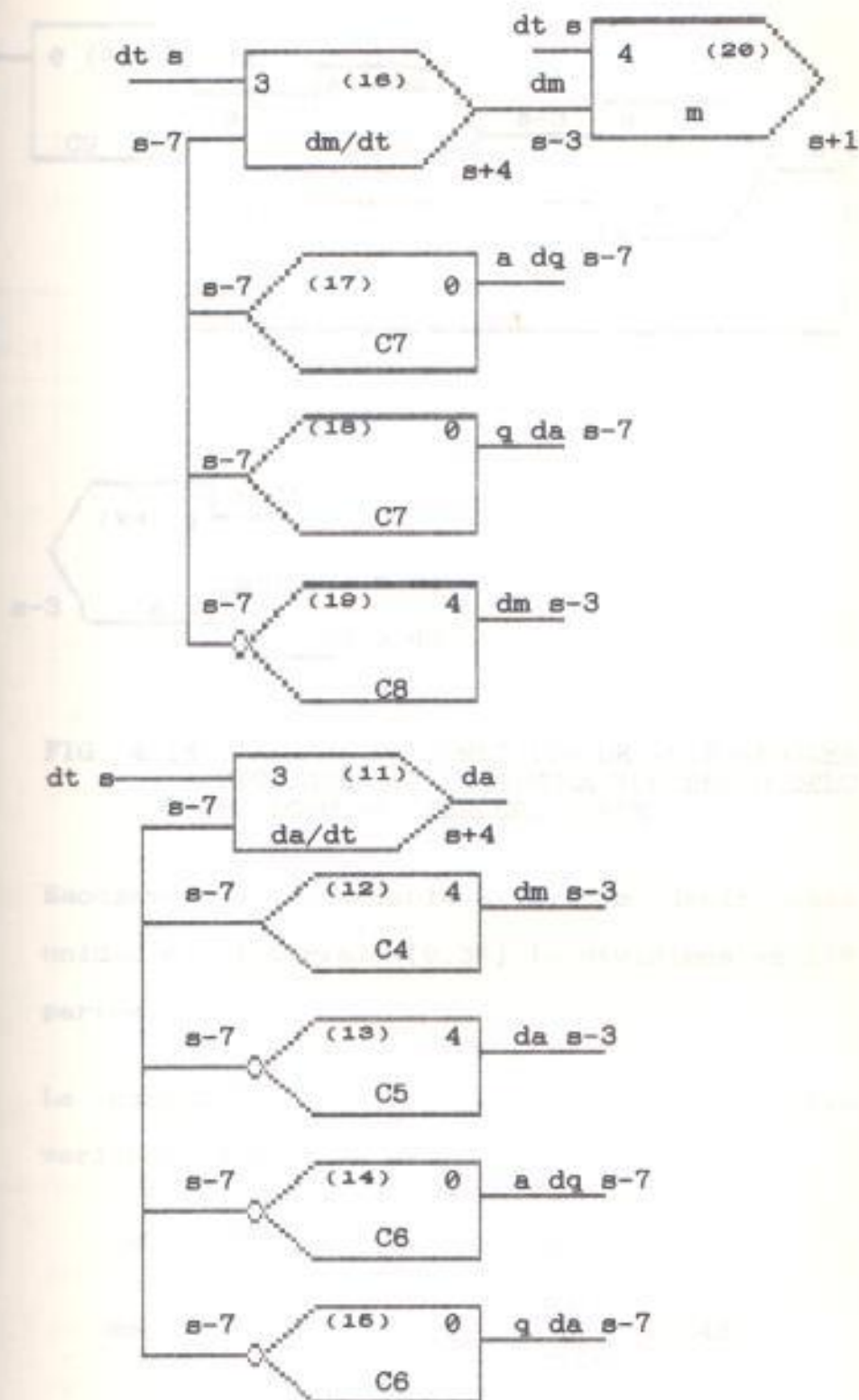


FIG. 4.14 DIAGRAMA DE CONECCION DE INTEGRADORES DIGITALES DE LA SIMULACION DEL MODELO DE ECOMUSE (SEGUNDA PARTE)

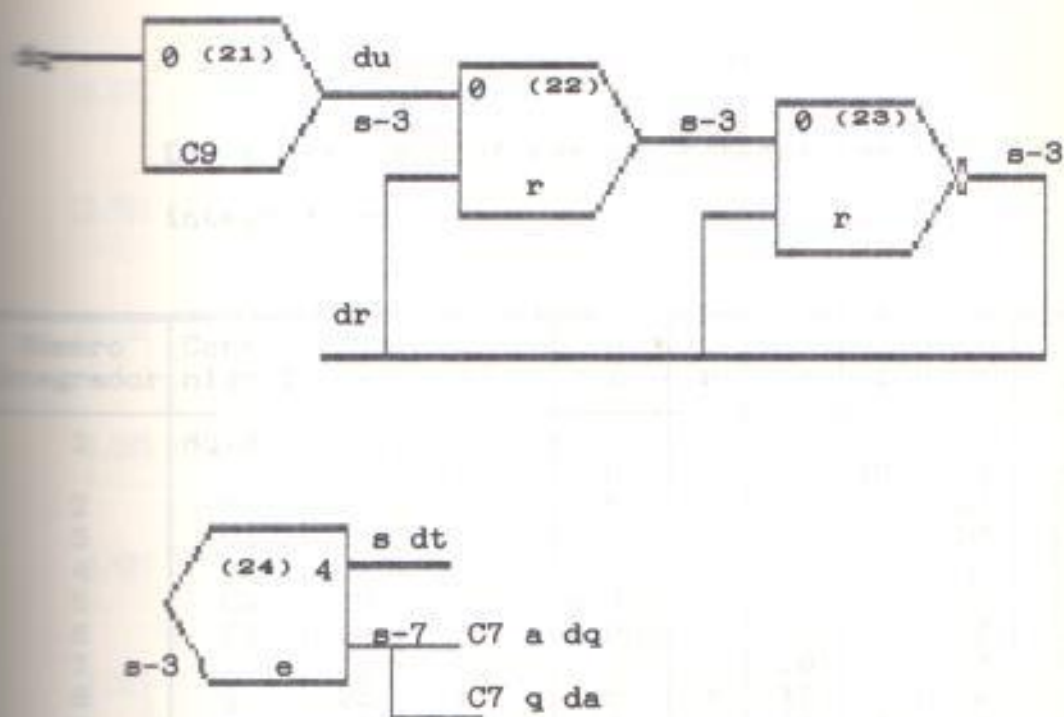


FIG. 4.14 DIAGRAMA DE CONECCION DE INTEGRADORES DIGITALES DE LA SIMULACION DEL MODELO DE ECONUSE (TERCERA PARTE)

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:

Número 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

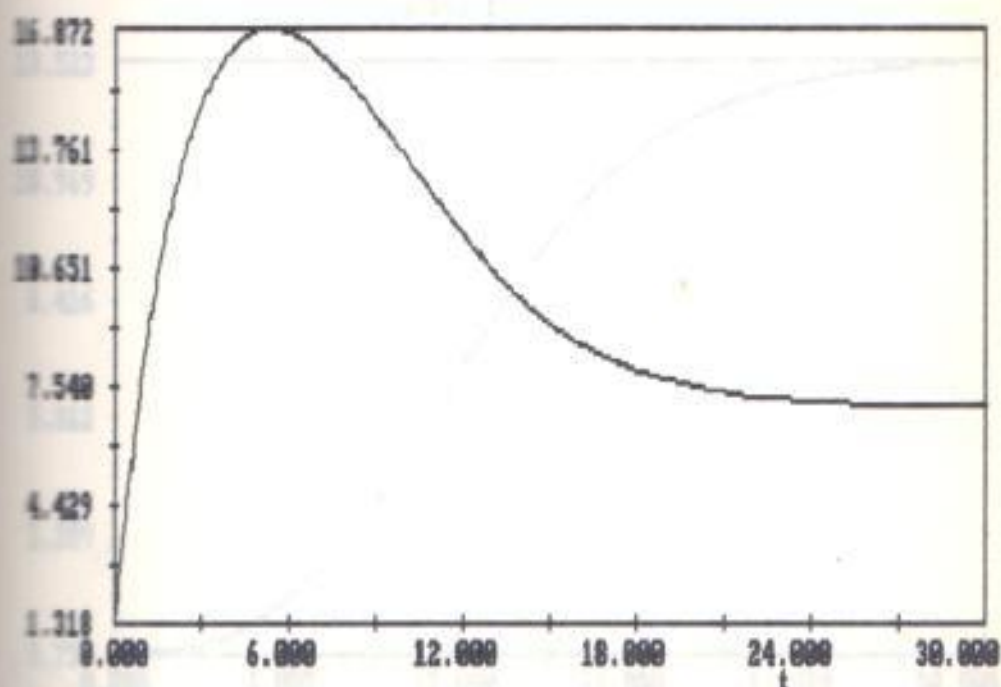


FIG. 4.15 PRODUCTO (Q) BAJO CONDICIONES OPTIMAS

n Vs. t

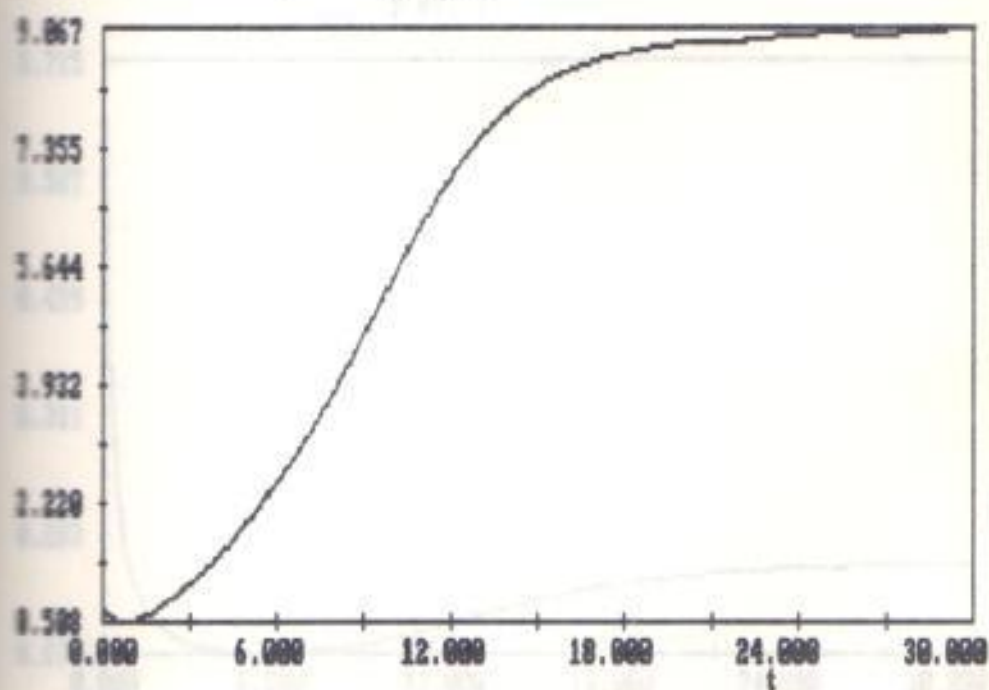


FIG. 4.16 DINERO (M) BAJO CONDICIONES OPTIMAS

a Vs. t

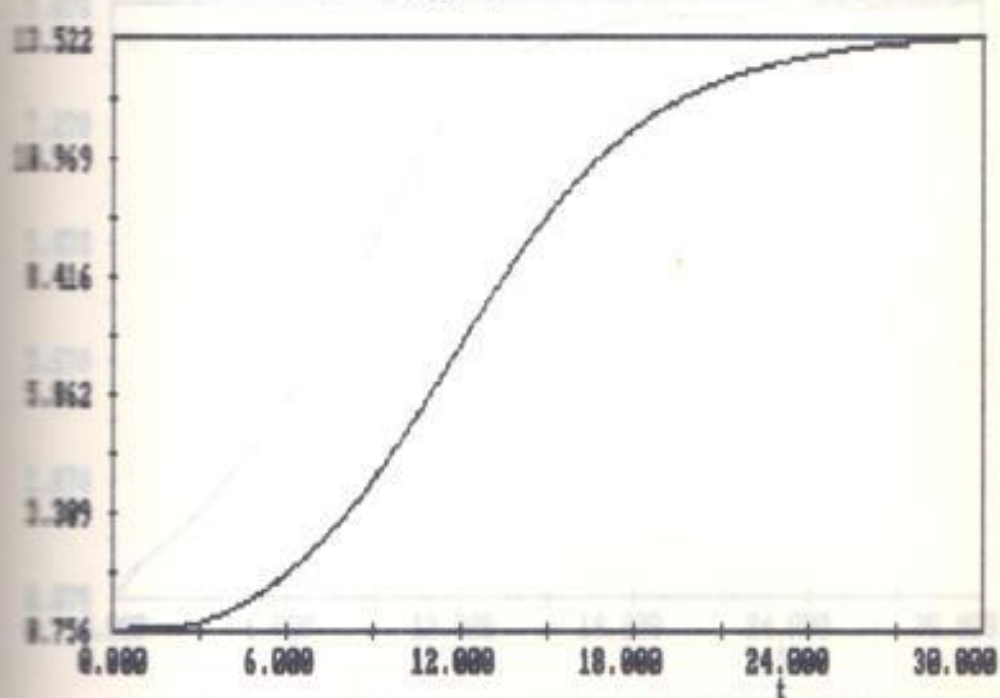


FIG. 4.17 ACTIVOS (A) BAJO CONDICIONES OPTIMAS

r Vs. t

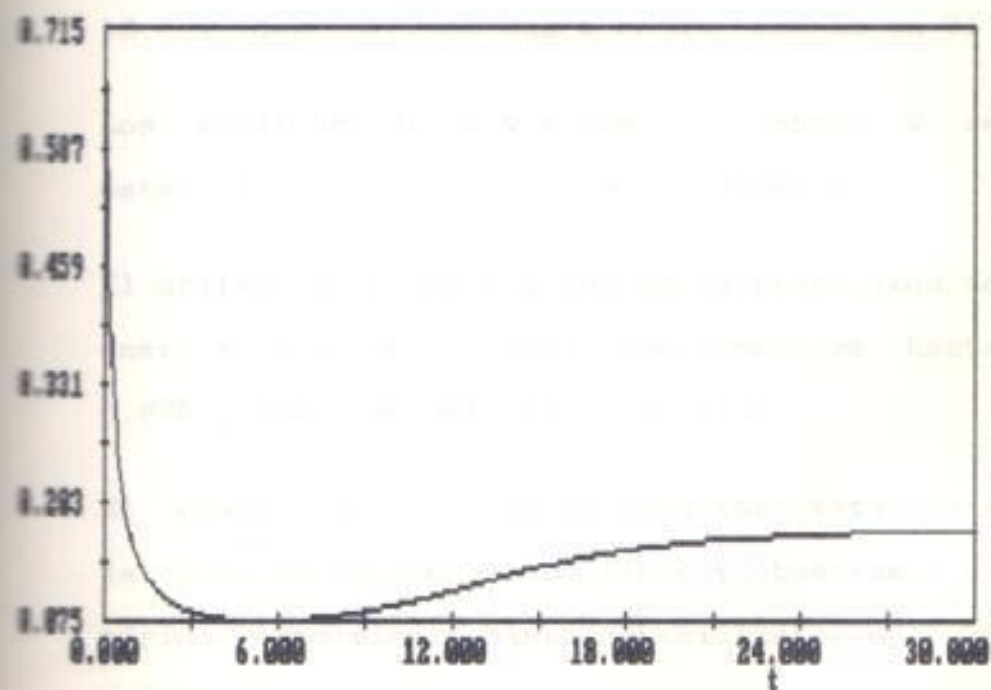


FIG. 4.18 PROPORCION DE ENERGIA UTILIZADA R

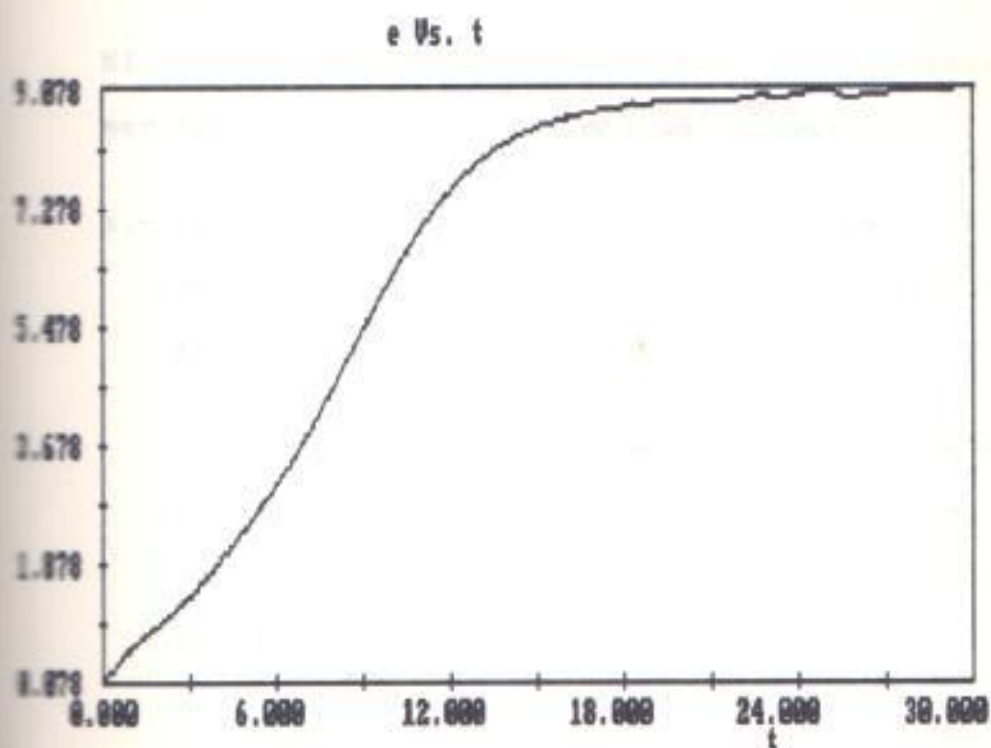


FIG. 4.19 CANTIDAD DE PRODUCTO LISTO PARA LA VENTA

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

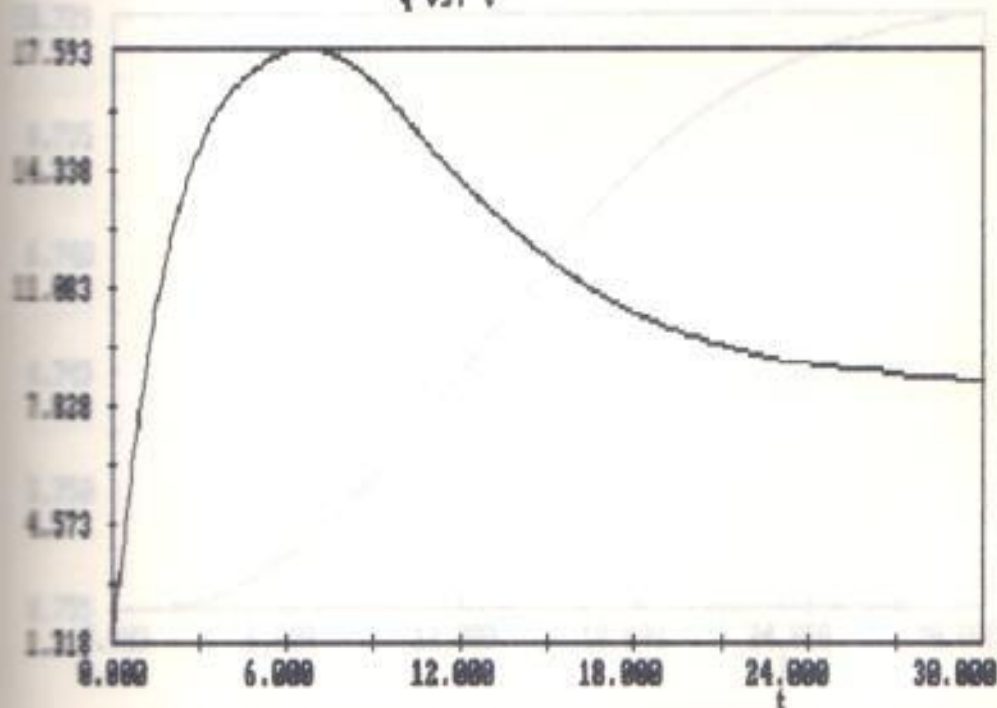


FIG. 4.20 PRODUCTO (Q) CON INCREMENTO DE PG EN 10%

n Vs. t

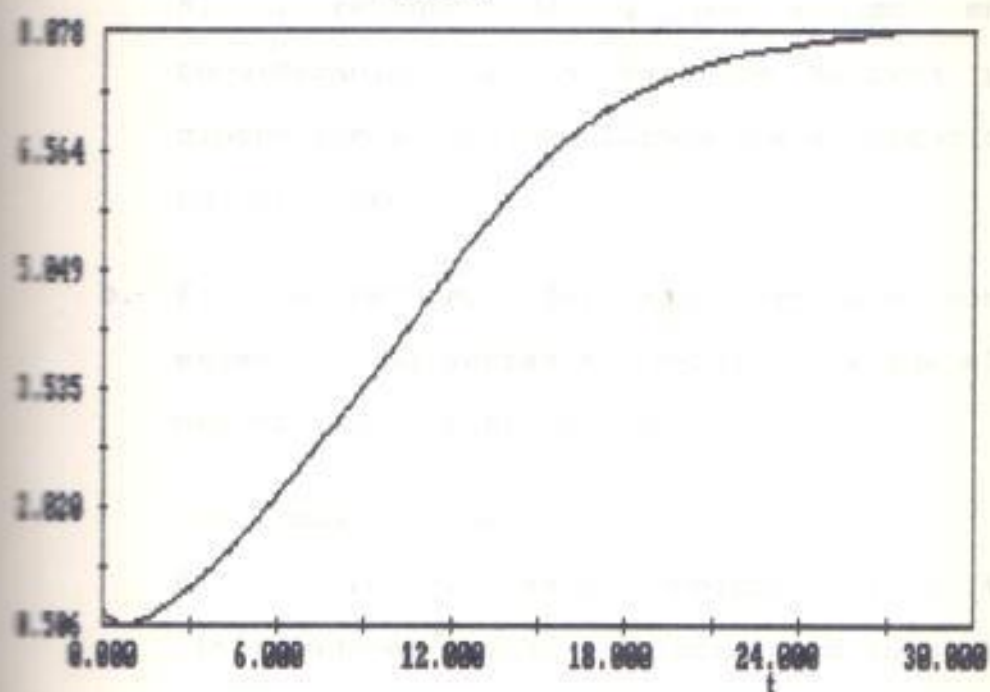


FIG. 4.21 DINERO (M) CON INCREMENTO DE PG EN 10%

a Vs. t

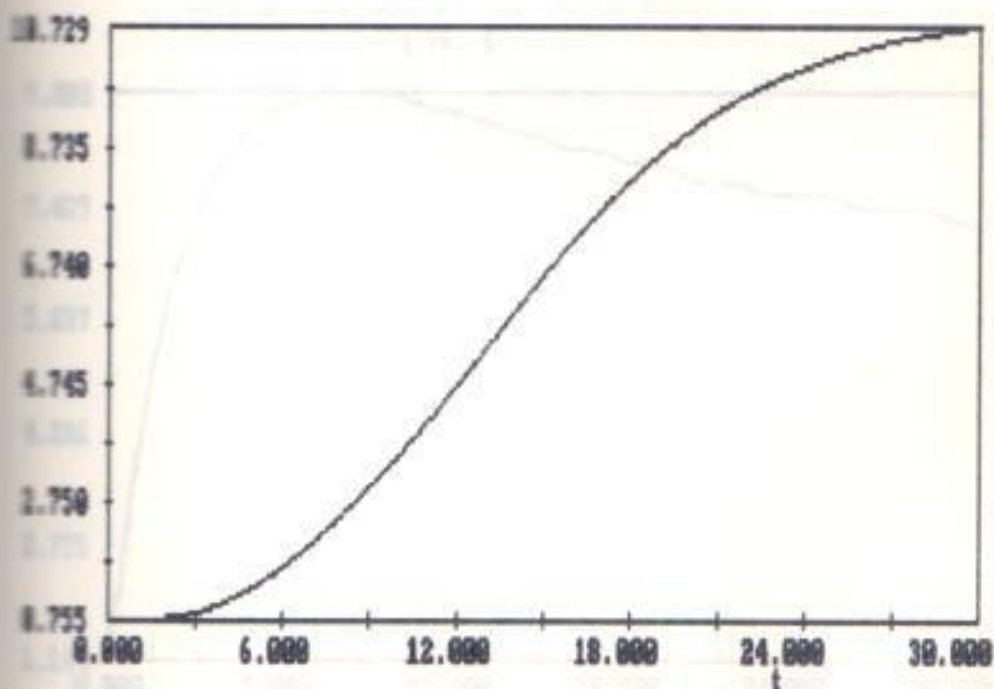


FIG. 4.22 ACTIVOS (A) CON INCREMENTO DE PG EN 10%

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

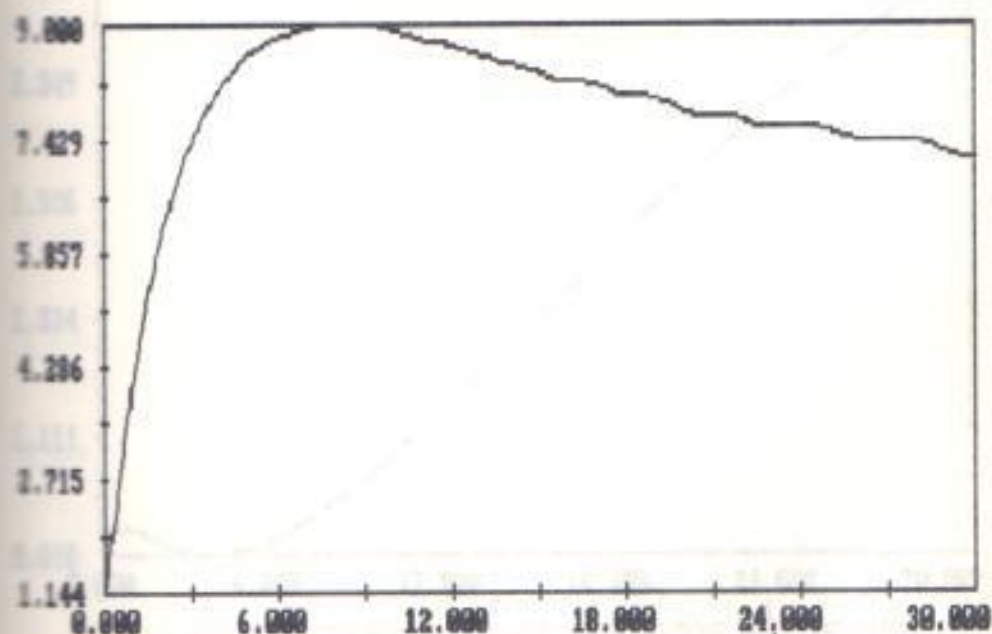


FIG. 4.23 PRODUCTO (q) SI RECURSOS I_0 SE REDUCEN A LA MITAD

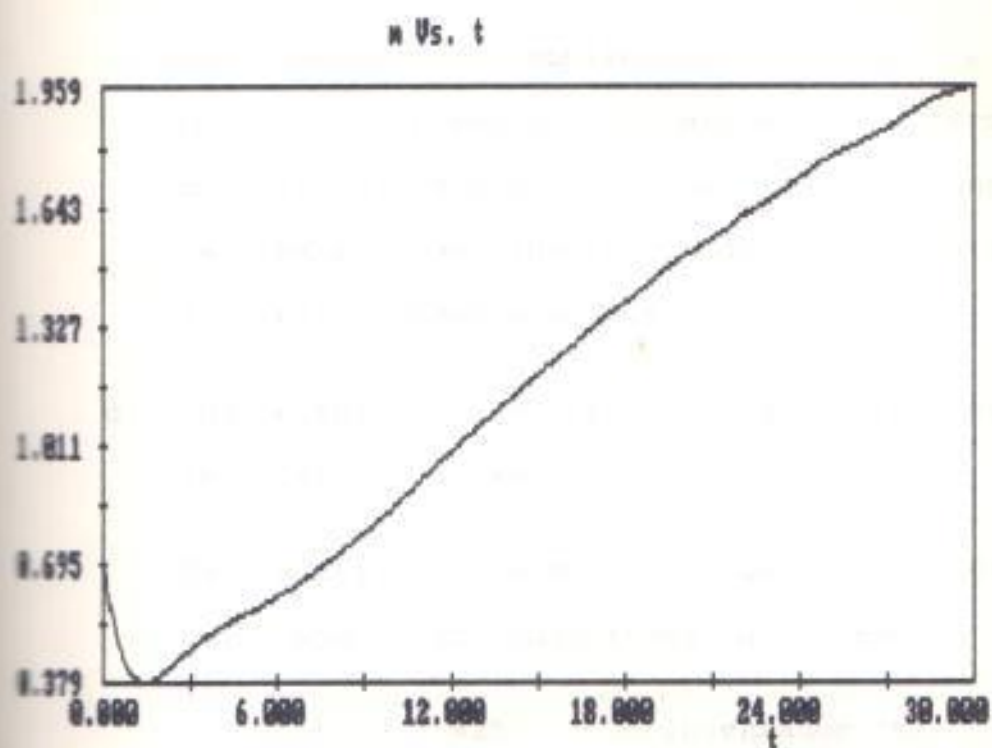


FIG. 4.24 DINERO (M) SI RECURSOS I_0 SE REDUCEN A LA MITAD

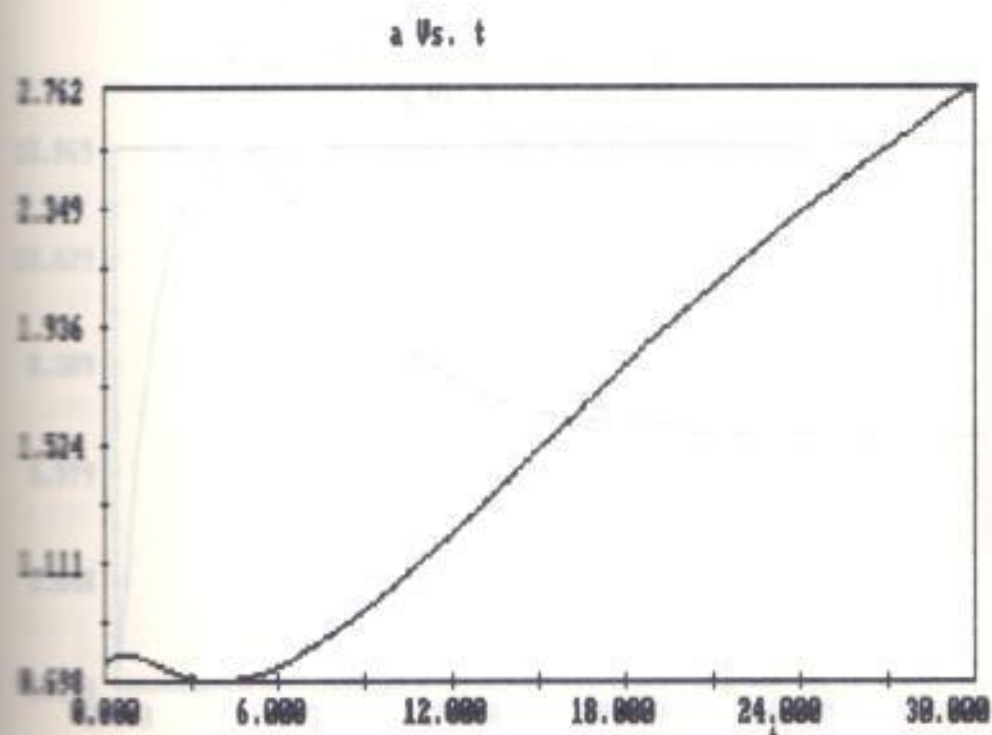


FIG. 4.25 ACTIVOS (A) SI RECURSOS I_0 SE REDUCEN A LA MITAD.

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:

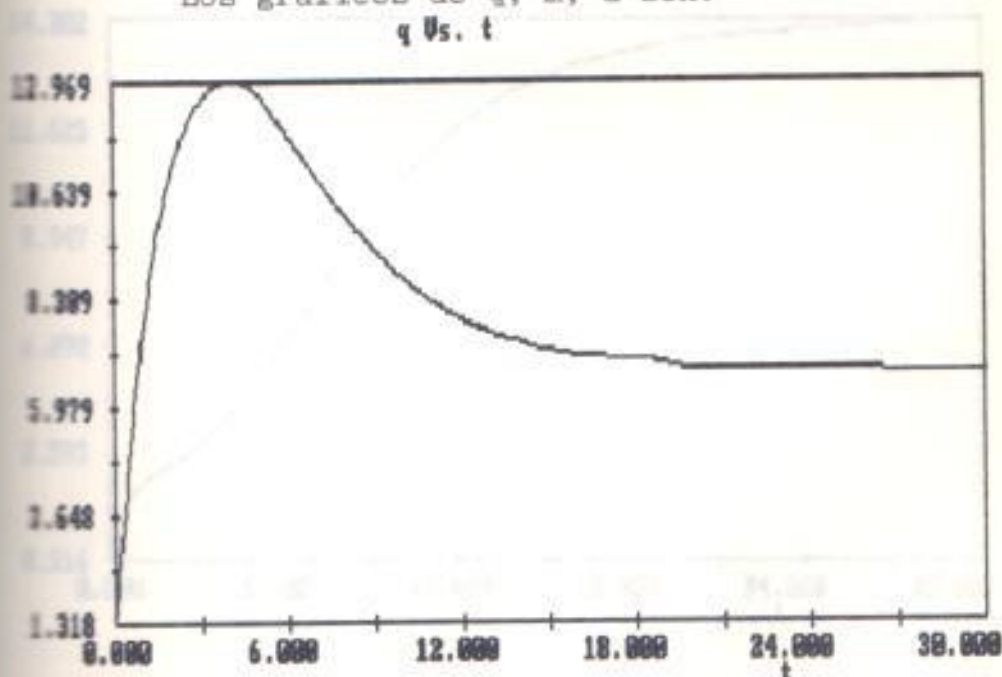


FIG. 4.26 PRODUCTO (Q) SI INVERSION INICIAL ES 10 VECES MAYOR

n Us. t

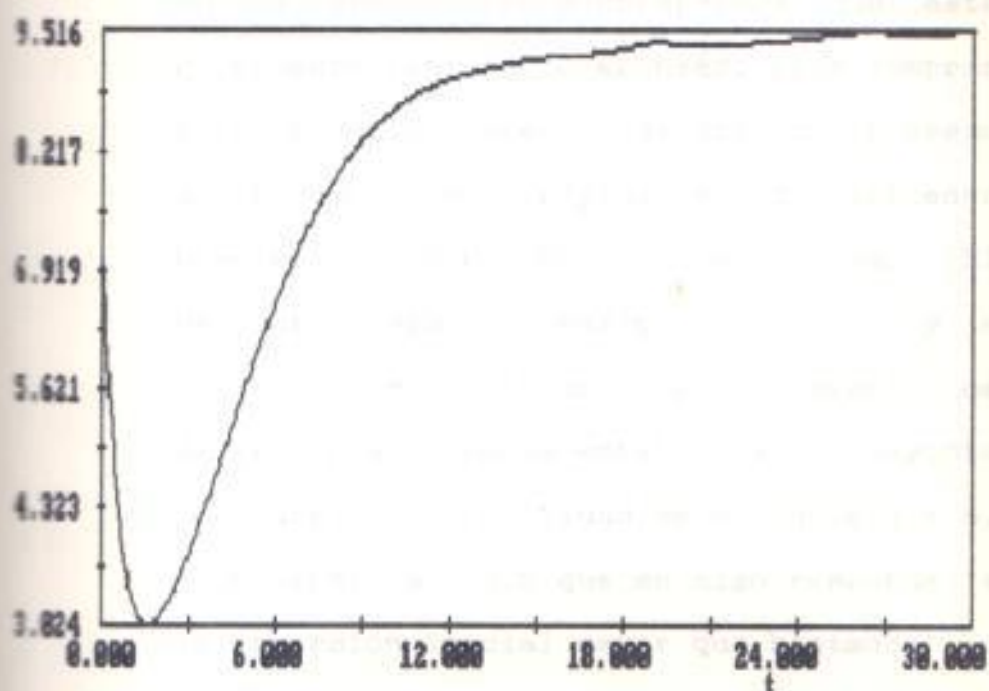


FIG. 4.27 DINERO (M) SI INVERSION INICIAL ES 10 VECES MAYOR

a Us. t

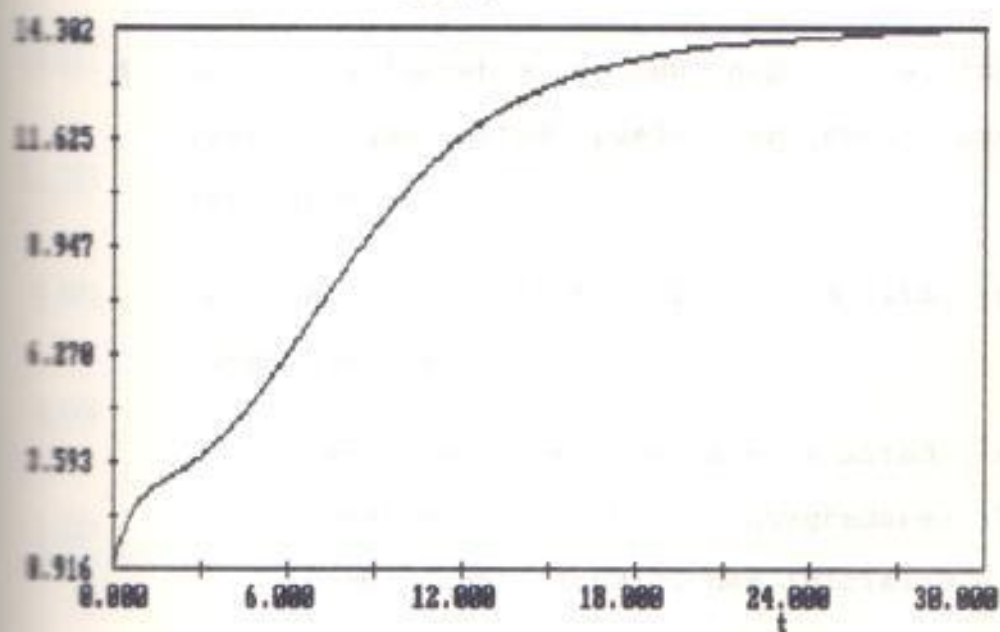


FIG. 4.28 ACTIVOS (A) SI INVERSION INICIAL ES 10 VECES MAYOR

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:

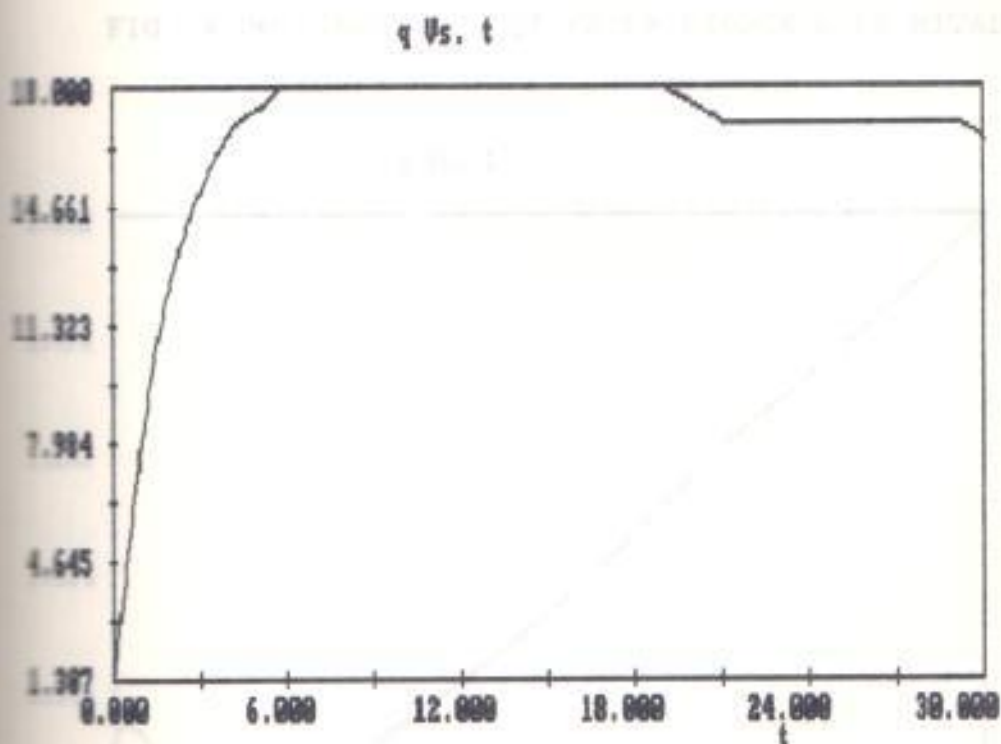


FIG. 4.29 PRODUCTO (Q) SI PE SE REDUCE A LA MITAD

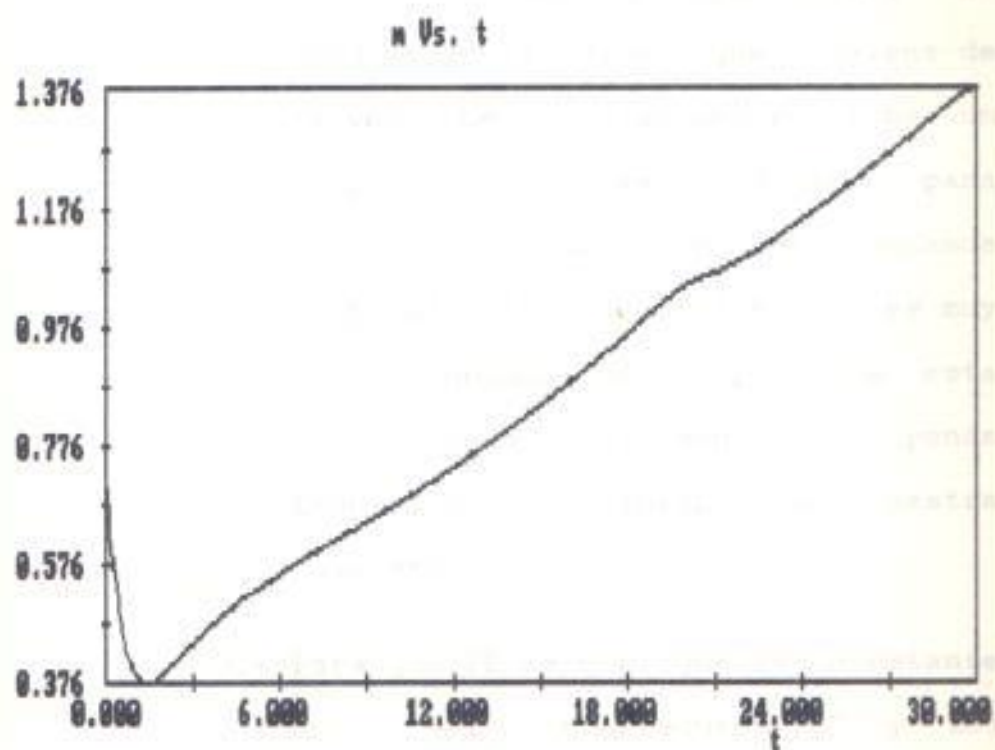


FIG. 4.30 DINERO (M) SI PE SE REDUCE A LA MITAD

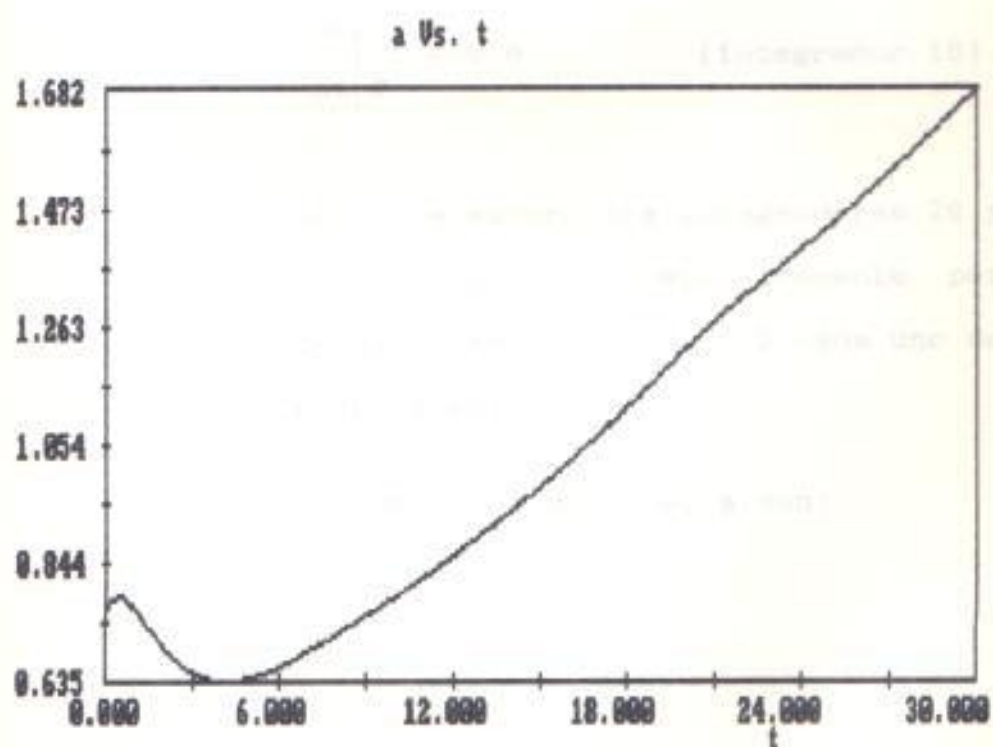


FIG. 4.31 ACTIVOS (A) SI PE SE REDUCE A LA MITAD.

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

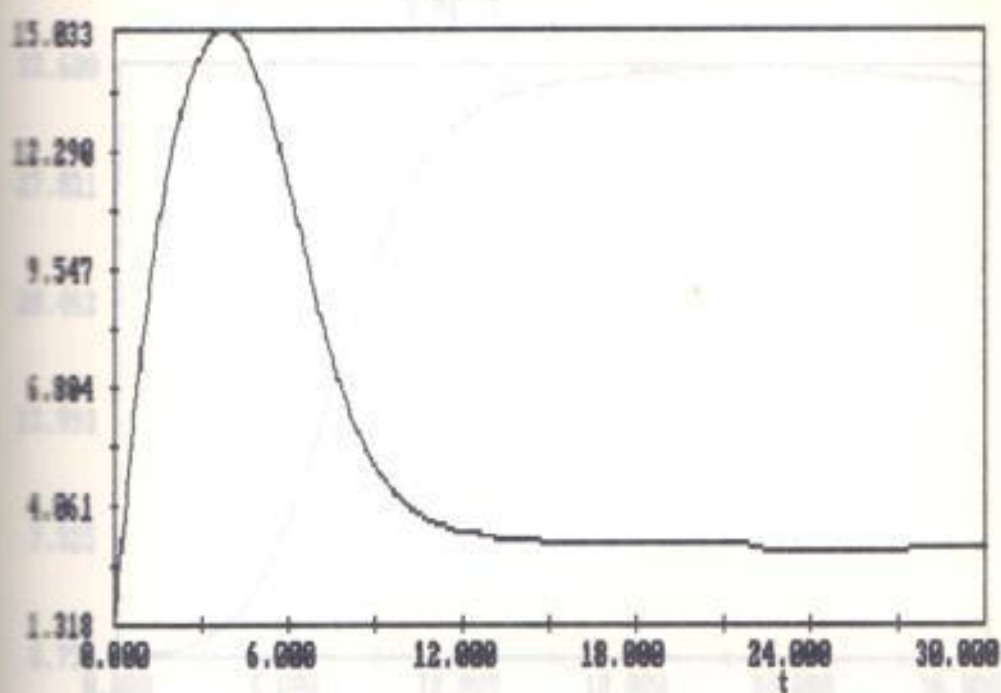


FIG. 4.32 PRODUCTO (Q) SI PE SE DUPLICA

n Vs. t

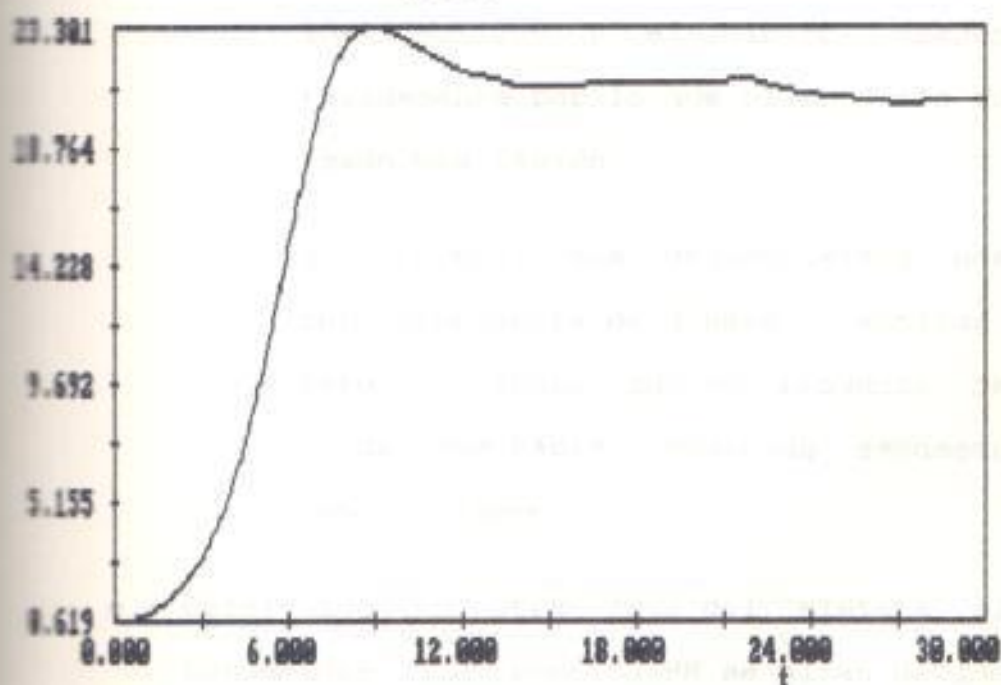


FIG. 4.33 DINERO (M) SI PE SE DUPLICA

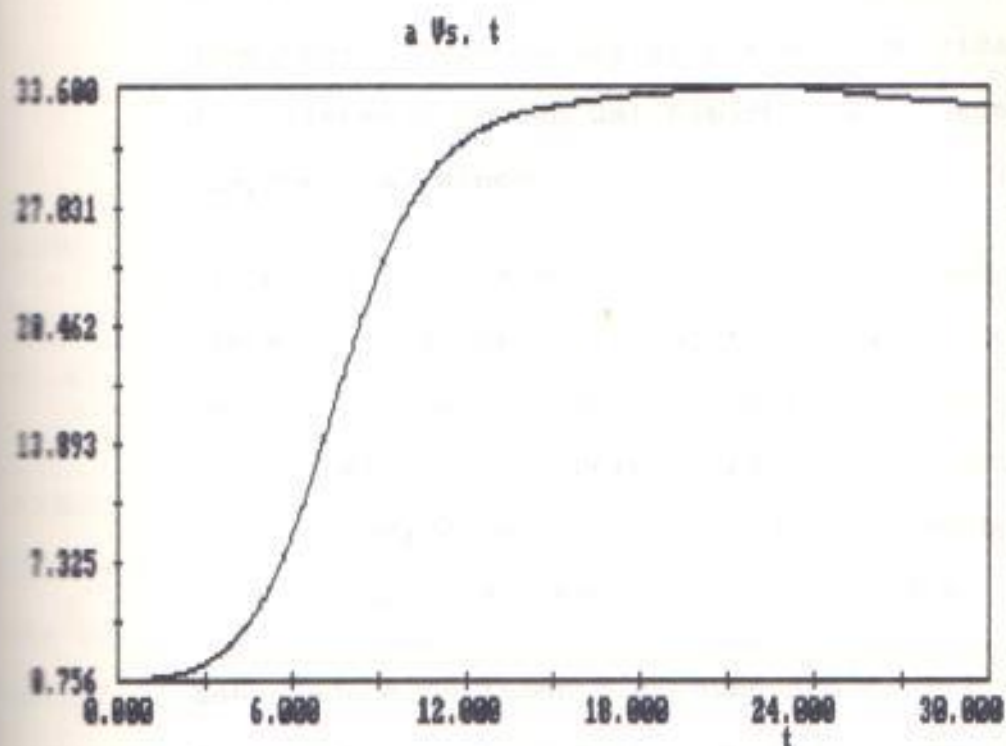


FIG. 4.34 ACTIVOS (A) SI PE SE DUPLICA

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 Techniques. 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.