

**UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE ESTUDIOS TECNOLÓGICOS**



"SOFTWARE DIDÁCTICO DE MODULACIÓN DELTA"



**TRABAJO DE GRADUACIÓN
PREPARADO PARA LA FACULTAD
DE INGENIERÍA**

**PARA OPTAR AL GRADO DE:
"TÉCNICO EN INGENIERÍA ELECTRÓNICA"
ESPECIALIDAD DE COMUNICACIONES**

**POR
PERDOMO ESCOBAR, HENRY JAVIER**

ABRIL-1997

SOYAPANGO

-

EL SALVADOR

-

CENTROAMERICA

DEDICATORIA:

A DIOS TODOPODEROSO.

A MIS PADRES: JOSÉ EVARISTO PERDOMO Y MARÍA MAGDALENA DE PERDOMO.

A MIS HERMANOS: MARVIN GUSTAVO, JOSÉ ALEXANDER, WALTER OSVALDO, HERBERT WILFREDO, FRIDA LORENA.

A MI FAMILIA Y AMIGOS EN GENERAL.

A TODAS ESAS PERSONAS QUE DE UNA U OTRA MANERA COLABORARON PARA EL LOGRO DE ESTA META.

HENRY J. PERDOMO.

AGRADECIMIENTO:

AL HACEDOR POR BRINDARME ESTA OPORTUNIDAD. DE HABERME DADO CUENTA QUE TODAS LAS PERSONAS SOMOS CAPACES DE HACER LO QUE NOS PROPONEMOS.

A MIS PADRES: JOSÉ EVARISTO PERDOMO Y MARÍA MAGDALENA DE PERDOMO POR SU APOYO Y POR SER SIEMPRE UNOS PADRES EJEMPLARES Y DIGNOS DE MUCHO RESPETO.

A MIS HERMANOS: MARVIN GUSTAVO. JOSÉ ALEXANDER. WALTER OSVALDO, HERBERT WILFREDO, FRIDA LORENA, QUE SIEMPRE ESTUVIERON APOYANDOME MORAL Y ESPIRITUALMENTE.

A MIS AMIGOS: OSCAR VILLALTA (ASESOR), ING. EDUARDO DURÁN. ING. GONZALO ESCOBAR, PROF. OMAR VAZQUES, ING. MARIO MONGE, LIC. MORRIS ALVARADO, DR. OSCAR RODRIGUEZ, DON RONALD ANZORA. BR. MIGUEL MARTINEZ Y DEMÁS MIEMBROS. TODOS ELLOS MIS MAS SINCERO RESPETO Y GRACIAS POR SER AMIGOS VERDADEROS.

A ESA FAMILIA QUE APRECIO MUCHO "FAMILIA DURÁN BARRAZA" EN ESPECIAL A MI NOVIA "PATRICIA", POR QUE ES UN TRIUNFO DE AMBOS.

HENRY J. PERDOMO.

AUTORIDADES UNIVERSITARIAS:

Ing. Federico Miguel Huguet Rivera
Rector de la Universidad Don Bosco.

Ing. Carlos Gutierrez
Decano de la Facultad de Ingeniería.

Ing. Ángel Ernesto Soriano
Director del Tecnológico de la Universidad Don Bosco.

PROLOGO

Antes de escoger el tema de el Trabajo de Graduación . se detectó un factor importante "La Necesidad". Partiendo de ahí se buscó una forma de como llenarla mediante la realización de un proyecto que estuviese dentro de la realidad nacional.

Las señales de información se han venido transmitiendo análogamente, es decir, sin sufrir ninguna clase de conversión para ser trasladadas a un lenguaje simbólico. Sin embargo, en la actualidad está dominando las técnicas de transmisión, un método en el cuál las señales son previamente transformadas a un lenguaje simbólico representado por niveles lógicos, unos y ceros: todo esto para poder enviarlos por un canal de transmisión, y en lugar de la recepción, estas señales son retornadas a su forma original, reconstruidas hasta obtener la misma que fue transmitida al principio. Debido a la gran demanda de estos sistemas y a la expansión de la tecnología, es necesario la capacitación de personas para estos sistemas.

La idea de crear un sistema de enseñanza adicional a los tradicionales se llevó en forma pausada, se obtuvo información, se analizaron las condiciones de los Laboratorios en El Tecnológico de la Universidad y se hecho a andar el proyecto. Es así como se presenta el "Software Didáctico de Modulación Delta", Como un instrumento más para los estudiantes de tecnológico y como complemento para sus prácticas de laboratorio.

INTRODUCCIÓN

Es preciso notar la importancia que han tenido las comunicaciones en el desarrollo de la humanidad, la transmisión de conocimientos no hubiese sido posible si el hombre no contara con los mecanismos para efectuar el intercambio de ideas.

En un principio el hombre comenzó a utilizar métodos rudimentarios para comunicarse con los demás. A un inicio, fueron las señales de humo, tambores y medios similares, limitando así el contenido y el alcance de estos mensajes. Posteriormente con el desarrollo de la civilización, se pasó a la etapa de la comunicación escrita. Fue en el siglo XIX, cuando se dio un salto grande en las comunicaciones, aparecen hombres como Hertz, Marconi y Edison, que vinieron a demostrar la posibilidad de enviar la voz humana a distancias largas. A partir de este paso, fue posible posteriormente la transmisión de datos, logrando adelantos tecnológicos en este campo.

Debido a la complejidad de los núcleos poblacionales, la comunicación se ha vuelto cada vez más vital para el desarrollo de los pueblos. El Salvador no es la excepción. Instituciones públicas y privadas han desarrollado proyectos que emplean sistemas de comunicación, tratando de satisfacer la demanda de la población, comunicarse mediante la red telefónica.

A través de este trabajo se ofrece un método de soporte para la enseñanza de uno de estos sistemas . El "Software didáctico de Modulación Delta" presenta en forma clara la función de un sistema Delta.

La estructura general del trabajo está dado en tres capítulos y se desglosan de la siguiente manera.

CAPITULO I:

Se hace mención de los aspectos generales del proyecto, los objetivos generales y específicos alcanzados, además de las justificaciones y limitantes del proyecto.

CAPITULO II:

Se habla de los sistemas de comunicación digital. multiplexación digital, haciendo mención de la digitalización como principio del sistema Delta. En este capítulo se abordan los conceptos generales de la Modulación Delta y su significado.

CAPITULO III:

Sobre la elaboración de el Software: Se explica las técnicas de programación utilizadas para la elaboración de los distintos programas en que compone el Software.

FINALIDAD DEL PROYECTO

La finalidad primordial de este proyecto consiste en hacer posible la capacitación del estudiante acerca de los diferentes bloques que conforman un sistema de Modulación Delta.

Por medio de este proyecto se puede ver paso por paso el comportamiento del sistema Delta, es por eso que se optó por poner separadas las señales y no en una sola cuadrícula dibujada en el monitor.

Que el usuario tenga acceso a la teoría de la modulación Delta, tanto como a la simulación y por supuesto hacer una práctica introduciendo sus propios datos.

INDICE.

CAPITULO I:

1 ASPECTOS GENERALES.....	1
1.1 OBJETIVOS.....	1
1.1.1 OBJETIVOS GENERALES.....	1
1.1.2 OBJETIVOS ESPECÍFICOS.....	1
1.2 JUSTIFICACIÓN DEL TEMA.....	2
1.3 ALCANCES Y LIMITACIONES.....	3
1.3.1 ALCANCES.....	3
1.3.2 LIMITACIONES.....	4
1.4 SITUACIÓN ACTUAL.....	4

CAPITULO II:

2 SISTEMAS DIGITALES DE COMUNICACIÓN.....	6
2.1 MULTIPLEXACIÓN DIGITAL.....	10
2.2 POR QUÉ LA DIGITALIZACIÓN.....	14
2.2.1 DESVENTAJAS DE LA DIGITALIZACIÓN.....	16
2.2.2 CARACTERÍSTICAS DE LA VOZ.....	18
2.3 MODULACIÓN EN DELTA.....	22
2.3.1 MODULACIÓN DELTA LINEAL	22
2.3.2 MD, MCP, MCPD.....	27
2.3.3 UMBRAL DE CODIFICACIÓN Y SOBRECARGA DE PENDIENTE.....	29
2.3.4 INTEGRACIÓN DOBLE.....	33

CAPITULO III:

3 ELABORACIÓN DEL SOFTWARE DIDÁCTICO.....	37
3.1 LENGUAJE TURBO PASCAL.....	37
3.1.1 RESEÑA HISTÓRICA.....	37
3.1.2 EL TURBO PASCAL Y OTRAS MODALIDADES DEL PASCAL.....	39
3.1.3 ESCRITURA DE UN PROGRAMA EN PASCAL.....	41
3.1.4 LA ESCRITURA DEL LENGUAJE PASCAL.....	42
3.1.5 FORMATOS DE UN PROGRAMA EN PASCAL.....	43
3.1.6 DECLARACIONES.....	44
3.1.7 RESUMEN DE LA ORGANIZACIÓN DE UN PROGRA- MA.....	45
3.1.8 ORGANIZACIÓN FORMAL DE UN PROGRAMA.....	46
3.2 LENGUAJE TURBO C.....	47
3.2.1 INTRODUCCIÓN AL C.....	47
3.2.2 HISTORIA DEL C.....	49
3.2.3 ESTRUCTURA DE UN PROGRAMA EN C.....	50
3.2.4 CARACTERÍSTICAS DESEABLES EN UN PROGRA- MA.....	55

ANEXOS:

ANEXO 1.....	58
--------------	----

- MANUAL DE USO DEL SOFTWARE.
- GUÍAS DE PRÁCTICAS DEL PROGRAMA "DELTA".
- PRESENTACIÓN DE TEORÍA DENTRO DEL SOFTWARE

ANEXO 2.....	76
--------------	----

- GRÁFICAS DE SIMULACIÓN
- GRÁFICAS DE PRÁCTICAS

ANEXO 3.....	90
--------------	----

- ALGORITMOS Y FLUJOGRAMAS
- LISTADO DE PROGRAMAS

REFERENCIAS BIBLIOGRÁFICAS:

BIBLIOGRAFÍA.....	116
-------------------	-----

CAPITULO I

1. ASPECTOS GENERALES.

1.1 OBJETIVOS.

1.1.1 OBJETIVOS GENERALES.

- Diseñar un Software que simule el funcionamiento de un sistema de modulación delta, mediante un algoritmo que siga el comportamiento del sistema.
- Aplicar los conocimientos adquiridos de una forma más eficiente.
- Basar el sistema de aprendizaje en medios visuales.
- Optimizar el proceso de enseñanza.

1.1.2 OBJETIVOS ESPECÍFICOS.

- Que el estudiante pueda introducir los valores de frecuencia de la señal muestra y su amplitud, frecuencia de muestreo y el delta: para observar los cambios en respuesta con diferentes valores.

- Presentar una introducción teórica dentro del Software, que de al estudiante una clara idea de lo que trata el tema de modulación delta.
- Presentar un manual de uso del Software. esto incluye una guía de práctica con instrucciones claras e ilustraciones del programa.
- Disponer al final del programa de una sección de evaluación, acerca de la modulación en delta, por ejemplo situaciones críticas o de alguna falla en un sistema delta.
- Hacer un disco "INSTALADOR" que permita grabar el programa en cualquier PC.

1.2 JUSTIFICACIONES DEL TEMA.

Lo que se busca con el Software didáctico de modulación delta es que un mayor número de alumnos tengan acceso a usuario, por eso se pensó en el disco instalador, así cada alumno tendrá acceso personal a su programa.

Los Alumnos del Tecnológico podrán capacitarse mejor, ya que se ha pensado en el Software como una forma de complementar

los métodos de enseñanza ya conocidos.

El sistema de modulación delta se utiliza como digitalizador de señales para telesupervisión, es el convertidor análogo-digital de más bajo costo y es conocido además como el sistema de radio comunicaciones digitales mas económico.

1.3 ALCANCES Y LIMITACIONES.

1.3.1 ALCANCES.

En la sección del Software que contiene la teoría, se incluyo lo necesario para que a nivel tecnológico se pueda comprender el sistema de modulación delta. Cuenta con gráficos ilustrativos que se explican en conjunto con la teoría.

En lo que a funcionamiento respecta, en la parte de simulaciones se puede decir que se han obtenido muy buenos resultados, con la implementación de un algoritmo que sigue muy de cerca el funcionamiento de un sistema delta, se logró llegar a una simulación muy buena de las señales escogidas en este caso para digitalizarlas.

En la parte de prácticas se utilizó el mismo principio del algoritmo, modificándolo para poder así introducir nuestros propios datos y se obtuvo muy buena resolución en las gráficas.

- Módulos de Elettronica Veneta con sus respectivas guías teóricas y prácticas.
- Un software de tipo didáctico que se utiliza con una llave de Hardware y solo se puede utilizar en una máquina a la vez.

4. Las señales digitales pueden ser codificadas para obtener índices de error extremadamente bajos y alta fidelidad, así como privacidad.

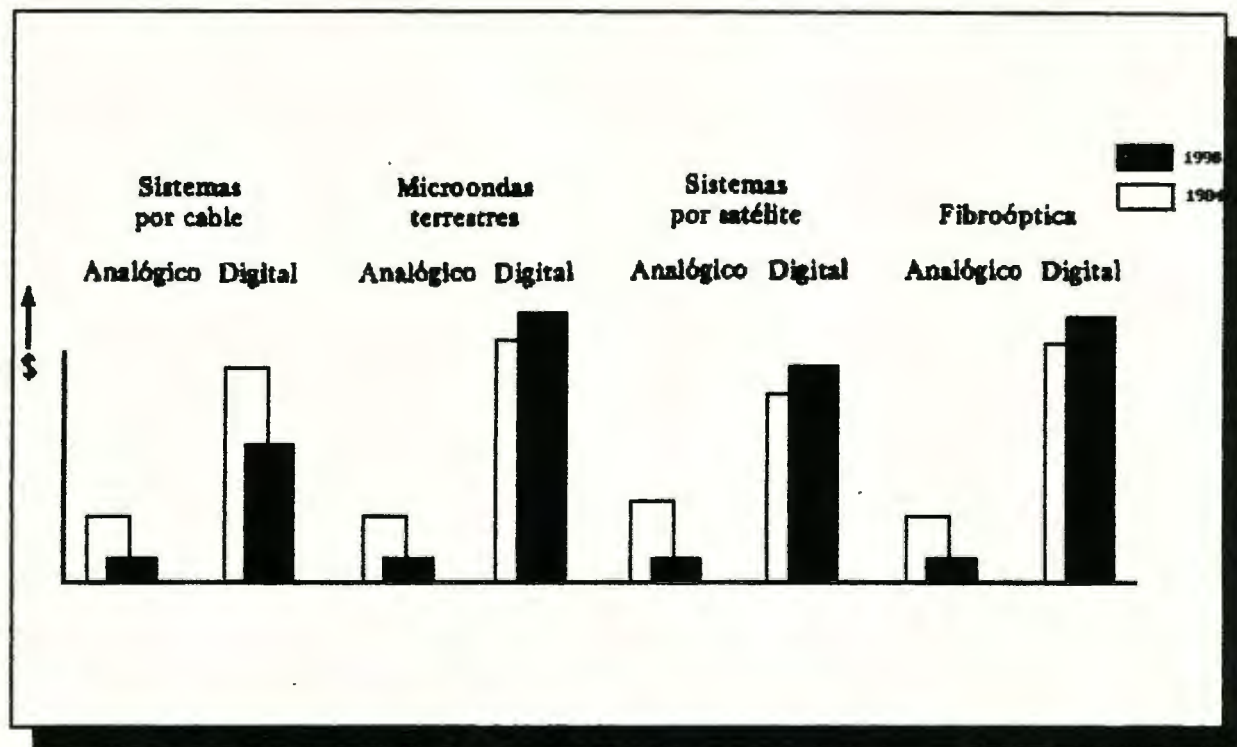


Figura 1.1: Proyecto de inversión para 1984 y 1990 del sistema de transmisión de banda ancha de la industria telefónica de distancias medias (500 a 2000 km) de Estados Unidos y Canadá.

5. Es más fácil y efectiva la multiplexación de varias señales digitales.
6. La comunicación digital es inherentemente más eficiente que la analógica en la realización del intercambio de la Relación Señal Ruido (RSR) por ancho de banda.

La entrada a un sistema digital se encuentra en la forma de una sucesión de dígitos. Podría ser la salida de un conjunto de datos, o de una computadora, o de una señal de voz digitalizada (MCP), facsímil digital o TV, o datos de telemetría, etc.

Las señales que provienen de varias fuentes digitales se pueden combinar mediante un multiplexor digital utilizando el proceso de intercalación. La salida del multiplexor se codifica en pulsos eléctricos para el propósito de transmisión a través de una línea digital. Existen varias formas posibles de lograr esto. Conceptualmente, la *transmisión* más simple o *código de línea*, es la de *encendido-apagado*, donde 1 se transmite mediante un pulso $p(t)$ y 0 indica la ausencia de un pulso. Otro código de línea que se utiliza comúnmente es el *polar* donde 1 se transmite mediante un pulso $p(t)$ y 0 se transmite como el pulso $-p(t)$.

Este es el código más eficiente, para una potencia dada de transmisión, ya que es el más inmune al ruido. Otro código de línea que se utiliza en la MCP es el bipolar o el pseudoternario, donde 0 se transmite por ausencia de pulso y 1 se transmite como un pulso $p(t)$ o $-p(t)$, dependiendo de si el 1 anterior se transmitió como $-p(t)$ o $p(t)$. En resumen los pulsos que presentan 1's consecutivos se alternan como se indica en la Fig. 1.2.

Este código presenta la ventaja de que un solo error en la detección de los pulsos viola la regla bipolar de los pulsos alternados, y el error se detecta rápidamente (aunque no se corrige).

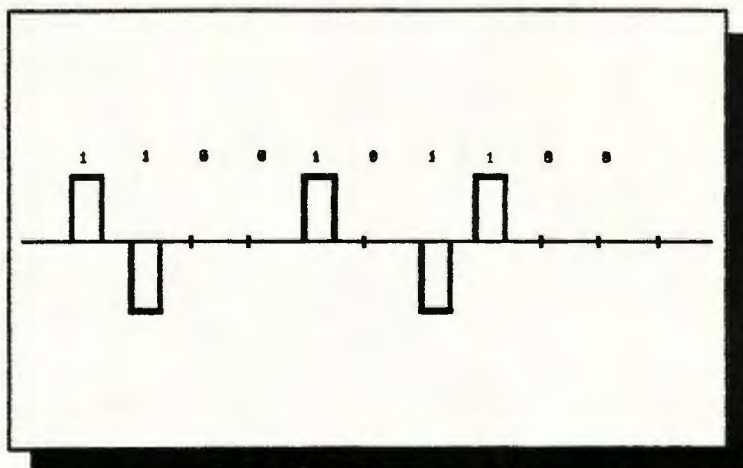


Figura 1.2 :Señal Bipolar

Los repetidores regenerativos se utilizan a intervalos regularmente espaciados a lo largo de una línea de transmisión digital para detectar la señal digital entrante y generar nuevos pulsos limpios para su posterior transmisión a través de la línea.

Este proceso periódicamente elimina, y en esta forma combate, la acumulación de ruido y distorsión de la señal, para detectar los pulsos, la información de sincronización, que hace que el transmisor y el receptor estén conectados al mismo canal, para muestrear cada pulso en el instante apropiado, debe estar disponible en cada repetidor. Esta información puede extraerse de la misma señal que se recibe, si el código de línea se elige apropiadamente.

Por ejemplo una señal digital, tal como una señal binaria de encendido-apagado, contiene una componente discreta de la misma señal de reloj. Esto también se puede ver en el hecho de que tal forma de onda se puede expresar como una suma de dos formas de

onda: 1) una componente aleatoria y 2) una componente periódica con la misma frecuencia fundamental de reloj.

Cuando esta señal se aplica a la entrada de un circuito resonante sintonizando a la frecuencia f_0 , la salida del circuito, que es una senoide de frecuencia f_0 , puede usarse para sincronización.

Esta señal de sincronización (la salida del circuito resonante) es sensible al patrón de bits entrante. Si existen demasiados 0 en una sucesión (ausencia de pulsos), la salida senoidal del circuito resonante comienza a decaer produciendo así error en la información de cronización. Este problema puede resolverse mediante el uso de un código de líneas apropiado o bien utilizando mezcladores de voz.

2.1 MULTIPLEXACIÓN DIGITAL

Ciertas señales de bajo índice de bit pueden multiplexarse, o combinarse, para formar una señal de alto *índice de bit*¹ para su transmisión a través de un medio de alta frecuencia. Ya que el medio es de tiempo compartido por varias señales entrantes, este es un caso de MDT. Las señales que provienen de varios canales de entrada, o tributarios, pueden ser naturaleza tan diversa como una

¹Índice de bit: se refiere a la velocidad de transmisión de una señal en forma serie, utilizando una sola línea de transmisión.

señal de voz digitalizada (MCP), la salida de una computadora, datos de telemetría, un facsímil digital , etc . El índice de bit de los diferentes tributarios no necesita ser el mismo.

Para comenzar, consideramos el caso de todos los tributarios con índices de bit idénticos. Se puede efectuar la multiplexión sobre una base de bit a bit (que se conoce como bit o intercalación digital) como se muestra en la Fig. 1.3a. o sobre una base de palabra a palabra (que se conoce como byte o intercalación de palabras).

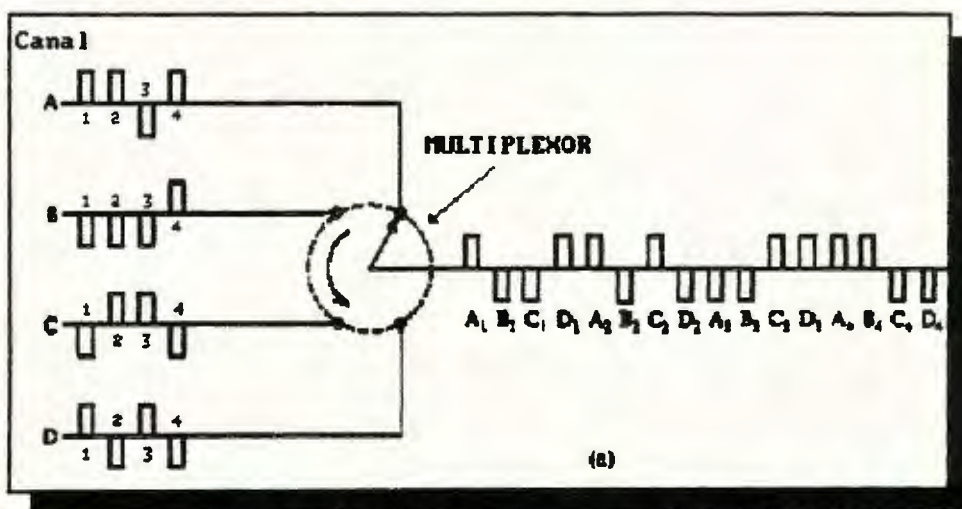


Fig.1.3a: Multiplexación por intercalación digital

La Fig.1.3b muestra la intercalación de palabras, o bytes, formada por cuatro bits. La Fig.1.3c muestra la multiplexación de cuatro

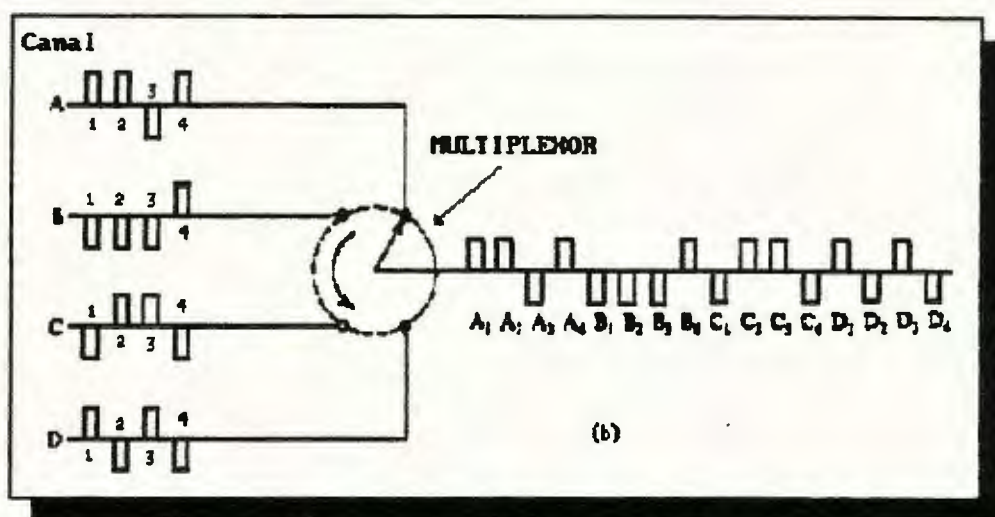


Fig.1.3b: Multiplexión por intercalación de palabras

canales que consta de cuatro canales de idéntico índice de bit B y un canal con índice de bit de $3B$.

Se pueden lograr resultados similares al combinar palabras de diferentes longitudes. Es evidente que la longitud mínima del cuadro de multiplexación debe ser un múltiplo del mínimo común múltiplo de los índices de bit de los canales de entrada y, en consecuencia, este tipo de esquema es práctico solamente cuando existe alguna relación simple entre estos índices.

En la terminal receptora, la corriente de dígitos que llega debe

dividirse y distribuirse hacia el canal de salida apropiado. Para este propósito, la terminal receptora deberá ser capaz de identificar correctamente cada bit. Esto requiere que el sistema de recepción se sincronice unívocamente en el tiempo con el inicio de cada cuadro, con cada espacio dentro de un cuadro y cada bit dentro de un espacio.

Esto se lleva a cabo agregando bits de encuadramiento y de sincronización a los bits de datos. Estos bits forman parte de los llamados *bits de control*.

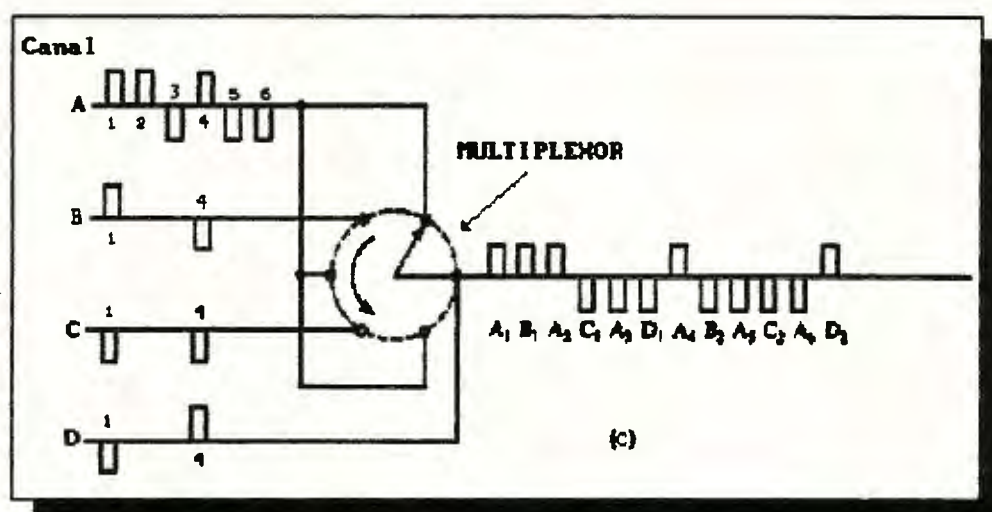


Fig.1.3c: Multiplexión de cuatro canales con 3 canales de idéntico índice de bit B.

2.2 POR QUE LA DIGITALIZACIÓN ?

El problema fue que los cables de enlace entre centrales estaban saturados y la canalización llena. La transmisión se hacía por MDT (Multiplexación por Distribución de Tiempo) por cables coaxiales y para tramos de 3 a 5km se hacía hasta con pares de hilos.

Luego en los años 60's que aparecen los circuitos integrados. Shannon para los Estados Unidos de Norte América y Reves para la República de Francia crearon un modelo matemático que les permitió codificar las señales analógicas (PCM).

De estos factores mencionados anteriormente surge PCM (Pulse Coded Modulation): Estados Unidos y Europa tomaron diferentes estándares.

Estados Unidos adoptó un sistema de 24 canales en MDT. que transmite a una velocidad de 1.55 Mbps con la capacidad de enviar 64 Kbps/canal y se utilizó un método de dos pares de hilos de cobre. 1 par para Tx y el otro par para Rx.

Europa. por su lado creó su sistema de 30 canales al igual que Estados Unidos en MDT. la velocidad de transmisión de 2.048 Mbps con capacidad para enviar 64 Kbps/canal con el método de los dos pares también.

Estos métodos son usados para cables de enlace entre centrales locales, como se tenían más canales por cable se aprovechó más la canalización.

El primer punto para digitalizar fueron los costos, la tecnología digital es más económica que la analógica, especialmente en el área de transmisión.

PCM utiliza tecnología avanzada de microelectrónica, otro aspecto importante es la calidad, si se crean dos equipos, uno análogo y el otro digital, será mejor el digital por que los filtros para el análogo serían de mala calidad.

Otro aspecto es la fiabilidad de los sistemas digitales porque tienen unas tolerancias mucho mayores a las analógicas. Por Ejemplo la marcación por tonos es digital !

El siguiente aspecto es la facilidad de la señalización, cuando un abonado A llama a B que está en otra central, la central A le manda el número a B, eso es digital.

En la marcación por tono MDTF (Multitone Doble Frequency) son solo seis tonos pero se combinan dos para cada tecla. La técnica digital permite una gran gama de servicios por los mismos medios. La técnica digital permite una alta flexibilidad.

La facilidad de encriptar que es la técnica por medio de la cuál se hace una codificación completa del mensaje. La ventaja de la digitalización es la facilidad de almacenamiento. También la inmunidad al ruido es muy alta porque es capaz de detectar errores, incluso eliminarlos. En digital se le ha denominado error de ruido.

En la técnica analógica el ruido es acumulativo, en digital se llama regeneración que es la reconstrucción del pulso. Hay que tener cuidado con la distancia a la que se deben poner los regeneradores porque se podrían generar errores. La desventaja es que se generan retardos de tiempo o inclusive adelanto. La variación de la fase en un regenerador se conoce como JITTER.

2.2.1 DESVENTAJAS DE LA DIGITALIZACIÓN

Como primer punto se tiene como desventaja la necesidad de conversión, al transmitir y al recibir. Al contrario de la señal analógica que se transmite y se recibe siempre analógica.

Otro problema es el ancho de banda, para el canal telefónico el ancho de banda que es de (0.3-3.4Khz) y para el canal digital que tiene un caudal de 64 Kbps (PCM), esto se refiere a la cantidad de información por segundo.

Según el teorema de NYQUIST

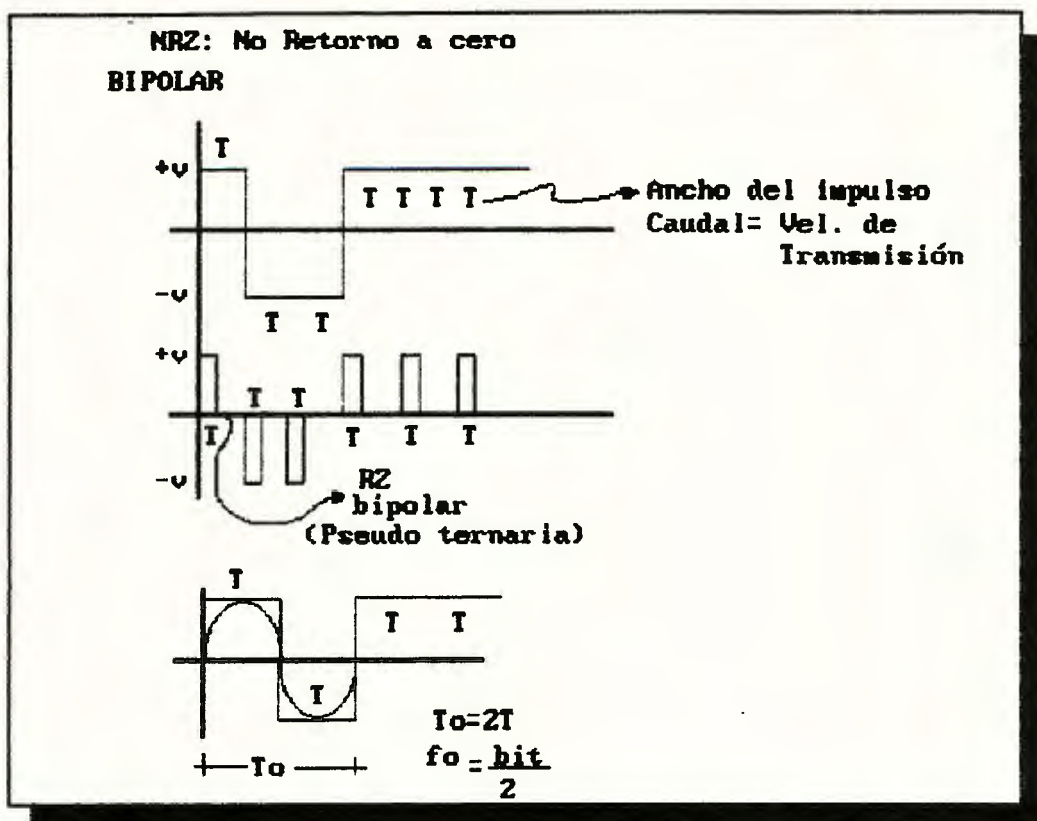


Fig. 1.4 Comparación de códigos de línea

En realidad el criterio de Nyquist da la frecuencia fundamental de la señal digital.

La duración del bitio (T) en la Fig. 1.4, es la mitad del período de la frecuencia fundamental. Por cada 2 bitios hay un periodo de la frecuencia fundamental, por lo tanto es el doble de la frecuencia fundamental.

La cantidad de bitios para ambos ejemplos es igual pero el Ancho de Banda del segundo es el doble de la del primero. O sea que solo en NRZ $\text{bit/s} = \text{baudio}$.

Para regenerar el reloj en el primer ejemplo es más difícil que para el segundo.

Para 64kbps $\rightarrow [64\text{Baudios (NRZ)} / 2] = 32\text{KHz}$

Cualquier tipo de información se puede convertir a sistema digital. ahora la preocupación de las telecomunicaciones no es la calidad sino la cantidad de servicio que se pueden ofertar.

La voz se convierte a una señal eléctrica a través de un transductor mecánico-eléctrico. Hay ciertos parámetros característicos que predominan como rangos de frecuencia, amplitud, etc.

2.2.2 CARACTERÍSTICAS DE LA VOZ.

1) *Redundancias en el dominio del tiempo*

1.1- *Distribución no uniforme de amplitud:* No todos los valores están presentes en un tiempo de observación. No todas las amplitudes tienen la misma probabilidad de aparecer. En el caso de no uniformes hay valores que pueden aparecer muchas veces y hay valores que pueden no aparecer.

En la señal de voz predominan las bajas amplitudes.

1.2- *Correlación muestra a muestra:* Correlación es redundancia o similitud.

En una señal de voz la correlación entre cada muestra es

del 85% en promedio para frecuencias de muestreo f_s de 8000Hz.

Si aumenta la f_s aumenta la correlación.

1.3- *Correlación ciclo a ciclo*: El sonido es una vibración que tiene una frecuencia fundamental y armónicos.

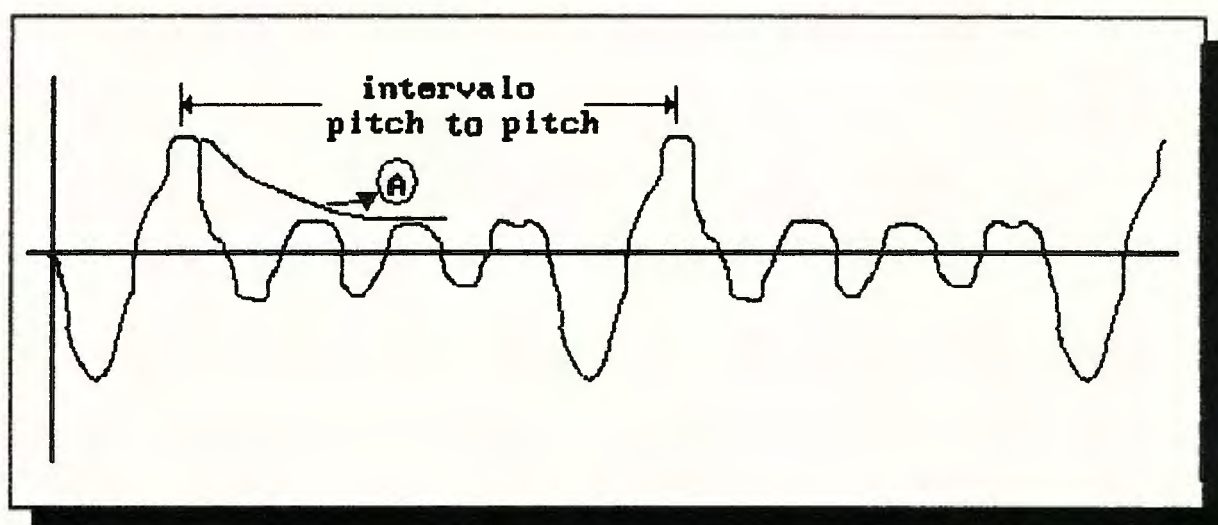


Fig. 1.5: Correlación ciclo a ciclo.

Para que se pueda escuchar un tono no basta con un ciclo, tienen que haber varios para que el oído pueda interpretarlos. Para que se escuche un fonema se debe repetir. Con técnicas más avanzadas de digitalización se puede memorizar un intervalo y enviar el número de veces que ha de repetirse.

La oscilación de las ondas vocales está entre 300 y 450Hz. Los fonemas son unidades fundamentales del habla. La correlación ciclo a ciclo es grande en 20 ms de la señal.

1.4- *Correlación en un intervalo pitch a pitch*: Viendo el punto A se da dentro de un intervalo una relación de amplitud amortiguada.

1.5- *Factores de inactividad*: Las máquinas que trabajan con factores de inactividad se llaman TASI (Time Assignment Speech Interpolation). El circuito que detecta los tiempos de inactividad se llama DSI (Digital Speech Interpolation).

El 40% de todo el tiempo una persona escucha o hace pausa en la conversación. Se deja un 20% para cuando ambos interlocutores hablan. Los factores de inactividad surgen de la interactividad de la voz o sea solo en conversación.

La distribución no uniforme genera la técnica adaptativa y la técnica muestra a muestra diferencial.

El análisis en el dominio de la frecuencia se da en:

- a) *Pequeños períodos*: Frecuencias que se atenúan.
- b) *Grandes períodos*: Hay distribución no uniforme de la frecuencia.

Con la técnica PCM se pueden digitalizar cualquier señal. una aplicación es la digitalización de la voz. Hay que limitar el ancho de banda (BW) y la amplitud. (BW será el Rango Dinámico)

Se estableció como frecuencia de muestreo 8KHz que es un poco superior al doble de la frecuencia mayor de la señal de mensaje. No puede ser más grande porque se tiene que acercarse al criterio de Nyquist y no aumentar el caudal de bits. Se hace así por que el costo de los filtros es más barato.

El primer paso es el filtrado, luego se obtiene la señal no continua en el tiempo (PAM). Sigue la cuantización que se caracteriza por el paso de cuantización.

Cuando se da la cuantización se cambian los valores originales para aproximar, entonces se genera el error de cuantización.

La señal de salida del cuantizador daría una señal escalonada, uniforme porque todos los escalones tienen la misma amplitud. Luego al trazar una señal desde el punto inicial hasta el punto final, da la señal rampa. La salida no es una réplica exacta pero eso (*el ruido de cuantización*) se disminuye incrementando los pasos de cuantización.

2.3 MODULACIÓN EN DELTA

Básicamente la *modulación en delta* (MD) se puede considerar como un método simple de convertir señales analógicas en señales digitales. Este proceso se realiza mediante las dos operaciones clásicas de muestreo y codificación de la señal a transmitir, por ello la modulación Delta se puede considerar un sistema MCP todos los efectos (aunque si con el término MCP se identifica una técnica de codificación específica).

Se distingue de la MCP por su simplicidad y eficiencia. El modulador Delta tiene la ventaja de que la circuitería precisada para el modulador y el demodulador es más sencilla que la precisada para el MCP tradicional.

2.3.1 Modulación Delta Lineal.

Un sistema de comunicación que utiliza la modulación Delta Lineal se muestra en la fig. 1.6a y la fig. 1.6b muestra las formas de onda características. Un generador de impulsos suministra al modulador un tren de impulsos $P_i(t)$. En la otra entrada el modulador recibe también la señal $\Delta(t)$. La salida $P_o(t)$ del modulador no es otra cosa que el tren de impulsos de entrada $P_i(t)$ multiplicado por +1 ó -1, en función del signo (no de la amplitud) de $\Delta(t)$. Si $\Delta(t)$ es positiva cuando llega el impulso $P_i(t)$,

entonces la multiplicación es por +1, mientras que si $\Delta(t)$ es negativa, la multiplicación será por -1.

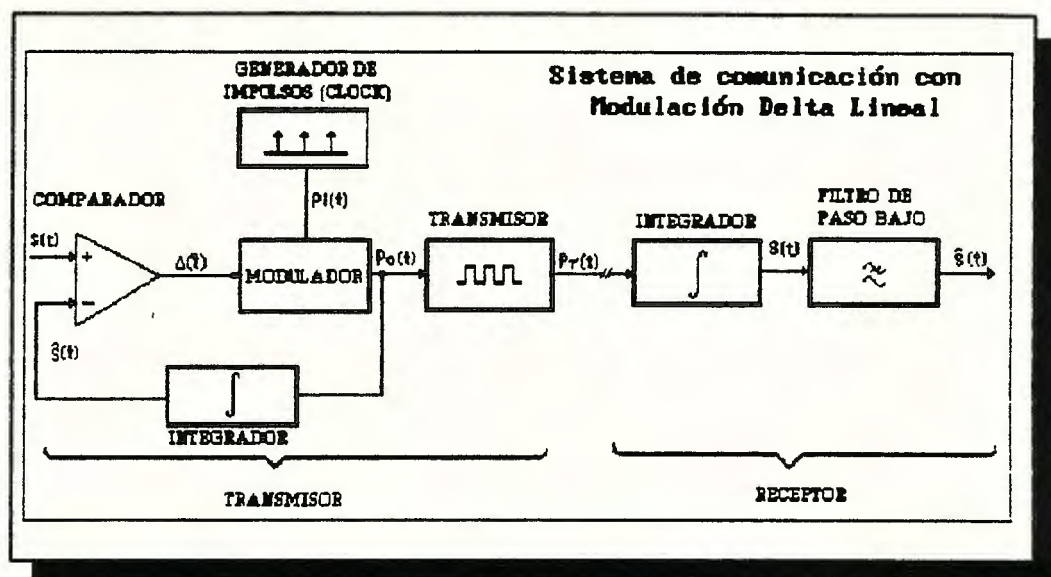


fig. 1.6a. Sistema de comunicación con Modulación Delta lineal.

La señal impulsiva $P_0(t)$ se aplica a un integrador en cuya salida encontramos $s'(t)$. Como se observa a través de la fig. 1.6b, la señal escalonada $s'(t)$ es una aproximación de la señal analógica de entrada $s(t)$.

Las señales $s(t)$ y $s'(t)$ son comparadas por un circuito comparador, cuya salida será positiva [$\Delta(t) > 0$ si $s'(t) > s(t)$] o negativa [$\Delta(t) < 0$ si $s'(t) < s(t)$].

Por lo tanto, ahora está claro el significado físico de $\Delta(t)$ como diferencia (en signo y no en amplitud) entre la señal a codificar $s(t)$ y la señal que se aproxima $s'(t)$.

Teóricamente se podría pensar de enviar al canal de comunicación el tren de impulsos positivos y negativos $P_0(t)$: sin embargo, en la práctica se envía en línea una señal de dos niveles obtenida de $P_0(t)$, de

modo que se aumente la energía de la señal transmitida. En línea encontramos la señal $P_r(t)$, obtenida de $P_0(t)$ asignando un nivel "alto" de duración τ para cada impulso positivo y un nivel "bajo" de duración τ para cada impulso negativo.

A este punto ya podemos observar una diferencia fundamental entre el MCP y la modulación Delta: en la modulación Delta no se transmite una codificación binaria de la señal $s(t)$ (como ocurre en MCP), sino una codificación binaria (de un solo bit) de la diferencia $\Delta(t)$ entre $s(t)$ y una señal que se aproxima $s(t)$.

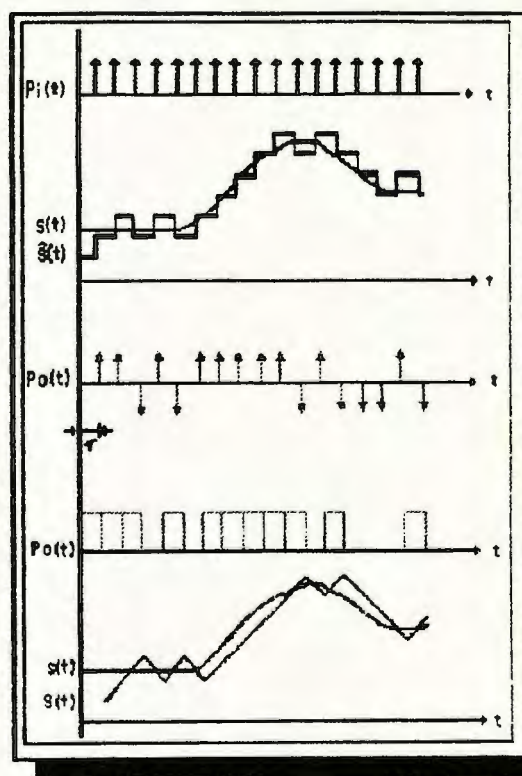


fig. 1.6b. Formas de onda.

Por lo tanto la modulación delta está representada por una serie de "0" y "1" lógicos. que codifican la derivada de una señal analógica: un "cero" indica disminución de la señal. mientras que un "uno" se asocia a un aumento.

El demodulador delta está constituido por un simple integrador seguido de un filtro de paso bajo. El integrador reconstruye una señal $s(t)$ a través de una sucesión de rampas. el filtro de paso bajo suprime los residuos de la frecuencia del generador de impulsos. Se obtiene así. a la salida del sistema transmisor. la señal $s(t)$ que. salvo diferencias debido a ruidos de distinta naturaleza. es idéntica a la señal moduladora $s(t)$.

Un modulador en delta básico (Fig.1.7) que consta de un comparador y un muestreador en trayectoria directa. y de un integrador amplificador en la trayectoria de retroalimentación.

Existe una señal analógica que se le llama $m(t)$ que se compara con una señal de retroalimentación $m'(t)$. La señal de error $e(t)=m'(t)-m(t)$ se aplica a un comparador. Si e es positiva. la salida del comparador será una señal constante de amplitud E . y si E es negativa la salida del comparador será $-E$. De este modo. la salida del comparador $m_c(t)$ se obtiene de

$$m_c(t)=Esgn[E(t)]$$

sen: es el signo (+ ó -) según la tendencia de la curva.

La salida del comparador se muestrea a un índice de f_s muestras/segundo, donde f_s es típicamente mucho mayor que el índice de Nyquist. El muestreador produce así un tren de pulsos $d(t)$ que consta de pulsos positivos cuando $m(t) > m'(t)$ y de pulsos negativos cuando $m(t) < m'(t)$. El tren de pulsos $d(t)$ es el tren de pulsos modulados en delta Fig.(1.7d). La señal modulada $d(t)$ se amplifica y se integra en la trayectoria de retroalimentación para generar a $m'(t)$ Fig.(1.7c), que trata de seguir a $m(t)$. Cada pulso en $d(t)$ da lugar a una función escalón (positiva o negativa, dependiendo de la polaridad del pulso) en $m'(t)$. Si, por ejemplo, $m(t) > m'(t)$, se generará un pulso positivo en $d(t)$, que da lugar a un escalón positivo en $m'(t)$ que trata de igualar $m'(t)$ con $m(t)$ en pequeños pasos en cada instante de muestreo. Las formas de onda $m(t)$ y $m'(t)$ aparecen en la Fig.(1.7c). Puede verse que $m'(t)$ es un tipo de aproximación de escalera de $m(t)$. Cuando $m'(t)$ se hace pasar por un filtro de pasabajos, se elimina lo abrupto de la escalera en $m'(t)$ y se obtiene una aproximación mas suave y mejor a $m(t)$. El demodulador del receptor consta de un amplificador integrador (idéntico al de la trayectoria de retroalimentación del modulador) seguido de un filtro pasabajos Fig.(1.7b).

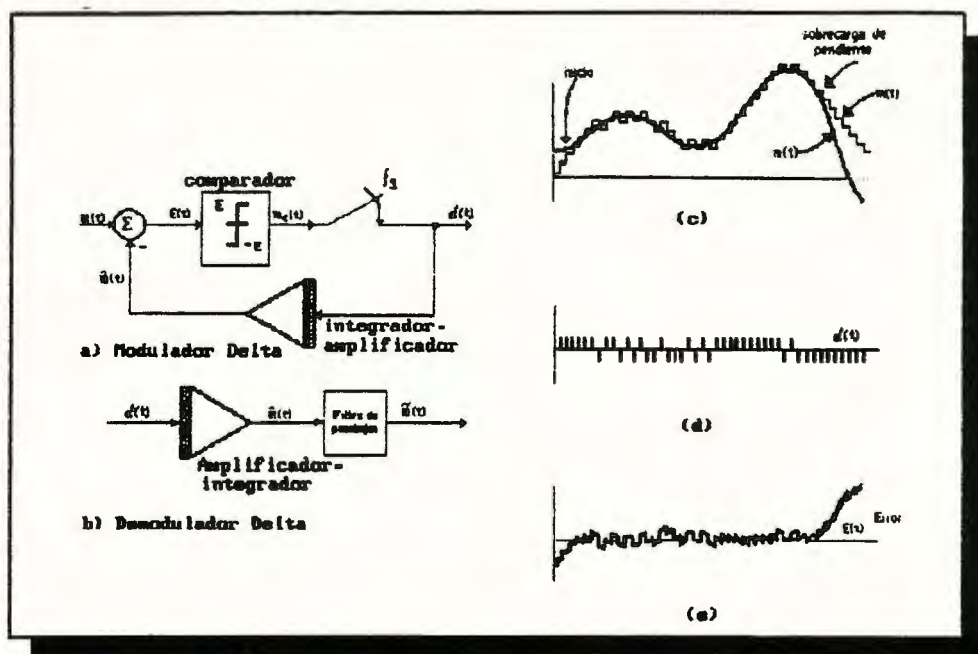


Fig. 1.7: Modulador en delta

2.3.2 Modulación Delta (MD), Modulación por Codificación de Pulsos (MCP), Modulación por Codificación de Pulsos Diferenciales (MCPD).

En la MCP, las muestras de la señal analógica se cuantifican en L niveles, y esta información se transmite mediante n pulsos por muestra ($n = \log_2 L$). Una pequeña reflexión muestra que en la MD, la señal modulada porta información no acerca de las muestras de la señal sino acerca de la diferencia entre muestras sucesivas.

Si la diferencia es positiva o negativa, se genera un pulso positivo o negativo (respectivamente) en la señal modulada $d(t)$. Básicamente por lo tanto, la MD porta la información acerca de la

derivada de $m(t)$ y, en consecuencia, resulta en nombre de modulación en delta.

Esto también se puede deducir del hecho de que la integración de la señal modulada da por resultado $m'(t)$, que es una aproximación de $m(t)$.

En la MCP, la información de cada muestra cuantificada se transmite mediante una palabra de códigos de n pulsos, mientras que en MD, la información de la diferencia entre muestras sucesivas se transmite mediante una palabra de código de un pulso. Podemos esperar entonces un gran error de cuantificación en MD.

Afortunadamente, esto no es cierto ya que en MD un factor de compensación es que el índice de muestreo es mucho más alto, típicamente de cuatro a ocho veces más alto que el de la MCP.

Una variante de la MD utiliza una palabra de código de k pulsos en lugar de solo uno para transmitir la información acerca de la diferencia entre muestras sucesivas. En este esquema el error $e(t)$ Fig.(1.7e) se cuantifica en 2^k niveles, y el valor cuantificado de $e(t)$ se transmite entonces mediante una palabra de código binario de k pulsos (igual que en MCP). Este esquema se conoce como *modulación en delta mediante MCP (MCPD)*. Se ha demostrado que para las señales de voz, el comportamiento de la MCPD es superior tanto al de la MD como al de la MCP. Desafortunadamente, el proceso de cuantificación y codificación hace a la MCPD tan compleja como la

MCP. La simplicidad de la MD así como la simplicidad en la multiplexión en MCP, se pierden ambas en la MCPD:

La MCPD se ha utilizado en las señales de telefonía visual digitalizada (teléfono visual) para conservar el ancho de banda. El índice de pulsos que requiere es solo de 6.312 Mbits/segundo en contraste con el índice de pulsos de cerca de 92 Mbits/segundo para las señales de video de TV digitalizadas (MCP).

2.3.3 Umbral de codificación y sobrecarga

Los efectos de umbral y de sobrecarga pueden verse claramente en la Fig.(1.7c). Las variaciones en $m(t)$ menores que el valor de escalón (umbral de codificación) se pierden en la MD. Al mismo tiempo, si $m(t)$ cambia con mucha rapidez (o sea, $m(t)$ es demasiado alta), $m'(t)$ no podrá seguir a $m(t)$, y se presenta la sobrecarga. Esta es la llamada *sobrecarga de pendiente*. Este efecto aparece cuando la señal moduladora varía su amplitud entre dos pulsos de reloj sucesivos en una cantidad superior a la amplitud de la rampa. Este tipo de distorsión no es causada por la amplitud de la señal moduladora, sino por su pendiente. se puede aumentar la frecuencia del reloj (pero incurriendo en el mismo inconveniente visto a propósito del ruido de cuantificación), o bien se puede aumentar la amplitud de la rampa, pero esto conllevaría a un aumento del

ruido de cuantificación. Este es uno de los factores limitantes básicos en el funcionamiento de la MD. Debemos esperar sobrecarga de pendiente más que sobrecarga de amplitud en la MD, ya que esta modulación básicamente porta la información acerca de $m(t)$.

La sobrecarga de pendiente se presenta cuando $m'(t)$ no puede seguir a $m(t)$. Durante el intervalo de muestreo T_s , $m'(t)$ es capaz de cambiar en σ , donde σ es la altura del escalón. Por lo tanto la máxima pendiente que $m'(t)$ puede seguir es σ/T_s , o bien en σf_s , donde f_s es la frecuencia de muestreo. En consecuencia no ocurrirá sobrecarga si

$$|m(t)| < \sigma f_s$$

Considerando el caso de la modulación del tono:

$$m(t) = A \cos \omega t \quad \text{Donde } A \text{ es la amplitud de la señal } m(t)$$

La condición de no sobrecarga es

$$|m(t)|_{\max} = \omega A < \sigma f_s$$

Por lo tanto la máxima amplitud A_{\max} de esta señal que puede tolerarse sin sobrecarga se obtiene de

$$A_{\max} = \sigma \frac{f_s}{\omega}$$

La amplitud de sobrecarga de la señal moduladora es inversamente proporcional a la frecuencia ω . Para frecuencias moduladoras más altas, la sobrecarga se presenta en las amplitudes menores. Para las señales de voz, que contienen todas las componentes de frecuencia hasta de 4 kHz, calculando A_{\max} utilizando $\omega = 2\pi \times 4000$ dará un valor muy conservador de A_{\max} . De Jager ha demostrado que A_{\max} para las señales de voz se puede calcular utilizando $\omega_r \approx 2\pi \times 800$

$$[A_{\max}]_{\text{voz}} \approx \sigma \frac{f_s}{\omega_r}$$

ω_r es la frecuencia inversamente proporcional a la amplitud de sobrecarga de la señal $m(t)$

Así, la máxima amplitud de una señal de voz, A_{\max} , que se puede usar sin ocasionar sobrecarga de pendiente en MD, es la misma amplitud máxima de una señal senoidal de frecuencia de referencia f_r ($f_r \approx 800$) que se puede utilizar sin causar sobrecarga de pendiente en el mismo sistema.

Afortunadamente, el espectro de la voz (así como la señal de video de TV) también decae con la frecuencia y sigue muy cerca la característica de sobrecarga (curva c, Fig.1.8). Por esta razón la MD resulta muy adecuada para señales de voz (y de TV).

En realidad, el espectro de la señal de voz (curva b) decrece según $1/\omega$ hasta 2000Hz, y más allá de esta frecuencia, decrece de acuerdo a $1/\omega^2$.

Si hemos usado una doble integración en el circuito de retroalimentación en lugar de una integración simple, $A_{\text{máx}}$ debe ser proporcional a $1/\omega^2$.

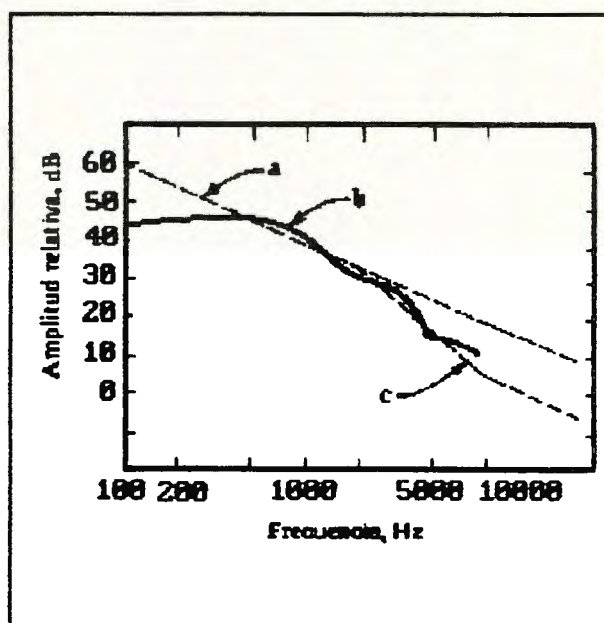


Fig.1.8:Espectro de señal de voz.

En consecuencia, se logra un mejor balance entre el espectro de voz y las características de sobrecarga utilizando una integración simple hasta 2000 Hz y una integración doble más allá de los 2000 Hz.

2.3.3 Integración doble

El efecto de la integración doble puede verse también desde otros puntos de vista. Cuando un pulso angosto se integra una vez, da lugar a una función escalón, y $m'(t)$ en el caso de la integración simple se forma a partir de una función escalón brusca. Si un pulso angosto se integra dos veces, la salida es una rampa, y se espera que en este caso $m'(t)$ sea una función más suave Fig.(1.9b) que siga a $m(t)$ más de cerca. La integración doble la proporcionan dos circuitos RC en cascada Fig.1.9, donde $R_2 C_2 \gg R_1 C_1$ de manera que la segunda sección de RC no sobrecargue a la primera. Ajustando los valores $R_2 C_2$ y $R_1 C_1$ de manera aproximada, la característica de sobrecarga se hace tan cercanamente igual al espectro de voz, como se muestra en la Fig.1.8(curva c). Aquí $1/R_1 C_1$ se elige como $200\pi(f_1=100 \text{ Hz})$, y $1/R_2 C_2$ se elige como $4000\pi(f_2=2000 \text{ Hz})$. Este circuito proporciona integración simple hasta 2000 Hz e integración doble más allá de los 2000 Hz para obtener una aproximación más cercana al espectro de voz Fig.1.8.

Desafortunadamente, la integración doble ocasiona un problema de cacería. En el caso de la integración simple, cada pulso sucesivo ocasiona un cambio en el escalón de σ en $m'(t)$. Por otra parte en la integración doble, cada pulso sucesivo causa un cambio fijo en la pendiente de $m'(t)$. Cuando $m(t)$ va en aumento, la salida del modulador $d(t)$ por lo general es una sucesión de pulsos positivos. Cada pulso aumenta la pendiente de $m'(t)$.

Ahora cuando $m(t)$ comienza a decrecer, $m'(t)$ podría tener en ese momento una pendiente grande, y podría transcurrir un tiempo largo antes que la pendiente decrezca lo suficiente como para enganchar con $m(t)$. Esto es lo que causa un disparo en $m'(t)$ en el punto en que $m(t)$ tiene máximos (Fig.1.9). El mismo proceso se repite cuando $m(t)$ tiene mínimos, con un disparo consecuente en $m'(t)$. Este tipo de cacería puede causar finalmente cierto tipo de oscilación y crear inestabilidad.

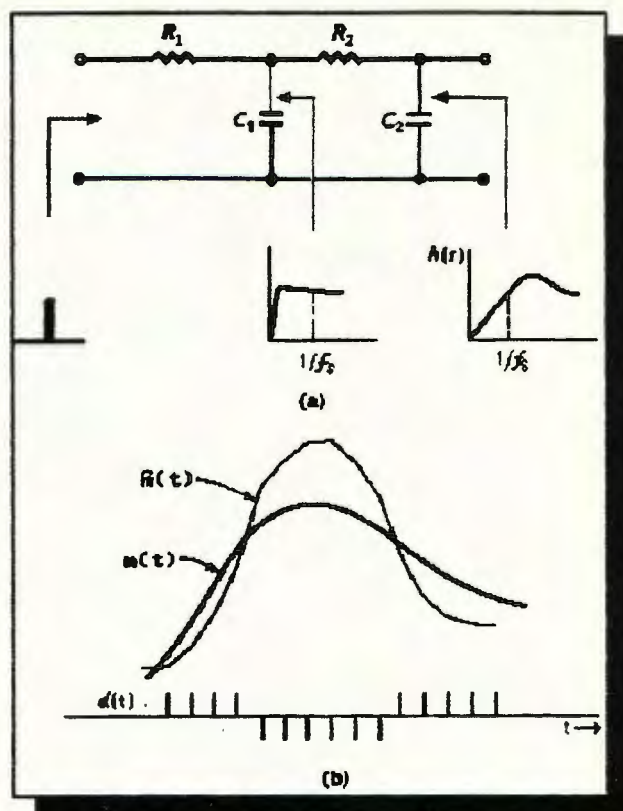


Fig. 1.9: Modulación en delta con integración doble

La cacería descrita anteriormente aparece debido a que el circuito no es capaz de reconocer con suficiente rapidez los cambios en $m'(t)$ con el fin de autoajustarse. Este defecto se podría corregir si fuera posible predecir el valor futuro de $m'(t)$ y compararlo con $m(t)$ para generar pulsos en $d(t)$.

El circuito de la (Fig.2.0a) proporciona integración doble así como predicción. Es posible demostrar que $E_2(t)$ aproximadamente representa el valor futuro de $E_1(t)$, esto es,

$$E_2(t) \approx E_1(t+\tau)$$

Para poder demostrar esto se observa que

$$E_1(t+\tau) \approx E_1(t) + \tau \frac{dE_1}{dt}$$

Se hace $r = \tau / C_2$ y tenemos

$$E_1(t+\tau) \approx E_1(t) + r i = E_2(t)$$

$$= E_1(t) + \tau \frac{i}{C_2}$$

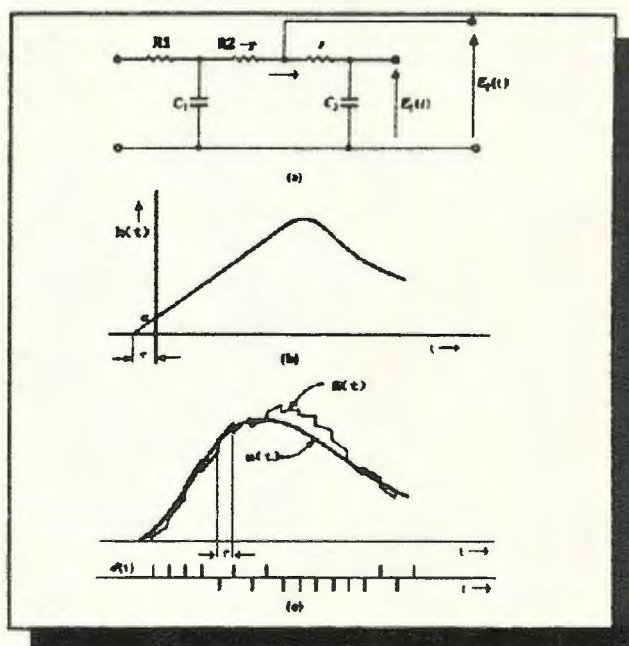


Fig. 2.0 Integrador doble con predicción

El tiempo de predicción τ se elige a ser un intervalo de muestreo. La forma de onda de $E_2(t)$ es aproximadamente la misma de $E_1(t)$ pero corrida hacia la izquierda τ segundos. La respuesta de impulso $h(t)$ de este circuito es entonces la misma de la (Fig.1.9) pero desplazada hacia la izquierda τ segundos (Fig.2.0b).

Así, la respuesta de impulso se inicia con un valor α en $t=0$, y su pendiente finalmente decrecerá hasta cero después de algunos intervalos de τ segundos.

CAPITULO III

3. ELABORACIÓN DEL SOFTWARE DIDÁCTICO

Para la elaboración del Software Didáctico se han utilizado variados tipos de lenguajes de programación.

El lenguaje Turbo Pascal es uno de ellos, otro lenguaje que se utilizo es el Turbo C. se notó que es muy fácil hacer enlaces de un lenguaje a otro por medio de sus programas compilados.

3.1 Lenguaje TURBO PASCAL

3.1.1 Reseña histórica del lenguaje PASCAL.

El lenguaje Pascal utilizado en el trabajo de graduación, se eligió por las siguientes razones:

1. Por la compatibilidad con otros lenguajes de programación.
2. Es el lenguaje de programación que ya se conoce por lo tanto una herramienta con la que ya se cuenta.

Este lenguaje nació de la necesidad de hacer una comunicación completa, inequívoca y fácil de aprender e implementar en una computadora.

Analizando la historia de los diferentes lenguajes de programación puede observarse que uno de los primeros en aparecer fue el FORTRAN (FORmula TRANslator). El FORTRAN fue uno de los lenguajes de uso más extendido en la rama de los cálculos científicos. Debido a su temprana aparición, actualmente se ha convertido en una colección de funciones que resultan útiles, pero incómodas para aprender y manejar en una computadora.

Se hicieron intentos para definir un lenguaje sencillo directamente inspirado en el FORTRAN y que fuera fácil de aprender y pudiera usarse en forma conversacional. El resultado de estos intentos fue el BASIC (Beginner's All-purpose Symbolic Instruction Code). El lenguaje BASIC es fácil de implementar en cualquier computadora y requiere solamente un pequeño esfuerzo de memoria. Estas dos cualidades (fácil implementación y sencillez) han propiciado que el BASIC fuera lenguaje de programación más usado en microcomputadoras, aunque, por otro lado, presenta grandes limitaciones debido a sus reglas de uso ("sintaxis") que, a menudo, le hacen inadecuado para la realización de programas complejos.

Otro lenguaje, ALGOL (ALGOritms Language), surgió como resultado del intento de crear otro lenguaje de programación distinto del FORTRAN que fuera consistente y apropiado para programar complicados algoritmos. El ALGOL alcanzó gran popularidad en ambientes educativos, aunque nunca llegó a ser ampliamente utilizado. Aunque este lenguaje es una herramienta eficaz para la

programación de complicados algoritmos, es difícil de aprender y de implementar. El lenguaje pascal se inspiró en los lenguajes ALGOL y PL/I, y representa un intento de crear un medio fácil de aprender, apropiado para la realización de algoritmos complejos y la definición de estructuras de datos. Fue creado por Niklaus Wirth, perteneciente al ETH, Instituto Técnico de Zurich, en los años 1970-1971, después de su regreso de la Universidad de Stanford. Rápidamente ganó aceptación en los círculos educativos como una buena herramienta para aprender a programar. Se presenta, además, como un lenguaje relativamente simple y muy coherente, por lo que el compilador necesario para su implementación requiere una pequeña porción de memoria. El nombre de este lenguaje rinde tributo al matemático francés Blaise Pascal, quien, en 1690, a la edad de dieciocho años, inventó la primera máquina de calcular.

3.1.2 El Turbo PASCAL y otras modalidades del lenguaje PASCAL.

El lenguaje Pascal fue definido originalmente por Niklaus Wirth y Kathleen Jensen en su libro *Pascal User Manual and Report*, publicado por primera vez en 1974. Más tarde, esta definición fue formalizada por International Standards Organizations (ISO) como Nivel 0 ISO Standards Pascal (ISO dp 7185) en 1980. La forma de pascal adoptada por ISO fue aceptada (con muy pequeñas variaciones) en Estados Unidos por American International Standards Institute (ANSI) y el Institute of Electrical and Electronics Engineers (IEEE) como el lenguaje Pascal patrón en 1983 (ANSI/IEEE770X3:97-1983).

La definición del lenguaje Pascal hecha por ISO se puede decir que es idéntica a la elaborada por Kirth originalmente, pero mucho más precisa.

ISO Standard Pascal se llamará Pascal estándar por simplicidad. Actualmente hay, sin embargo, gran número de modificaciones y ampliaciones hechas sobre el Pascal estándar.

A medida que se extendía el lenguaje Pascal se fue modificando la definición actual, se añadieron algunos elementos nuevos y otros fueron interpretados de distintas formas cuando aparecieron ambigüedades.

Como ocurre con cualquier lenguaje de programación, el Pascal depende, en última instancia, de su implementación concreta. En teoría, aprender Pascal supone no solamente aprender pascal estándar, sino, también aprender los elementos y las diferencias propias de la versión que se utilice. Afortunadamente, en la práctica, todas las versiones responden al lenguaje Pascal estándar con algunos pocos elementos nuevos y cambios poco significativos. El lenguaje Pascal fue diseñado, originalmente, para computadoras que trabajan por "lotes", donde el programa se introdujera en un conjunto de tarjetas, o de una cinta magnética; pero, a medida que el uso de este lenguaje se hizo más popular, comenzó a instalarse en sistemas de tiempo compartido y microcomputadoras, en los que el usuario tiene acceso directo a una terminal.

Debido precisamente a la continua interacción entre la terminal y el usuario, posible en este tipo de computadoras, fue necesaria la implementación de nuevos elementos y surgió la versión UCSD Pascal. Esta versión fue desarrollada en la Universidad de California de San Diego (UCSD), y fue instalada en muchas computadoras pequeñas, incluido el Apple II, por el año de 1978. Esta nueva versión contenía varias innovaciones respecto del Pascal estándar, incluida una técnica que permitía fácilmente la realización de gráficos en la pantalla y que recibió el nombre de Turbo Graphics.

En 1983, Borland International ultimó el "Turbo Pascal", un compilador muy sofisticado para las computadoras CP/M, IBM PC-DOS y MS-DOS. La versión Turbo Pascal es un compilador potente, barato y rápido que incluye, además, detección de errores y procesador de textos en el programa editor. Todos estos elementos configuran en la programación en Turbo Pascal.

3.1.3 Escritura de un programa en PASCAL.

Una vez que la solución de un problema cualquiera ha sido expresada en forma de algoritmo, éste debe ser convertido en un programa en Pascal. Una vez que se ha escrito el programa, el compilador debe traducirlo y proceder a su ejecución. Algunos otros programas como el editor y el sistema de archivos están en continua interacción con la computadora para facilitar estos procesos.

3.1.4 La escritura del lenguaje PASCAL.

El Pascal es un lenguaje de alto nivel y permite al programador escribir sus instrucciones en un idioma similar al inglés pero muy restringido. Para evitar cualquier ambigüedad en la traducción que del programa hace el compilador a código binario, es necesario que la sintaxis imponga unas reglas muy estrictas.

La programación requiere ingenio e inteligencia, pero también necesita de una férrea disciplina. Cualquier introducción u orden escrita en Pascal debe seguir estrictamente las reglas de sintaxis, porque cualquier instrucción que viole alguna de estas reglas establecidas inutiliza el programa. No hay ninguna excepción. Es por tanto esencial entender y respetar la sintaxis para llevar a buen fin la escritura de cualquier programa. La omisión de una simple coma hace que el programa no funcione. por más que se diga al respecto nunca se resaltarán lo suficiente la necesidad de seguir una estricta disciplina a la hora de programar.

Las reglas del lenguaje Pascal pueden describirse de muchas formas: usando palabras, la notación BNF (Backus-Naur Form), o diagramas sintácticos.

3.1.5 Formatos de un programa en PASCAL.

El lenguaje Pascal fue creado para motivar la programación por bloques, es decir: cada paso o grupo de pasos que guarden una fuerte relación lógica dentro de un algoritmo pueden agruparse en un bloque o módulo. Cada uno de estos módulos recibe el nombre de bloque, función o procedimiento, dependiendo de la forma en que se usen.

La sintaxis del lenguaje Pascal requiere que todas las definiciones y declaraciones aparezcan al principio de l programa, por lo que la organización general puede representarse como en la fig. 3.1.

Como puede apreciarse en la fig. 3.1 todas las definiciones y declaraciones aparecen al principio del programa, seguidas por el bloque principal, que está determinado claramente por las palabras BEGIN y END.

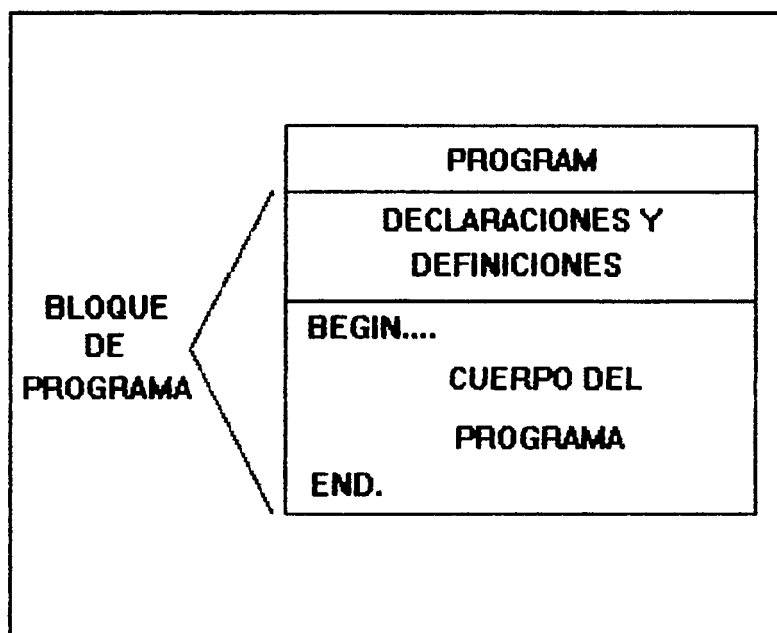


FIGURA ###.

Ahora se va a considerar la organización de cada uno de los módulos reflejados en la fig. 3.1 . Un programa en Pascal puede verse en detalle en la fig. 3.2

3.1.6 Declaraciones.

Los diferentes tipos de declaraciones que existen en Pascal deben aparecer, exactamente, en orden en que se han colocado en la fig. 3.2. Primero las etiquetas, después las constantes, etc. Sin embargo, ninguna de ellas es obligatoria. Por ejemplo, en el siguiente programa:

```
PROGRAM SALUDO (OUTPUT);  
    { ESTE ES UN PROGRAMA EN PASCAL }  
    BEGIN  
        WRITELN('HOLA')  
    END.
```

no existen declaraciones. Únicamente se incluyen la cabecera y el cuerpo principal de la parte ejecutable. Hay que recordar siempre que los comentarios son ignorados por el compilador. Normalmente, en cuanto el programa tenga más que unas pocas líneas, es necesario usar variables y, por tanto, es obligada su declaración explícita.

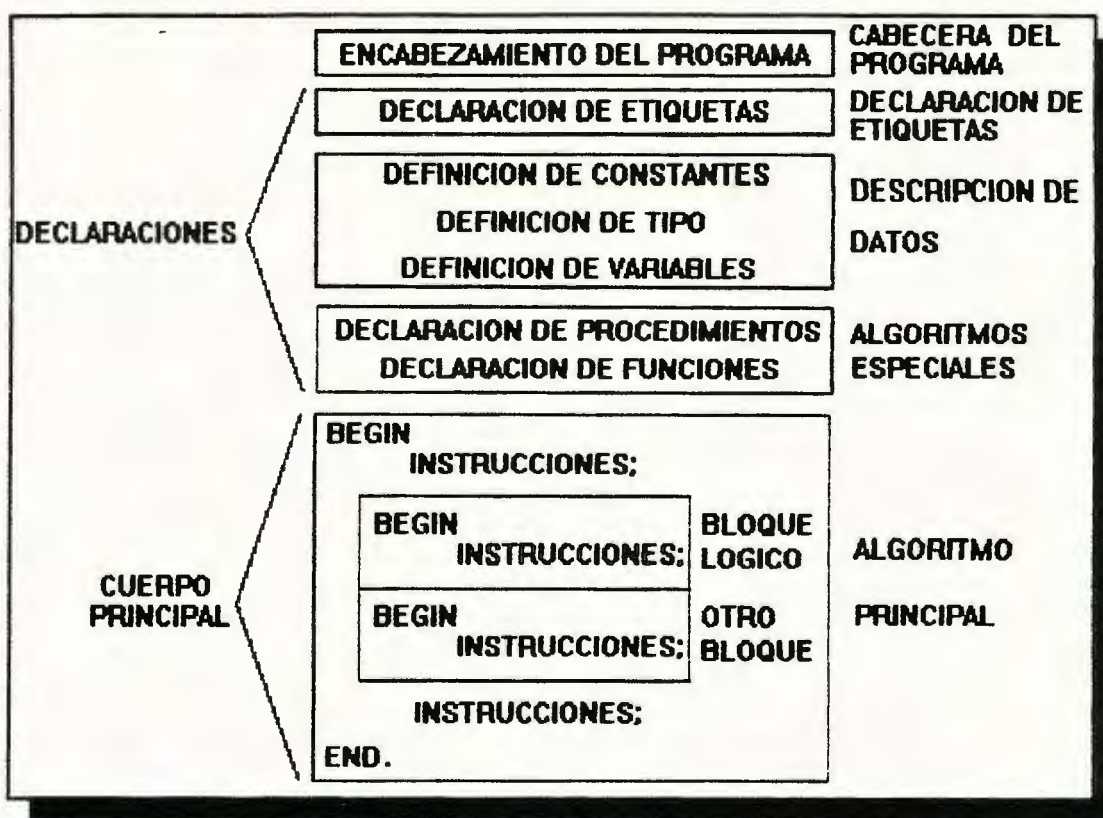


FIGURA 3.2

Refiriéndose al esquema de la fig. 3.2. hay que decir que las etiquetas se usan algunas veces y que las funciones y los procedimientos se usan cuando el programa tiene una longitud grande.

Las únicas declaraciones necesarias en programas cortos son: CONST, TYPE y VAR.

3.1.7 Resumen de la organización de un programa.

En resumen, cada programa en Pascal debe tener, al menos, un encabezamiento o cabecera y una instrucción. Además, el programa

puede contener varias declaraciones o definiciones siguiendo a la cabecera (en el orden adecuado), y un número cualquiera de instrucciones. Los comentarios, los espacios en blanco y los sangrados que se hagan al escribir el programa pueden colocarse en cualquier lugar para facilitar su lectura.

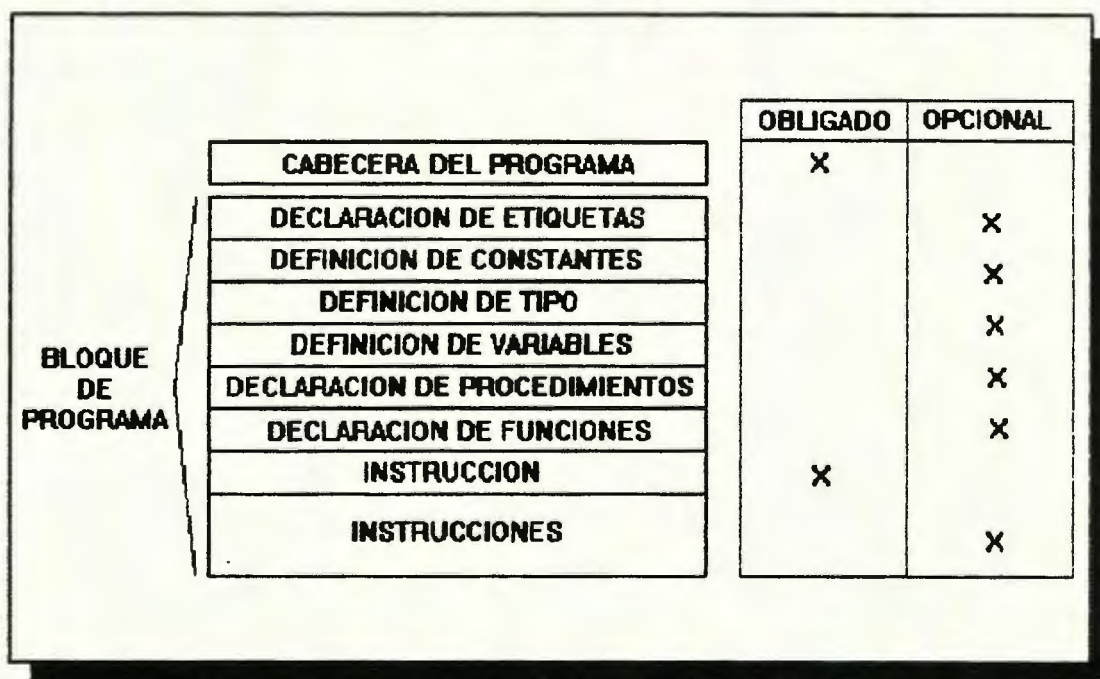


FIGURA 3.3

3.1.8 Organización formal de un programa.

En relación con esto, la organización del programa será representado en la fig. 3.3.

Nótese que en la definición formal de un programa en Pascal el bloque del programa se refiere a todo lo que sigue en la cabecera, incluido el conjunto de declaraciones.

3.2 Lenguaje Turbo C

3.2.1 Introducción al C.

C es un lenguaje de programación estructurado de propósito general. Sus instrucciones constan de términos que se parecen a expresiones algebraicas, además de ciertas *palabras clave* inglesas como if, else, for, do, while. En este sentido, C recuerda a otros lenguajes de programación estructurados de alto nivel como FORTRAN-77 y Pascal. C tiene también algunas características adicionales que permiten su uso a un nivel más bajo, cubriendo así el vacío entre el lenguaje máquina y los lenguajes de alto nivel más convencionales. Esta flexibilidad permite el uso de C en la programación de sistemas (diseño de sistemas operativos) así como en la programación de aplicaciones (por ejemplo redactar un programa que resuelva un complicado sistema de ecuaciones matemáticas, o un programa que escriba las facturas de los clientes).

C se caracteriza por hacer posible la redacción de programas fuente

muy concisos, debido en parte al gran número de operadores que incluye el lenguaje. Tiene un conjunto de instrucciones relativamente pequeño, aunque las implementaciones actuales incluyen numerosas *funciones de biblioteca* que mejoran las instrucciones básicas. Es más, el lenguaje permite a los usuarios escribir funciones de biblioteca adicionales para su uso propio. De esta forma, las características y capacidades del lenguaje se pueden ampliar fácilmente por el usuario.

Hay compiladores de C disponibles para computadoras de todos los tamaños y los intérpretes de C se están haciendo cada vez más comunes. Los compiladores son frecuentemente compactos, y generan programas objeto que son pequeños y muy eficientes en comparación con los programas generados a partir de otros lenguajes de alto nivel. Los intérpretes son menos eficientes, aunque son de uso más cómodo en el desarrollo de nuevos programas. Muchos programadores comienzan utilizando un intérprete, y una vez que han depurado el programa (eliminado los errores del programa) utilizan un compilador.

Otra característica importante de C es que los programas son muy portables, más que los escritos en otros lenguajes de alto nivel. La razón de esto es que C deja en manos de las funciones de biblioteca la mayoría de las características dependientes de la computadora. Toda versión de C se acompaña de su propio conjunto de funciones de biblioteca, que están escritas para las características particulares de la computadora en la que se instale. Estas

funciones de biblioteca están relativamente normalizadas y se accede a cada función de biblioteca de igual forma en todas las versiones de C. De esta forma, la mayoría de los programas en C se pueden compilar y ejecutar en muchas computadoras diferentes con muy pocas o ninguna modificación.

3.2.2 Historia del C.

C fue desarrollado originalmente en los años setenta por Dennis Ritchie en Bell Telephone Laboratories. Inc.(ahora AT&T Bell Laboratories). Es el resultado de dos lenguajes anteriores, el BCPL y el B. que se desarrollaron también en los laboratorios Bell. C estuvo al uso en los laboratorios Bell hasta 1978. cuando Brian Kernighan y Ritchie publicaron una descripción definitiva del lenguaje. La definición de Kernighan y Ritchie se denomina frecuentemente "K&RC".

Tras la publicación de K&R. los profesionales de las computadoras. impresionados por las muchas características deseables del C. comenzaron a promover el uso del lenguaje. Por la mitad de los ochenta, la popularidad del C se había extendido por todas partes. Se habían escrito numerosos compiladores e intérpretes de C para computadoras de todos los tamaños y se habían escrito originalmente

en otros lenguajes se reescribieron en C para tomar partido de su eficiencia y portabilidad.

La mayoría de las implementaciones comerciales de C difieren en algo de la definición original de K&R. Esto ha creado algunas pequeñas incompatibilidades entre las diferentes implementaciones del lenguaje, disminuyendo la portabilidad que éste intentaba proporcionar.

Consecuentemente, el Instituto Americano de Estándares (ANSI) comenzó a trabajar en una definición normalizada del lenguaje C (comité ANSI X3J11). Es de esperar que la mayoría de los compiladores e intérpretes comerciales adopten el estándar ANSI una vez que se haya completado (muchos siguen ahora las recomendaciones parciales ya fijadas). Algunos compiladores también pueden proporcionar características suyas propias.

3.2.3 Estructura de un programa en C

Todo programa en C consta de una o más *funciones*, una de las cuales se llama `main`. El programa siempre comenzará por la ejecución de la función `main`. Las definiciones de las funciones adicionales pueden preceder o seguir a `main`.

Cada función debe contener:

1. Una *cabecera* de la función, que consta del nombre de la función, seguido de una lista opcional de *argumentos* encerrados con paréntesis.
2. Una lista de *declaración* de argumentos, si se incluyen estos en la cabecera.
3. Una *sentencia compuesta*, que contiene el resto de la función.

Los argumentos son símbolos que representan información que se le pasa a la función desde otra parte del programa. (También se llaman *parámetros* a los *argumentos*.)

Cada sentencia compuesta se encierra con un par de llaves, {y}. Las llaves pueden contener combinaciones de sentencias elementales (denominadas sentencias de expresión) y otras sentencias compuestas. Así las sentencias compuestas pueden estar anidadas, una dentro de otra. Cada Sentencia de expresión debe acabar en punto y coma (;).

Los comentarios pueden aparecer en cualquier parte del programa, mientras estén situados entre los delimitadores /*y*/ (por ejemplo: /*esto es un comentario*/). Los comentarios son útiles para identificar los elementos principales de un programa o para la

explicación de la lógica subyacente de éstos.

Ejemplo. Area de un Circulo. He aquí un programa en C elemental que lee el radio de un círculo, calcular el área y escribir el resultado calculado.

```
#include <stdio.h>                                /*ACCESO A ARCHIVO DE BIBLIOTECA*/
/*programa para calcular el área de un círculo*/ /*TITULO (COMENTARIO)*/
main()                                              /*CABECERA DE FUNCION*/
{
    float radio, area;                            /*DECLARACION DE VARIABLES*/

    printf("Radio=?");                            /*SENTENCIA DE SALIDA*/
    scanf("%f",&radio);                          /*SENTENCIA DE ENTRADA*/
    area=3.14159 *radio*radio;                   /*SENTENCIA DE ASIGNACION*/
    printf("Area=%f",area);                      /*SENTENCIA DE SALIDA*/
}
```

Se han añadido los comentarios al final de cada línea para hacer un mayor énfasis en la organización del programa en conjunto. Normalmente un programa en C no se parecerá al anterior, sino más bien al que aparece a continuación.

```
#include <stdio.h>
/*programa para calcular el área de un círculo*/
main()
{
    float radio, area;

    printf("Radio=?");
    scanf("%f",&radio);
    area=3.14159 *radio*radio;
```



```
printf("Area=%f",area);
```

```
}
```

Se deben subrayar las siguientes características del último programa.

1. El programa está escrito en minúsculas sin acentuar (salvo los comentarios). Se pueden usar las mayúsculas o minúsculas, aunque es costumbre escribir en minúsculas las instrucciones ordinarias. La mayoría de los comentarios también se escriben en minúsculas, aunque a veces se hace en mayúsculas para poner especial énfasis, o distinguir ciertos comentarios de instrucciones (las letras mayúsculas y minúsculas no son equivalentes en C).
2. La primera línea contiene una referencia a un archivo especial(`stdio.h`) que contiene información que se debe incluir (`include`) en el programa cuando se compila. La inclusión requerida de esta información será manejada automáticamente por el compilador.
3. La segunda línea es un comentario que describe el propósito del programa.
4. La tercera línea es la cabecera de la función `main`. Los paréntesis vacíos que siguen al nombre de la función indican

que ésta no incluye argumentos.

5. Las cinco líneas restantes del programa están adentradas (con tabuladores a la izquierda) y encerradas por un par de llaves. Estas cinco líneas integran la sentencia compuesta dentro de main.
6. La primera línea adentrada es una declaración de variables. Establecer los nombres simbólicos de radio y area como variables de coma flotante.
7. Las cuatro líneas adentradas son un tipo particular de sentencia de expresión. La segunda línea adentrada (printf) genera una solicitud de información (el valor del radio). Este valor se introduce en la computadora al ejecutarse la tercera línea adentrada (scanf).
8. La cuarta línea adentrada es un tipo particular de sentencia de expresión llamada sentencia de asignación. Esta sentencia hace el cálculo del área a partir del valor del radio dado. Dentro de esta sentencia los asteriscos (*) representan signos de multiplicación.
9. La última línea adentrada (printf) presenta el valor calculado del área. El valor numérico será precedido por un breve mensaje ("Area").

10. Nótese que cada sentencia de expresión dentro de la sentencia compuesta acaba con un punto y aparte. Esto es necesario en toda sentencia de expresión.
11. Finalmente, señalar el abundante espaciado y tabuladores, creando espacio en blanco dentro del programa. Las líneas separan partes diferentes del programa en componentes lógicamente identificables. La adentración indica relaciones de subordinación entre varias instrucciones. Estos elementos no son esenciales gramaticalmente, pero su presencia es fundamental en la práctica de una buena programación.

La ejecución del programa conlleva un dialogo interactivo que se muestra a continuación. La respuesta del usuario está subrayada para mayor claridad.

Radio= ? 3

Area= 28.274309

3.2.4 Características deseables de un programa

Antes de concluir este capítulo, examinemos brevemente algunas características importantes de los programas bien escritos. Estas características se pueden aplicar a programas escritos no solo en C, sino en cualquier lenguaje de programación. Pueden aportarnos

una serie de normas generales muy útiles para cuando comencemos a escribir nuestros propios programas en C.

1. *Integridad.* Esto se refiere a la corrección de los cálculos. Está claro que toda posible ampliación del programa no tendrá sentido si los cálculos no se realizan de forma correcta. Así la integridad de los cálculos es absolutamente necesaria en cualquier programa de computadora.
2. *Claridad.* Esto hace referencia a la facilidad de lectura del programa en conjunto, con particular énfasis en la lógica subyacente. Si un programa está escrito de forma clara, será posible para otro programador seguir la lógica del programa sin esfuerzo innecesario. También hará posible al autor original seguir su propio programa después de haberlo dejado durante un período largo de tiempo. Uno de los objetivos al diseñar C, fue el desarrollo de programas claros y de fácil lectura a través de un enfoque de la programación ordenado y disciplinado.
3. *Sencillez.* La claridad y corrección de un programa se suelen ver favorecidas con hacer las cosas de forma tan sencilla como sea posible, consistente con los objetivos del programa en su conjunto. De hecho puede ser deseable sacrificar cierta cantidad de eficiencia computacional con vistas a no complicar la estructura del programa.

4. *Eficiencia.* Esto está relacionado con la velocidad de ejecución y la utilización eficiente de la memoria. Este es uno de los objetivos importantes, aunque no se debe conseguir a expensas de la pérdida de claridad o la sencillez. Muchos programas complicados conducen a un enfrentamiento de estas características. En estas situaciones es necesario recurrir a la experiencia y al sentido común.
5. *Modularidad.* Muchos programas se pueden dividir en pequeñas subtarefas. Es una buena práctica de programación implementar cada una de estas subtarefas como un módulo separado del programa. En C estos módulos son las funciones. El diseño modular de los programas aumenta la corrección y claridad de éstos y facilita los posibles cambios futuros del programa.
6. *Generalidad.* Normalmente queremos que un programa sea lo más general posible, lo razonable. Podemos hacer un programa que lea los valores de ciertos parámetros en lugar de dejarlos fijos. Como norma general se puede conseguir con muy poco esfuerzo adicional una cantidad considerable de muestra, pues estos valores están oscilando entre (250 y 2500Hz),

MANUAL DE USO DEL SOFTWARE

1. COMO INSTALAR EL PROGRAMA:

1.1 Como primer paso, introduzca el disco intalador en el drive A:

1.2 Escriba la palabra "INSTALAR" y presione Enter, espere un momento, la clave es "EMULADOR"

1.3 Espere a que el programa sea instalado en su totalidad.

1.4 Ahora está dentro del programa "DELTA", presione Enter para continuar.

2. SECCIÓN MENU DE AYUDAS:

A continuación de la pantalla de presentación aparecerá el menú de ayudas generales, estas dan una introducción del contenido del programa.

2.1 Aqui se despliega una pantalla que nos muestra 5 opciones de ayuda, una Acerca de... , Salir al DOS y otra para correr el programa. puede hacer uso de las teclas de cursor o también del mouse haciendo doble "click" en cada una de las opciones.

2.2 Se le recomienda leer cada una de la ayudas, ya que en ellas están los pasos de como utilizar cada una de las opciones del menú principal, luego de haber leído estas corra el programa "Run Program DELTA Now".

3. MENU PRINCIPAL:

3.1 Sección Teoría: Esta sección se compone de 18 páginas que contienen toda la información básica acerca de la Modulación Delta, puede utilizar el "Mouse" o las teclas (+ -) del "teclado numérico" para cambiar las páginas, "+" para adelantar página y "-" para retroceder página, si desea salir en cualquier momento presione la tecla "Esc".

Con el "Mouse" use la tecla de la izquierda para adelantar página y la tecla izquierda para retroceder página, si desea salir en cualquier momento haga click en las dos teclas del mouse al mismo tiempo.

3.2 Sección Simulación: La parte de simulaciones muestra una serie de pantallas que contienen un diagrama de bloques de un modulador delta con sus respectivas respuestas en señal y explicaciones, después de leer cada una de estas y tratar de entenderlas puede pasar a la siguiente presionando "Enter". y al final presionando "Esc" le dará acceso a el menú de simulaciones.

3.2.1 Menú de simulaciones: Puede acceder cualquiera de las cinco señales que se mencionan en el menú, escoja una presionando las teclas de función que se indican, si no está muy claro, con la tecla "Esc" puede ir al menú de ayuda de Simulaciones, el diagrama de bloques y las gráficas le explican cuál es la procedencia de cada señal. Para seguir

haciendo uso de la simulación salga hacia el menú principal con la opción #6, regresará al menú principal y podrá nuevamente acceder al menú de simulaciones.

3.3 *Práctica:* La opción #3 del menú principal es el de la práctica, podrá realizar una práctica que le orientará a un mejor uso del programa, otra práctica que trata acerca de EL Ruido Granular y una última práctica que trata de La Sobrecarga de Pendiente.

3.1 *Menú de Prácticas:* Este menú, se puede ver que tiene solamente tres opciones, que son las que comprenden la práctica. Con las teclas de función indicadas seleccione el tipo de señal que desea utilizar y luego introduzca sus valores. Esta parte del Software la complementaremos con la guía de prácticas que se dará posteriormente.

3.4 *Cuestionario:* Para obtener un buen resultado en esta parte del programa se recomienda que se haga una lectura previa a la sección de la teoría.

3.4.1 *Operación del Cuestionario:* Cada pregunta es de selección múltiple, usted entra con puntaje de Diez, por cada respuesta equivocada que de se le restará un punto, hasta que la respuesta sea correcta podrá pasar a la siguiente. Su nota mínima para aprobar es de Seis punto Cero (6.0).

ANEXO 1.1

-Guías prácticas del programa Delta-



Departamento: ELECTRÓNICA **Nivel:** TECNOLÓGICO
Materia: APLICACIONES EN COMUNICACIÓN
Título: SOFTWARE DIDÁCTICO DE MODULACIÓN DELTA
Profesor:
Instructor:
Lugar de Ejecución: Laboratorio de TELECOMUNICACIONES
Tiempo de ejecución: 4 hrs.

I. Objetivos Específicos

El estudiante será capaz de:

- Describir el comportamiento del sistema Delta para los distintos tipos de señal.
- Enunciar los conceptos básicos de modulación Delta por medio de la teoría presentada dentro del software.

II. Introducción Teórica

La Modulación Delta es una técnica que permite codificar una señal analógica en cifras binarias (bit). Este proceso se realiza mediante las dos operaciones clásicas de muestreo y codificación de la señal a transmitir, por ello la Modulación Delta se puede considerar un sistema PCM o MIC todos los efectos aunque si con el término PCM se identifica una técnica de codificación específica). La Modulación Delta tiene la ventaja de que la circuitería presisada para el modulador es mas sencilla que la precisada para el PCM tradicional. Algunas de las limitaciones de Modulación Delta más sencilla (denominada también Lineal) se desprecian en la versión Delta Sigma y Delta Adaptativa.

Un Modulador Delta básico consta de un comparador y un muestreador en trayectoria directa, y de un integrador amplificador en la trayectoria de retroalimentación. El Demodulador Delta del receptor consta de un amplificador integrador (identico al de la trayectoria de retroalimentación del Modulador) seguido de un filtro pasa bajas. En PCM, las muestras de la señal analógica se cuantifican en L niveles, y esta información se transmite mediante n pulsos por muestra ($n = \log_2 L$). Una pequeña reflexión muestra que en la MD, la señal modulada porta información no acerca de las muestras de la señal sino acerca de la diferencia entre muestras sucesivas. Si la diferencia es positiva o negativa, se genera un pulso positivo o negativo (respectivamente) en el tren de pulsos modulado en Delta. Básicamente por lo tanto, la Modulación Delta porta la información acerca de la derivada de la señal analógica de entrada y, en consecuencia, resulta el nombre de Modulación en Delta. esto se puede deducir del hecho de que la integración de la señal modulada da por resultado la señal escalonada, que se aproxima a la señal analógica.

III. Materiales y Equipo / Requerimientos

No	Nombre	Requerimiento
1	Monitor	VGA ó superior.
2	Memoria RAM	De 4 a 8 MB.
3	Disco Duro	HD / con un espacio por lo menos de 3.5M.
4	Procesador	386 ó superior.
5	Programas	Software Didactico de Modulación Delta.

IV. Procedimiento

PARTE I: CORRER EL PROGRAMA DELTA

- 1) Cambiase al subdirectorio DELTA "C:\>CD DELTA"
- 2) Escriba la palabra DELTA y presione "Enter" para cargar el programa.
- 3) Luego de la pantalla de presentación presione Enter para continuar, se recomienda que haga una lectura en las opciones de ayuda ya que estas le indicarán como utilizar mejor el programa.
- 4) Ahora coloque la barra indicadora en <<Run Program Delta now!!>> y presione Enter.

PARTE II: TEORIA DENTRO DEL SOFTWARE

- 5) Del Menú Principal elija la opción 1, haga una lectura detenida a las pantallas que le presentan la teoría fundamental del Sistema de Modulación Delta.
- 6) En cualquiera de las páginas puede presionar la tecla de Escape para salir nuevamente al menú principal, aunque lo recomendable es que lo lea todo.

PARTE III: SIMULACIÓN

- 7) Del Menú Principal elija la opción 2, el diagrama de bloques y la señal que aparecen en la pantalla indican el tipo de señal y el dispositivo por el cuál es aplicada la señal. (presione Enter)
- 8) En esta gráfica aparece la señal analógica de entrada $m(t)$ con la señal escalonada $m'(t)$ que se aproxima a $m(t)$. (presione Enter)
- 9) La gráfica nos muestra la salida $d(t)$, que es el tren de pulsos modulado en Delta, puede observar en el diagrama de bloques que la señal $m'(t)$ es producto de la integración de $d(t)$, para luego ser comparada con $m(t)$. (presione Enter)
- 10) En esta gráfica se muestra la señal de error, que es la que nos dirá que nivel tendrá la señal $d(t)$, si $m(t) > m'(t)$, el pulso constante será positivo, caso contrario será si $m'(t) > m(t)$. (presione Escape)
- 11) Las opciones de <F1> a <F5> nos muestran el comportamiento de cada una de las señales en el sistema Delta, escoja cualquiera de ellas.
- 12) La primera gráfica (cuadrícula superior izquierda) muestra la señal analógica de entrada, la segunda (cuadrícula superior derecha) muestra la señal analógica con su aproximación en escalera (señal modulada), la tercer gráfica (cuadrícula inferior izquierda) es la señal $m'(t)$ sola y la ultima gráfica (cuadrícula inferior izquierda) muestra el tren de pulsos modulado en delta.
- 13) Si desea ver información de estas señales presione "Escape" para ir al menú de Ayuda de Simulación, o presione "Enter" para regresar al menú de simulaciones, escoja otra señal para simular.
- 14) Si accesó a la Ayuda de Simulaciones, escoja la señal que simuló en el paso anterior, o cualquiera si usted lo desea, despues de cada pantalla presione Enter para continuar, luego que aparezca la explicación del tren

de pulsos presione **Escape**, regresará al menú de Ayuda de Simulaciones, si desea salir escoja la opción salir y regresará al Menú Principal.

15) Si quiere hacer uso de las otras simulaciones, repita desde el paso número 7.

V. Análisis de resultados

- Que tipo de información se codifica en un sistema de Modulación Delta?
- Cuantos Bitios se utilizan en la Modulación Delta?
- Que tipo de información porta la señal modulada en Delta?



citt
CENTRO DE INVESTIGACIONES Y
TECNOLOGÍA

GUÍA No. 2



Departamento: ELECTRÓNICA Nivel: TECNOLÓGICO

Materia: APLICACIONES EN COMUNICACIÓN

Título: MODULACIÓN DELTA (Ruido Granular).

Profesor:

Instructor:

Lugar de Ejecución: Laboratorio de TELECOMUNICACIONES

Tiempo de ejecución: 4 hrs.

I. Objetivos Específicos

El estudiante será capaz de:

- Diferenciar el Ruido Granular en un sistema Delta.
- Describir las formas de reducir el Ruido granular.
- Introducir valores dentro del programa, ver los resultados y corregirlos.

II. Introducción Teórica

Refierase a la fig. 4 de la página 7 del programa Delta. La señal $s'(t)$ de la fig. 4 es la señal que encontramos también después del integrador de recepción (ver fig.1 y 2 , páginas 2 y 3 del programa Delta). La diferencia entre la señal original $s(t)$ y la señal reconstruida $s'(t)$ (ver fig. 4) se denomina ruido de cuantificación o "Ruido Granular" y es típico del proceso de Modulación Delta.

Este ruido se puede disminuir aumentando la frecuencia del clock y/o disminuyendo la amplitud "d" de la rampa; sin embargo, el aumento de la frecuencia de clock implica un aumento de la velocidad de transmisión de los bits, presisando un ensanchamiento de la banda de transmisión.

En cambio, la disminución de la amplitud "d" de la rampa acentúa otro inconveniente, la denominada "sobrecarga de pendiente" o "Slope Overload"

Dos soluciones a este problema que parece indisoluble son posibles gracias a la Modulación Delta-Sigma (hoy en desuso) y la Modulación Delta Adaptativa.

En la fig. 2 (página 3 del programa Delta) se observa que si el nivel de la señal se mantiene constante, la onda de Modulación Delta reconstruida muestra el comportamiento fluctuante que ya antes lo nombramos Ruido Granular, y que es una onda cuadrada cada medio paso del reloj. Si la rapidez del reloj es mucho mayor que el doble de la frecuencia de la señal de entrada (teorema de muestreo), la mayor parte del Ruido Granular puede eliminarse por filtrado en el receptor.

III. Materiales y Equipo / Requerimientos

No	Nombre	Requerimiento
1	Monitor	VGA ó superior.
2	Memoria RAM	De 4 a 8 MB.
3	Disco Duro	HD / con un espacio por lo menos de 3.5M.
4	Procesador	386 ó superior.
5	Programas	Software Didactico de Modulación Delta.

IV. Procedimiento

PARTE I: CORRER EL PROGRAMA DELTA

- 1) En forma de repaso, cargue el programa Delta en la sección de teoría, cite las páginas 9 y 10, trate de entenderlas.
- 2) Regrese al Menú Principal, escoja la opción de la práctica (3).
- 3) Escoja la señal Senoidal <F1>.
- 4) Introduzca los siguientes valores:
 - a) Frecuencia de la señal de entrada: 250 Hz
 - b) Amplitud de la señal de entrada: 7V
 - c) Frecuencia de muestreo: 25000 Hz
 - d) Delta (altura del escalón): 15

nota: Si existe un error en los datos, antes de graficar el programa preguntara si están correctos, si están correctos presione Enter de lo contrario presione Escape.

5) Se puede observar que, cuando la señal analógica alcanza su máxima amplitud, el sistema Delta detecta como si fuese un pulso constante en amplitud y lo mismo sucede cuando está en la parte más baja de la señal. A este fenómeno se le conoce como Ruido Granular. Esto sucede porque el tamaño de el escalón es grande y no logra diferenciar las pequeñas pendientes. (presione Enter para continuar)

6) Introduzca estos nuevos datos:

- a) 250 Hz
- b) 7 V
- c) 25000 Hz
- d) 5

- Anteriormente se observó que el Ruido Granular existía no solo en la cresta de la señal, sino también a media amplitud es decir en la trayectoria de su ascenso y descenso, ahora podemos ver que el tamaño del escalón es más pequeño y puede detectar los cambios en la pendiente.

7) Sería lógico pensar que con disminuir más el tamaño del escalón (delta) se solucionaría todo ; hagamos una prueba e introduzca los siguientes datos:

- a) 250 Hz
- b) 7 V
- c) 25000 Hz
- d) 2.5



Departamento: ELECTRÓNICA **Nivel:** TECNOLÓGICO
Materia: APLICACIONES EN COMUNICACIÓN
Título: MODULACIÓN DELTA (Sobrecarga de Pendiente).
Profesor:
Instructor:
Lugar de Ejecución: Laboratorio de TELECOMUNICACIONES
Tiempo de ejecución: 4 hrs.

I. Objetivos Específicos

El estudiante será capaz de:

- Detectar la Sobrecarga de Pendiente en un sistema Delta.
- Describir las formas de reducir la Sobrecarga de Pendiente.
- Introducir valores dentro del programa, ver los resultados y corregirlos.

II. Introducción Teórica

La disminución de la amplitud "d" de la rampa acentúa otro inconveniente, la denominada "Sobrecarga de Pendiente" o "Slope Overload". El efecto debido a la sobrecarga de pendiente se muestra en el trecho "A" de la fig. 4 de la página 7 del programa Delta. Este efecto aparece cuando la señal moduladora varía su amplitud entre dos clocks sucesivos en una cantidad superior a la amplitud "d" de la rampa.

Este tipo de distorsión no es causada por la amplitud de la señal moduladora, sino por su pendiente. Para disminuir el efecto debido a la Sobrecarga de Pendiente, se puede aumentar la frecuencia de clock (pero incurriendo en el mismo inconveniente visto a propósito del ruido granular), o bien se puede aumentar la amplitud "d" de la rampa, pero esto conllevaría un aumento del ruido granular.

Las variaciones en la señal analógica de entrada $m(t)$ menores que el valor del escalón (umbral de codificación) se pierden en la Modulación Delta. Al mismo tiempo, si $m(t)$ cambia con mucha rapidez (o sea $m(t)$ es demasiado alta), la señal aproximativa $m'(t)$ no podrá seguir a $m(t)$, y se presenta la sobrecarga.

Esta es la llamada sobrecarga de pendiente, y es uno de los factores limitantes básicos en el funcionamiento de la Modulación Delta. Debemos esperar sobrecarga de pendiente más que sobrecarga de amplitud en la Modulación Delta, ya que esta modulación básicamente porta la información acerca de $m(t)$.

La sobrecarga de pendiente se presenta cuando $m'(t)$ no puede seguir a $m(t)$. Durante un intervalo de muestreo T_s , $m'(t)$ es capaz de cambiar en σ/T_s , o bien en σf_s , donde f_s es la frecuencia de muestreo.

III. Materiales y Equipo / Requerimientos

No	Nombre	Requerimiento
1	Monitor	VGA ó superior.
2	Memoria RAM	De 4 a 8 MB.
3	Disco Duro	HD / con un espacio por lo menos de 3.5M.
4	Procesador	386 ó superior.
5	Programas	Software Didactico de Modulación Delta.

IV. Procedimiento

PARTE I: CORRER EL PROGRAMA DELTA

- 1) En forma de repaso, cargue el programa Delta en la sección de teoría, cite las páginas 9 y 10, trate de entenderlas.
- 2) Regrese al Menú Principal, escoja la opción de la práctica (3).
- 3) Escoja la señal Senoidal <F1>.
- 4) Introduzca los siguientes valores:
 - a) Frecuencia de la señal de entrada: 750 Hz
 - b) Amplitud de la señal de entrada: 7V
 - c) Frecuencia de muestreo: 50000 Hz
 - d) Delta (altura del escalón): 5

nota: Si existe un error en los datos, antes de graficar el programa preguntara si están correctos, si están correctos presione Enter de lo contrario presione Escape.

- 5) Se puede observar que, cuando la señal analógica decrece en su amplitud, su pendiente es muy alta y la señal escalonada no es capaz de reconocer los cambios de pendiente que $m(t)$ presenta, es por eso que se produce la Sobrecarga de Pendiente o Slope Overload. (presione Enter para continuar)
- 6) Introduzca estos nuevos datos:
 - a) 375 Hz
 - b) 7 V
 - c) 25000 Hz
 - d) 5

- Observe al principio de la gráfica, $m(t)$ parte de cero con una pendiente alta, la cuál no puede ser alcanzada por la señal escalonada, cuando esta detecta la cresta de la señal $m(t)$ reconoce el cambio de pendiente y es capaz de seguirla al principio de su trayectoria hacia abajo, luego se experimenta un cambio de pendiente muy brusco que el sistema Delta no lo puede detectar.

- 7) Seleccione una señal coseno.
- 8) Introduzca los siguientes valores:

- a) 250 Hz
- b) 7 V
- c) 25000 Hz
- d) 5

- No fueron muy buenos los resultados, pero hemos descubierto un fenómeno llamado sobrecarga de pendiente, que se estudiará posteriormente. (presione Escape)

8) Repita los pasos 4,5,6 y 7 cambiando el tipo de señal, escogiendo primero una coseno y luego una cuadrada. Que observa? _____

- Se puede apreciar que para la señal cuadrada es difícil eliminar el ruido granular.

9) Introduzca los siguientes valores para una señal senoidal.

- a) 250 Hz
- b) 7 V
- c) 50000 Hz
- d) 2.5

- Se ve que la señal escalonada se aproxima mejor a la señal analógica, menor ruido granular y sobrecarga de pendiente, el sistema detecta mejor los cambios de pendiente.

10) Repita el paso 9 con los otros tipos de señal.

Haga su propia conclusión: _____

V. Análisis de resultados

- Como se le llama a la diferencia entre la señal originaria $m(t)$ y la señal aproximada $m'(t)$?

- ¿Mencione un método para eliminar el Ruido Granular?

-Cuál es la relación que debe existir entre la frecuencia del clock y la de la señal analógica para poder eliminar la mayor parte del Ruido Granular ?

- Observese que las dos señales parten de cero, y que la señal coseno experimenta un cambio brusco de 0-7V, por lo tanto hay sobrecarga de pendiente en la señal $m'(t)$.

Al alcanzar $m'(t)$ a $m(t)$ en amplitud, comienza a detectar la forma de onda y reconoce los cambios en la pendiente, de aquí en adelante detecta fácilmente los cambios de amplitud.

9) Seleccione una señal cuadrada e introduzca los siguientes valores .

- a) 250 Hz
- b) 7 V
- c) 50000 Hz
- d) 5

- El cambio de estado de una señal cuadrada lo hace en $t=0$, es por eso que el sistema Delta no puede registrarlo y ocurre la Sobrecarga de Pendiente.

10) Repita el paso 9 con los otros tipos de señal.

Haga su propia conclusión: _____

- La sobrecarga de pendiente siempre existe en la señal cuadrada.

V. Análisis de resultados

- Que factor causa Sobrecarga de Pendiente en un sistema Delta?
- ¿Mencione un método para disminuir la Sobrecarga de Pendiente?
- Que tipo de información porta la Modulación Delta?

ANEXO 1.2

-Presentación de la Teoría dentro del Software-

MODULACIÓN DELTA

La Modulación Delta es una técnica que permite codificar una señal analógica en cifras binarias (bit).

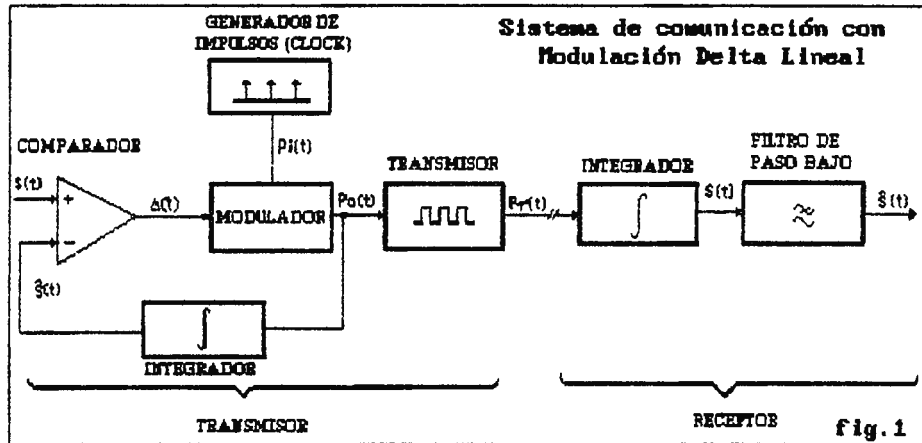
Este proceso se realiza mediante las dos operaciones clásicas de muestreo y codificación de la señal a transmitir, por ello la modulación Delta se puede considerar un sistema PCM o NIC todos los efectos (aunque si con el término PCM se identifica una técnica de codificación específica).

La modulación Delta tiene la ventaja de que la circuitería precisada para el modulador y el demodulador es más sencilla que la precisada para el PCM tradicional.

Algunas de las limitaciones de modulación Delta mas sencilla (denominada también Lineal) se desprecian en la version Delta Signa y Delta Adaptativa .

MODULACIÓN DELTA LINEAL.

Un sistema de comunicación que utiliza la modulación Delta Lineal se muestra en la fig.1 y la fig.2 muestra las formas de onda características.



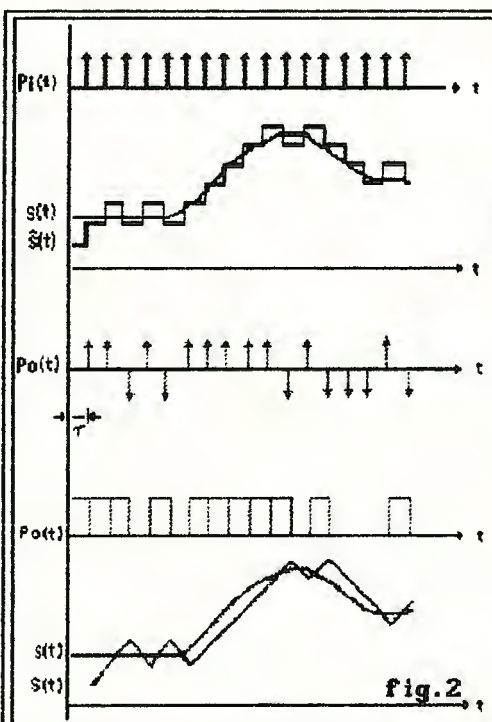


fig.2

encontramos $\hat{s}(t)$. Como se observa en la fig.2, la señal escalonada es una aproximación de la de la señal analógica de entrada $s(t)$.

Un generador de impulsos suministra al modulador un tren de impulsos $P_i(t)$. En la otra entrada el modulador recibe la señal $Q(t)$. La salida $P_o(t)$ del modulador no es otra cosa que el tren de impulsos de entrada $P_i(t)$ multiplicado por $+1$ ó -1 , en función del signo (no de la amplitud) de $Q(t)$. Si $Q(t)$ es positiva cuando llega el impulso $P_i(t)$, entonces la multiplicación es por $+1$; mientras que si $Q(t)$ es negativa, la multiplicación será por -1 .

La señal impulsiva $P_o(t)$ se aplica a un integrador en cuya salida



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14

Las señales $s(t)$ y $\hat{s}(t)$ son comparadas por un circuito comparador cuya salida será positiva [$A(t) > 0$ si $\hat{s}(t) > s(t)$] o negativa [$A(t) < 0$ si $\hat{s}(t) < s(t)$].

Por lo tanto, ahora está claro el significado físico de $\Delta(t)$ como diferencia (en signo y no en amplitud) entre la señal a codificar $s(t)$ y la señal que se aproxima $\hat{s}(t)$.

Teóricamente se podría pensar de enviar al canal de comunicación el tren de impulsos positivos y negativos $P_o(t)$; sin embargo, en la práctica se envía en línea una señal de dos niveles, obtenida de $P_o(t)$, de modo que se aumente la energía de la señal transmitida. En línea encontramos la señal $P_r(t)$, obtenida de $P_o(t)$ asignando a un nivel "alto" de duración τ para cada impulso positivo y un nivel "bajo" de duración τ para cada impulso negativo. A este punto ya podemos observar una diferencia fundamental entre el PCM y la modulación Delta: en Delta no se transmite una codificación binaria de la señal $s(t)$ (como ocurre en PCM), sino una codificación binaria (de un bit) de la diferencia $\Delta(t)$ entre $s(t)$ y una señal que se aproxima $\hat{s}(t)$.



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Por lo tanto, la modulación delta está representada por una serie de "0" y "1" lógicos, que codifican la derivada de una señal analógica: un "cero" indica una disminución de la señal, mientras que un "uno" se asocia a un aumento.

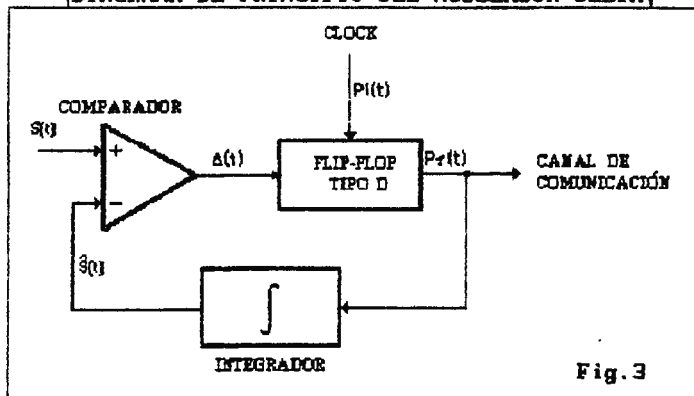
El demodulador Delta está constituido por un simple integrador seguido de un filtro de paso bajo. EL integrador reconstruye una señal $s(t)$ que aproxima la señal moduladora $s(t)$ a través de una sucesión de rampas, el filtro de paso bajo suprime los residuos de la frecuencia de clock. Se obtiene así, a la salida del sistema transmisor, la señal $s(t)$ que, salvo diferencias debidas ruidos de distinta naturaleza, es idéntica a la señal moduladora $s(t)$.

Otro esquema típico de un modulador Delta Lineal se muestra en la fig. 3. El comportamiento de este modulador es totalmente similar al de la fig.1, pero con la diferencia de que al integrador llegan impulsos de duración t , por lo tanto la salida del integrador es una sucesión de rampas.



1
2
3
4
5
6
7
8
9
10
11
12
13
14

DIAGRAMA DE PRINCIPIO DEL MODULADOR DELTA



1
2
3
4
5
6
7
8
9
10
11
12
13
14

FORMAS DE ONDA RELATIVAS AL MODULADOR DE LA FIG. ANTERIOR

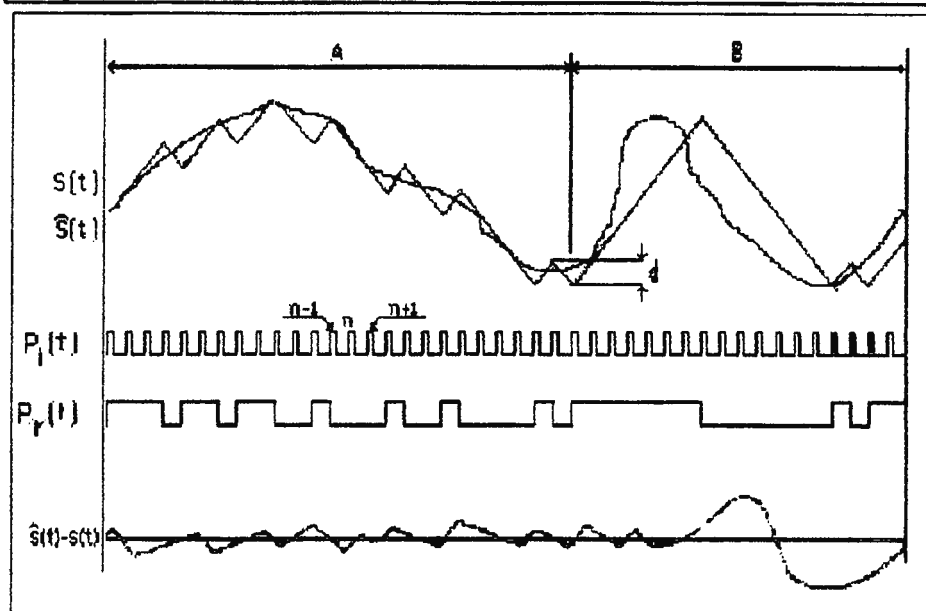


Fig. 4

presione Esc. para salir..

EXPLICACIÓN DEL FUNCIONAMIENTO

El funcionamiento se puede explicar haciendo referencia a las figuras 3 y 4. Considerese, a título de ejemplo, el instante correspondiente al genérico impulso de clock "n": la señal $\hat{s}(t)$ presenta pendiente negativa ya que, en el instante de muestreo anterior, el valor de la señal analógica $s(t)$, era inferior al de la salida del integrador, por lo tanto, la salida del flip-flop tipo D era "0".

En el instante "n", $s(t)$ es todavía inferior a $\hat{s}(t)$, por lo que la salida del comparador es todavía baja y también la salida del flip-flop: $s(t)$ continúa a descender.

En el instante "n+1", la situación cambia ya que $s(t)$ resulta mayor que $\hat{s}(t)$, por lo que la salida del comparador es alta y también la salida del flip-flop: $\hat{s}(t)$ invierte la pendiente y comienza a subir, siguiendo la marcha de $s(t)$.

42
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

La señal $\hat{s}(t)$ de la Fig.4 es la señal que encontramos también después de 1 integrador de recepción (ver Fig.1 y 2). La diferencia entre la señal original $s(t)$ y la señal reconstruida $\hat{s}(t)$ (se muestra en la Fig.4) se denomina ruido de cuantificación o "ruido granular" y es típico del proceso de modulación Delta.

Este ruido se puede disminuir aumentando la frecuencia de clock y/o disminuyendo la amplitud "d" de la rampa ; sin embargo el aumento de la frecuencia de clock implica un aumento de la velocidad de transmisión de los bits, presizando un ensanchamiento de la banda de transmisión.

En cambio, la disminución de la amplitud "d" de la rampa acentúa otro inconveniente , la denominada "sobrecarga de pendiente" o "slope overload". El efecto debido a sobrecarga de pendiente se muestra en el trecho "A" de la Fig.4 . Este efecto aparece cuando la señal moduladora varía su amplitud entre dos clocks sucesivos en una cantidad superior a la amplitud "d" de la rampa.

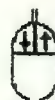


- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14

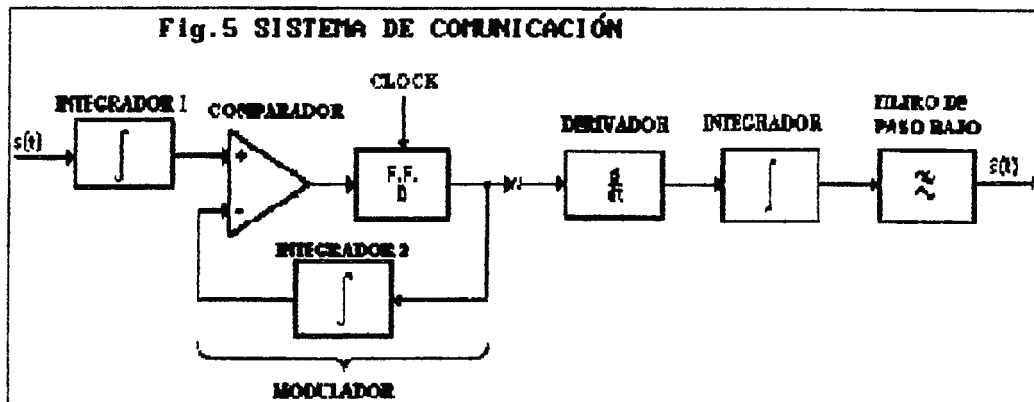
cont. Ruido de Cuantificación..

Este tipo de distorsión no es causada por la amplitud de la señal moduladora, sino por su pendiente. Para disminuir el efecto debido a la sobrecarga de pendiente, se puede aumentar la frecuencia de clock (pero incurriendo en el mismo inconveniente visto a propósito del ruido de cuantificación), o bien se puede aumentar la amplitud "d" de la rampa, pero esto conllevaría a un aumento del ruido de cuantificación.

Este problema que al parecer no tiene solución se puede resolver con el método llamado modulación Delta Sigma (hoy en desuso) y la modulación Delta Adaptativa.



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14



MODULACION DELTA SIGMA.

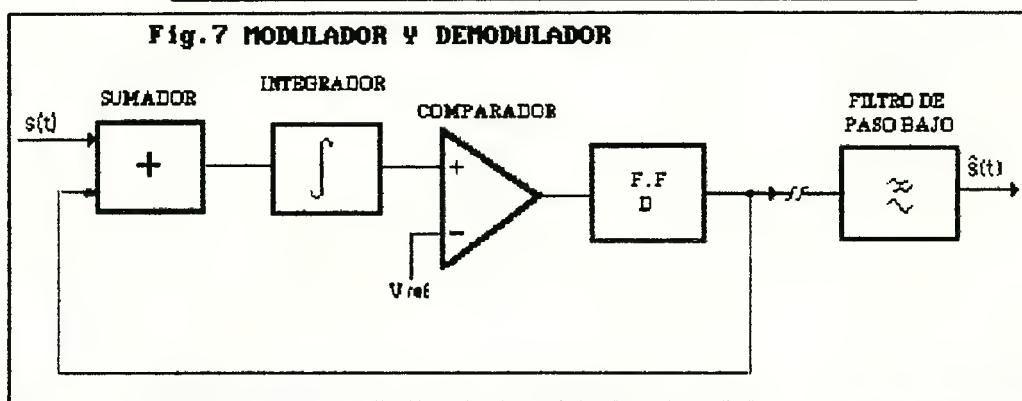
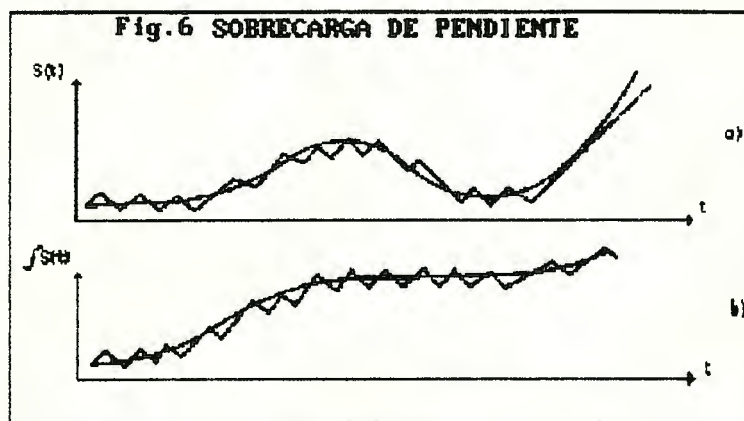
Una técnica para disminuir la pendiente de la señal analógica aplicada al modulador Delta es la de integrar previamente la señal introduciendo así una espacie de preénfasis . en recepción se tendrá que añadir, con respecto a la modulación Delta Lineal , la función inversa a la integración, o sea una derivación . El diagrama de bloques de un sistema de comunicación Delta - Sigma se muestra en la Fig.5.

El modulador no opera ahora en la señal $s(t)$ sino en su integral, y como es conocido la función obtenida integrando $s(t)$ tiene una pendiente que generalmente es inferior a la de $s(t)$ y por lo tanto se percibirá menos el efecto debido a la sobrecarga de pendiente. La Fig.6 muestra las curvas de la señal analógica que se aproxima, en donde en a) se consideró la señal analógica $s(t)$ y en b) su integral: en el caso b) es notablemente inferior el efecto de sobrecarga de pendiente.

El diagrama de bloques de la Fig.5 se puede reducir si se considera que: la función desarrollada por el integrador 2 (del modulador) puede realizarse con el integrador 1; las funciones de derivación y de integración (en el receptor) se eliden . Se obtiene así el diagrama de bloques de la Fig.7.



1
2
3
4
5
6
7
8
9
10
11
12
13
14



Como se describió anteriormente , la modulación Delta Lineal presenta los inconvenientes del ruido de cuantificación y de la sobrecarga de pendiente ; la disminución de uno de ellos implica generalmente el aumento del otro y viceversa. Utilizando la modulación Delta Sigma es posible disminuir la sobrecarga de pendiente sin empeorar el ruido de cuantificación, pero el mejoramiento es todavía limitado.

Un método eficaz para reducir al mínimo los inconvenientes expuestos utiliza el [companding (compressing / expanding) en castellano "Compansión" (compresión /expansión)], mediante el cual el modulador varía su ganancia en base a la amplitud de la señal analógica de entrada. De esta manera se podrá tener una pequeña amplitud de la rampa cuando la señal es baja y una amplitud mayor cuando la señal aumenta.

presione Esc. para salir..

Por lo tanto, la amplitud de la rampa se adapta a la amplitud de la señal y esta es la razón por la cual la modulación se denomina adaptativa. Un esquema difundido y eficaz para realizar la modulación Delta Adaptativa es conocido como CVSD (Continuously Variable Slope Deltamod), implementado en distintos circuitos integrados presentes en el mercado. En la Fig.8 se muestra un diagrama de bloques genérico de un modulador y demodulador realizado con CVSD.

Como se puede observar en el diagrama del modulador, la única diferencia respecto a la modulación Delta Lineal está constituida por la presencia, en la cadena de realimentación, de un amplificador controlado en tensión.

La tensión de control no es otra cosa que la señal digital suministrada por el modulador, integrado (por el integrador 2 , denominado también *filtro silábico*) con una constante de tiempo mayor que la del integrador principal (integrador 1):esto por que la tensión de control obtenida de esta manera tiene que poder seguir en promedio variaciones más lentas en el tiempo.

presione Esc. para salir..

ESQUEMA DE BLOQUES de MODULADOR DELTA ADAPTIVO realizado con CVSD

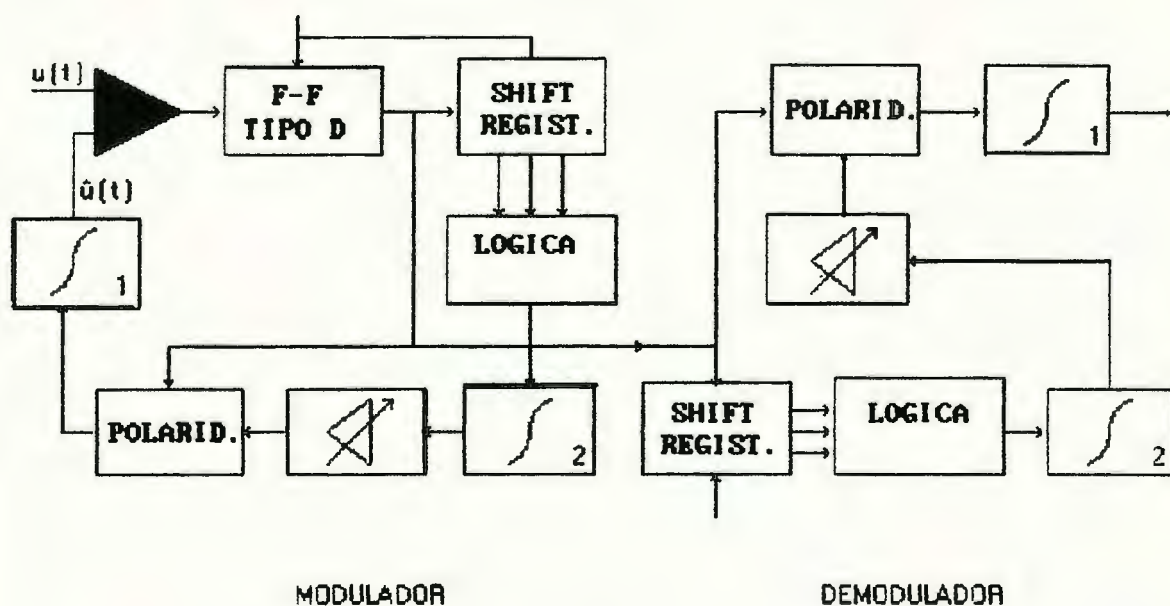


Fig. 8

presione Esc. para salir..



16

17

18

cont.

Analícenos ahora de manera profundizada el funcionamiento de este modulador.

La señal digital saliente del flip-flop D, que constituye el resultado de la modulación, además que en la línea se aplica también a un registro de desplazamiento de 3 bit. El algoritmo utilizado para el CUSD de la Fig.8 simplemente realiza un control de los tres últimos bits suministrados por el modulador y verifica si se generaron tres "0" o tres "1" seguidos. Esta codificación se denomina también coincidencia .

Cuando la misma se verifica, significa que la ganancia del integrador 1 es demasiado baja. La salida del detector de coincidencia (marcado con lógica en la Fig.8) es integrado por el integrador 2. La tensión de salida del filtro silábico controla la ganancia del integrador 1 a través de un circuito que le suministra al integrador también el del bit ("0" ó "1") suministrado en ese instante por el modulador.

presione Esc. para salir..

cont...

Con esta técnica se obtiene una medida del nivel promedio de la señal de entrada y en base al nivel promedio se varía la ganancia del integrador. De esta manera, para señales débiles la rampa tendrá una altura reducida (por lo tanto, con ruido de cuantificación reducido) y para señales fuertes una altura mayor (por lo tanto, sobrecarga de pendiente reducida).

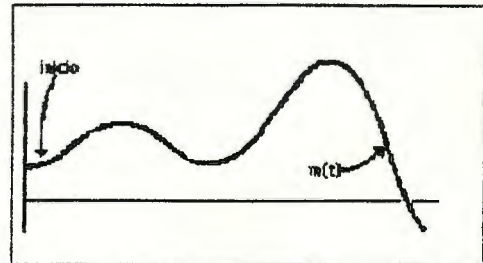
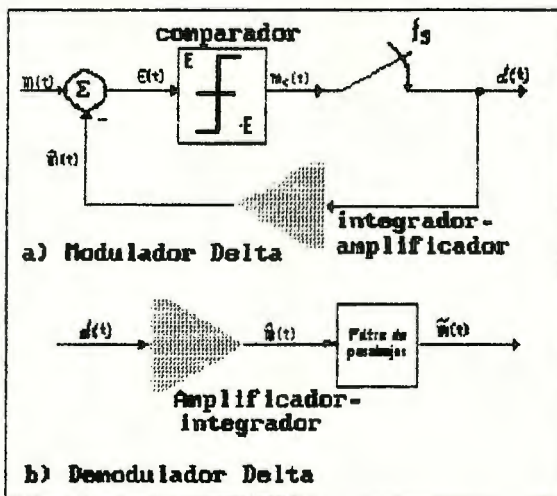
En recepción , el demodulador tendrá que utilizar la misma técnica para reconstruir la señal.

presione Esc. para salir..

ANEXO 2.

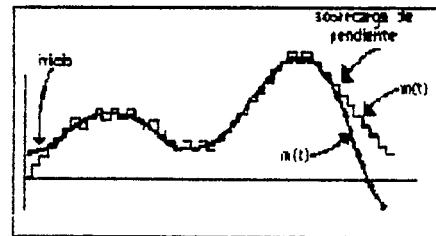
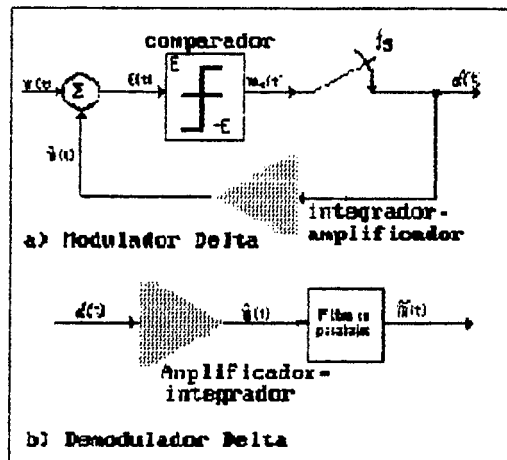
-GRÁFICAS DE SIMULACIÓN-

-GRÁFICOS DE PRÁCTICAS-



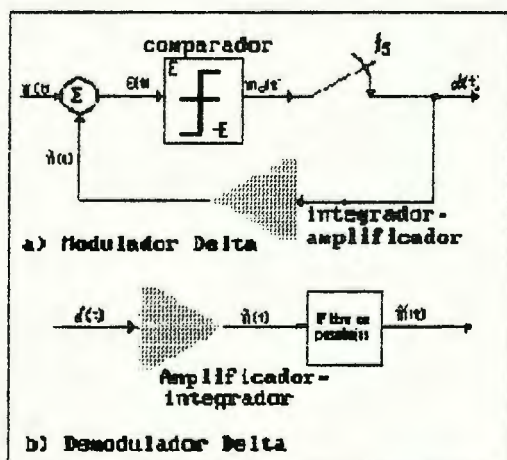
La señal analógica $m(t)$ se compara con la señal de retroalimentación $\hat{m}(t)$.

Primera imagen del menú de simulación.

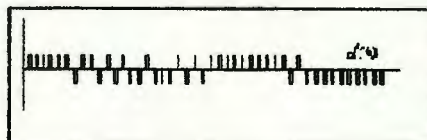
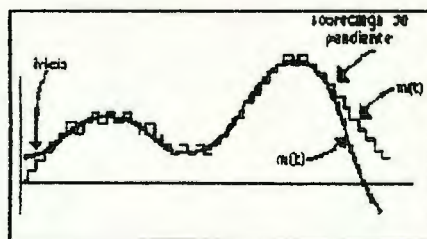


Puede verse que $\hat{x}(t)$ es un tipo de aproximación en escalera de $m(t)$.

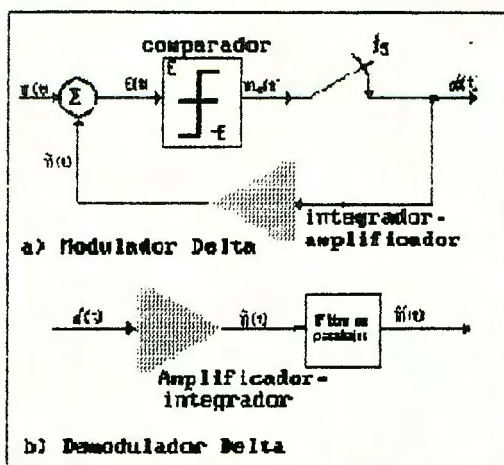
Segunda imagen del menú de simulaciones.



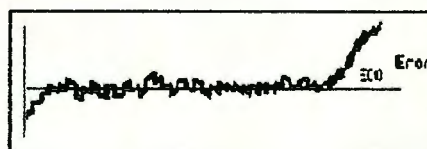
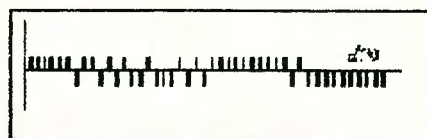
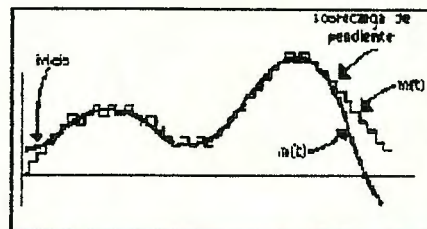
El tren de pulsos $d(t)$ es el tren de pulsos modulado en Delta.



Tercera imagen del
menú de simulaciones.



La señal de error $E(t)$ se obtiene de : $E(t) = n(t) - \hat{n}(t)$



Proxima Esc. para continuar...?

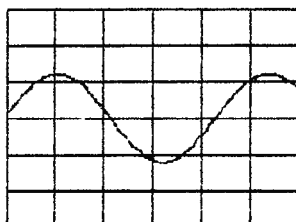
Última imagen del menú de simulaciones

MENU DE SIMULACION

- <F1> SENOIDAL
- <F2> COSENOIDAL
- <F3> RAMPA
- <F4> CUADRADA
- <F5> COMPUESTA
- <Esc> Menu Principal

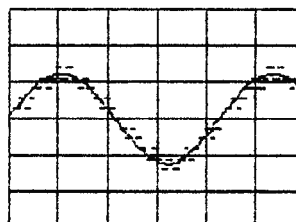
Menu de Simulaciones

Esc Menu Ppal.

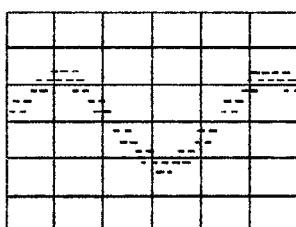


Señal de Entrada

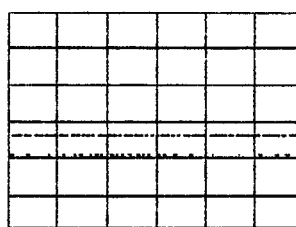
Enter Menu Simul.



Señal Modul. Entr.



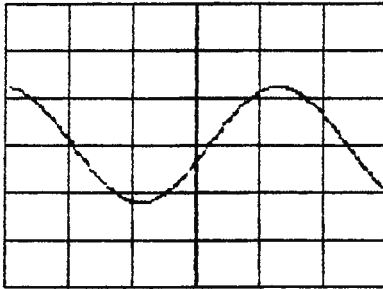
Señal Modulada



Señal Digital

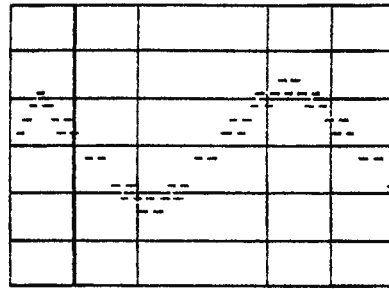
Simulación de un señal Seno en un sistema Delta.

Esc Menu Ppal.

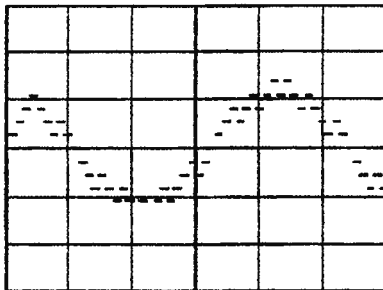


Señal de Entrada

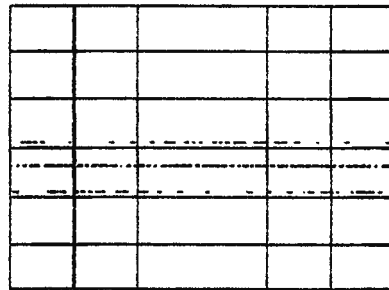
Enter Menu Sinul



Señal Modu./Ent.



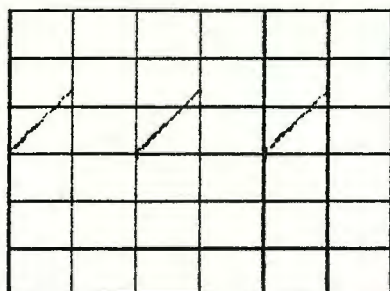
Señal Modulada



Señal Digital

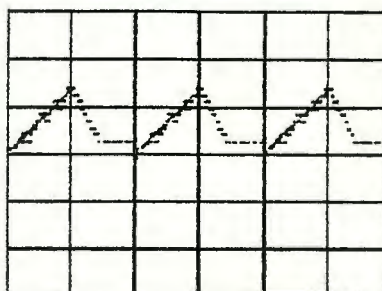
Señal Coseno simulada en un sistema Delta.

Esc Menu Ppal.

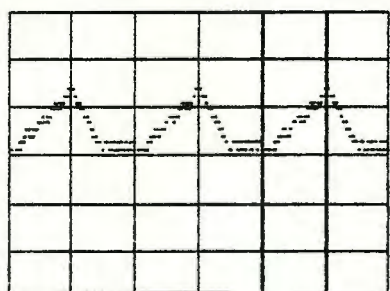


Señal de Entrada

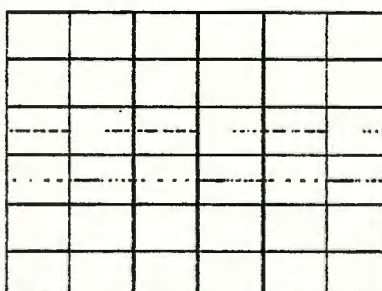
Enter menu Simul.



Señal Modu. / Ent.



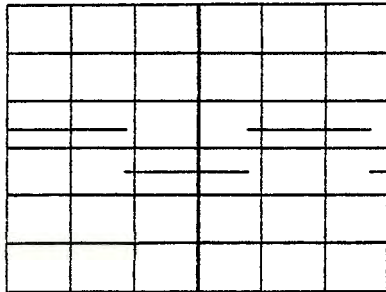
Señal Modulada



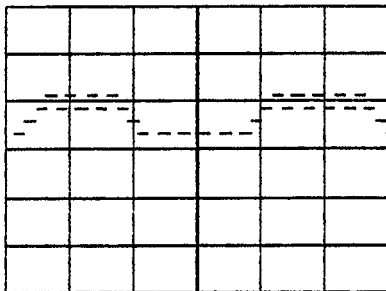
Señal Digital.

Simulación de una señal Rampa en un sistema Delta

Esc Menu Ppal.

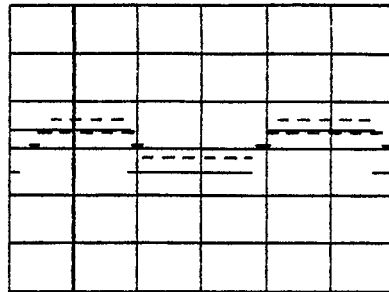


Senal de Entrada

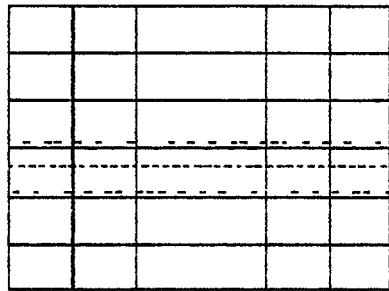


Senal Modulada

Enter menu Sinl.



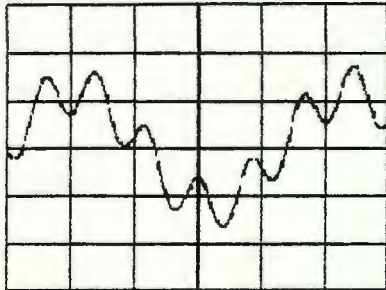
Senal Modul./Ent.



Senal Digital.

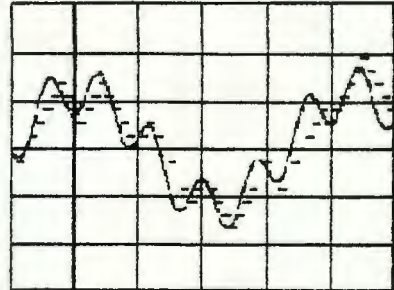
Simulación de una señal cuada en un sistema Delta.

Esc Menu Ppal.

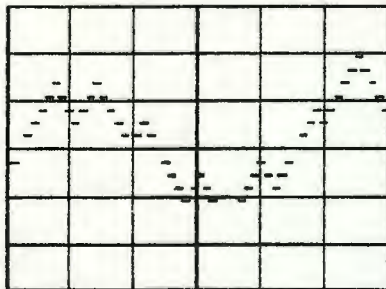


Señal de Entrada

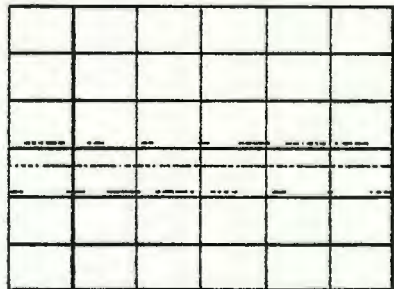
Enter Menu Simul.



Señal Modul./Ent.



Señal Modulada

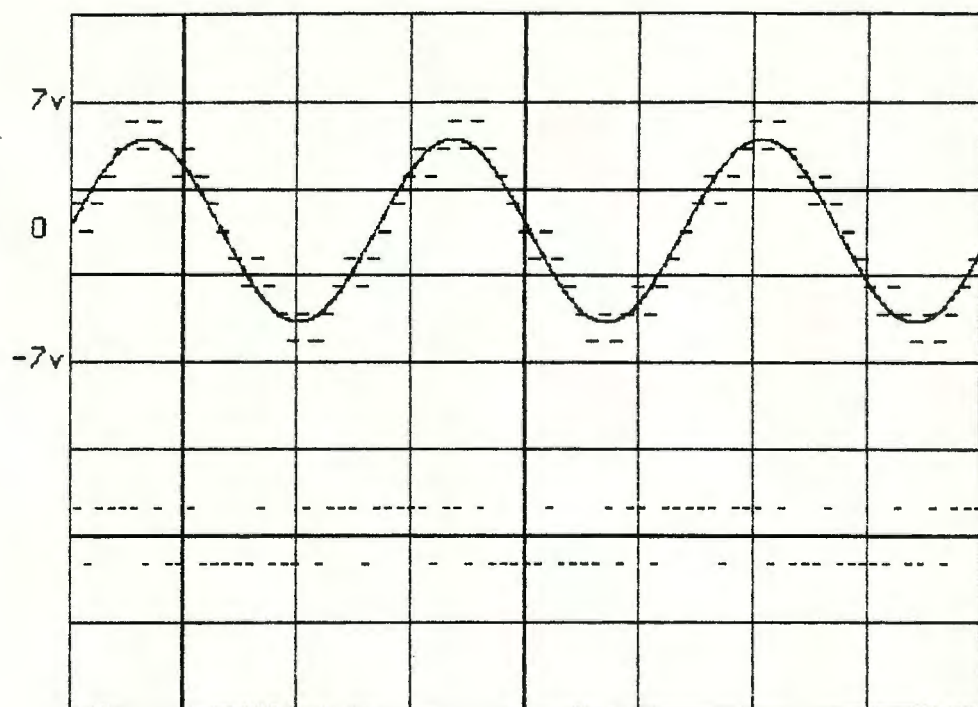


Señal Digital

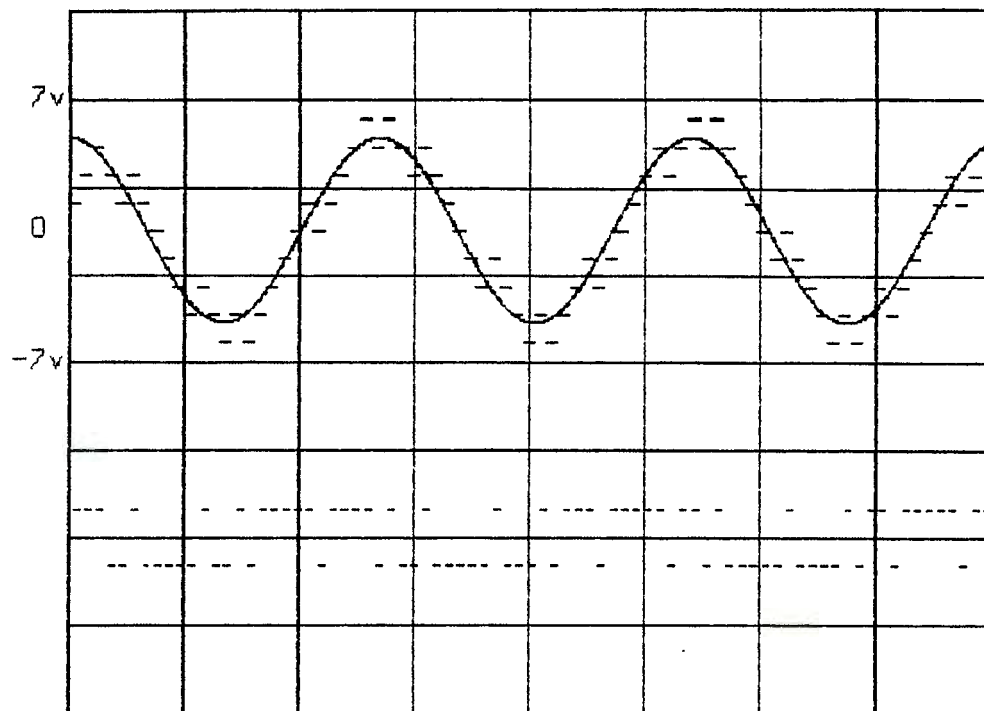
Simulación de una señal compuesta en un sistema delta



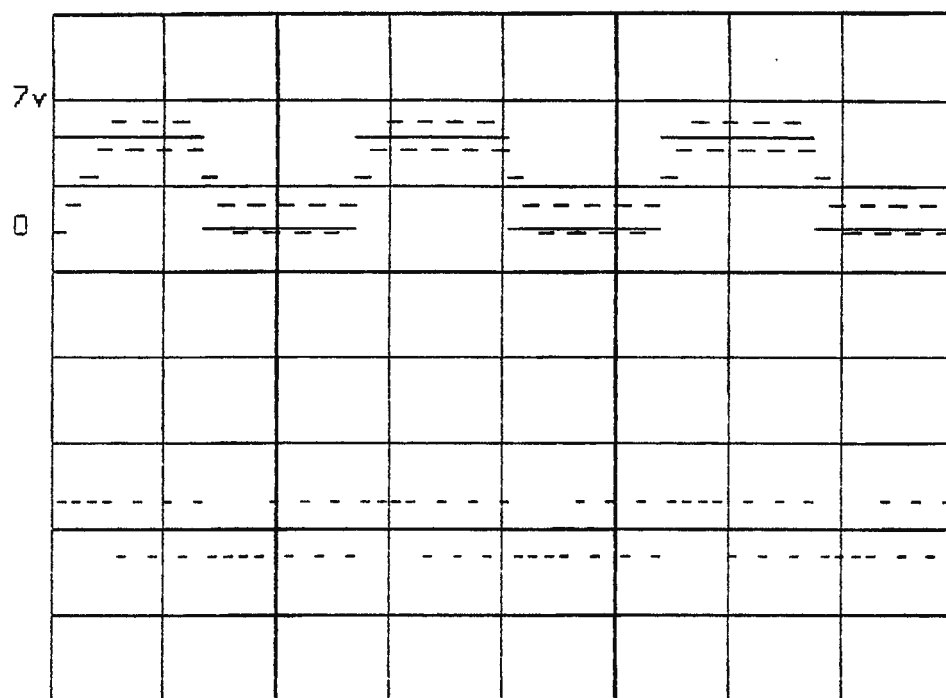
Menú de prácticas



Señal Senoidal (Práctica)



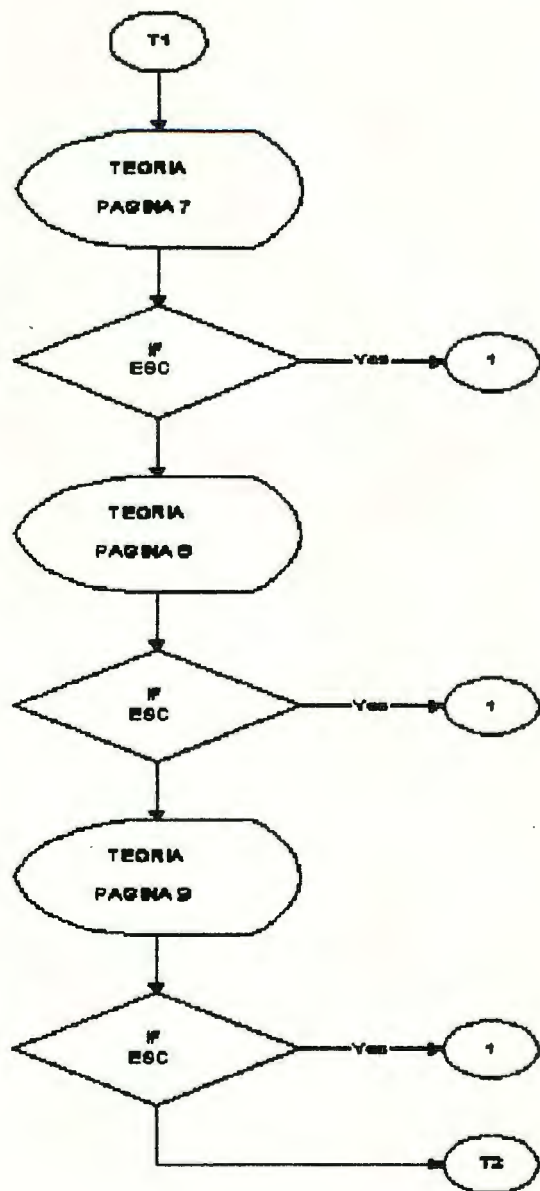
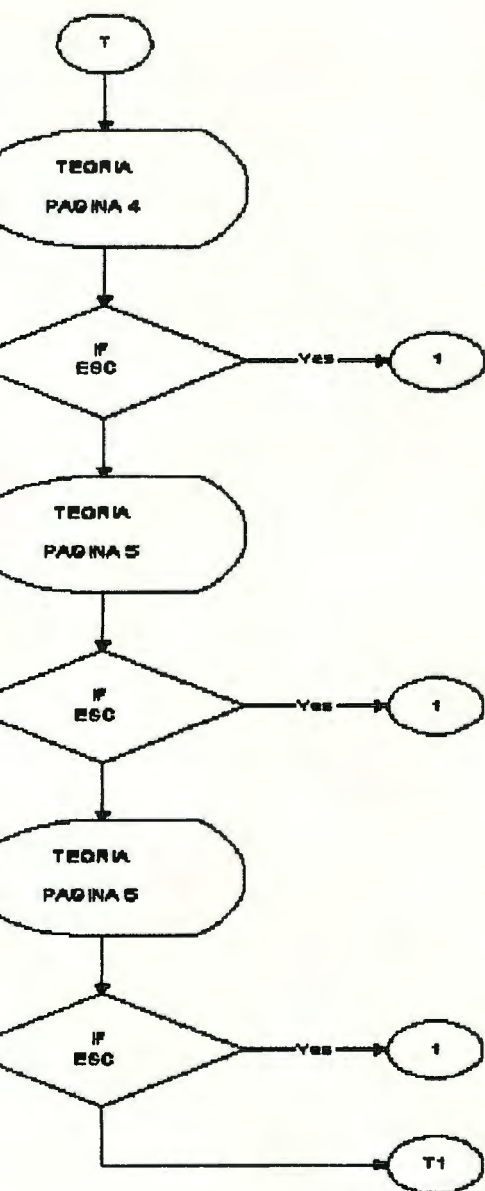
Señal Coseno (Práctica).

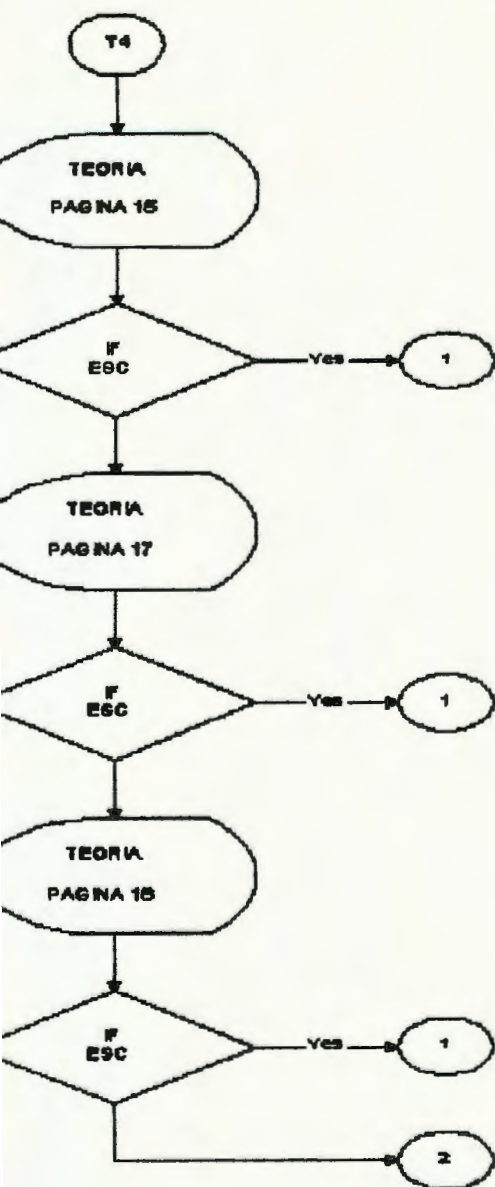


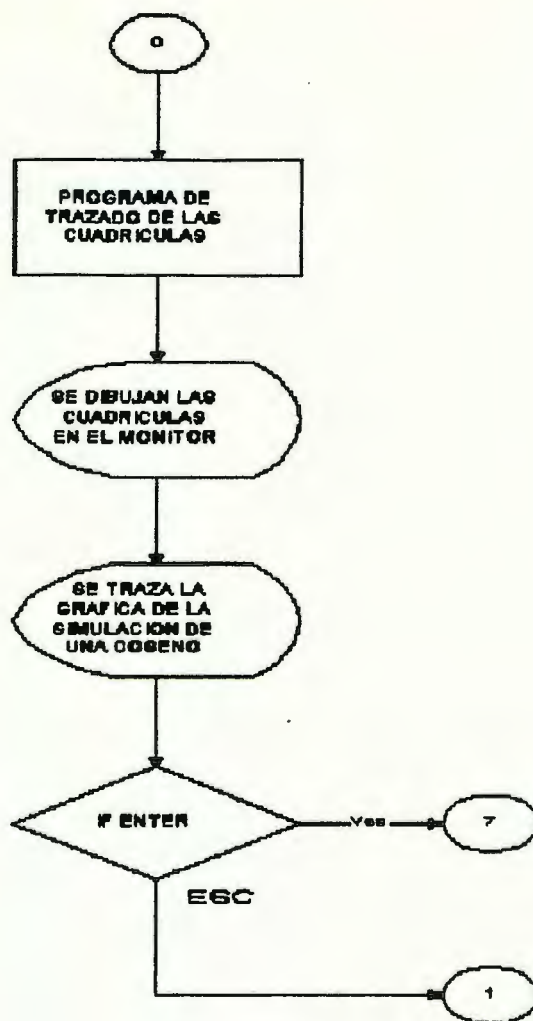
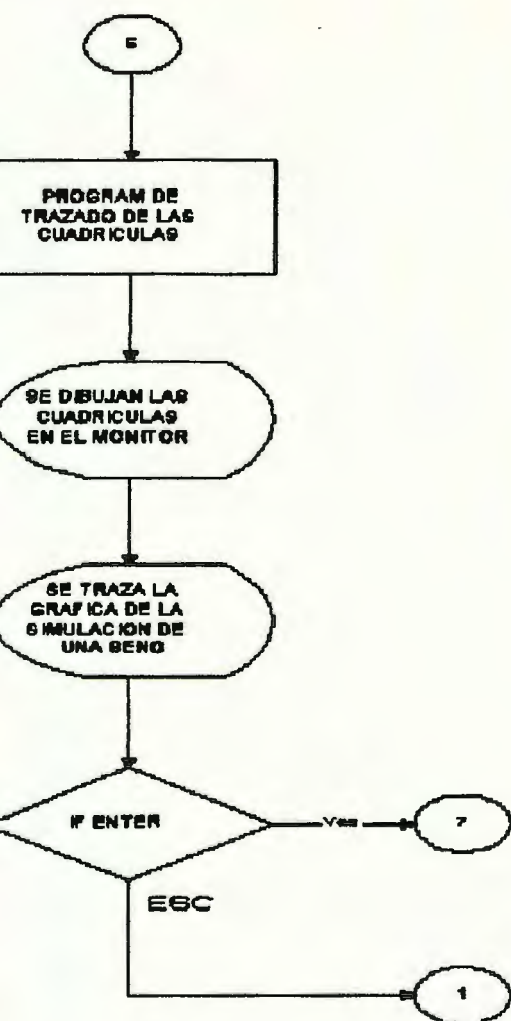
Esc. Menu Practicas

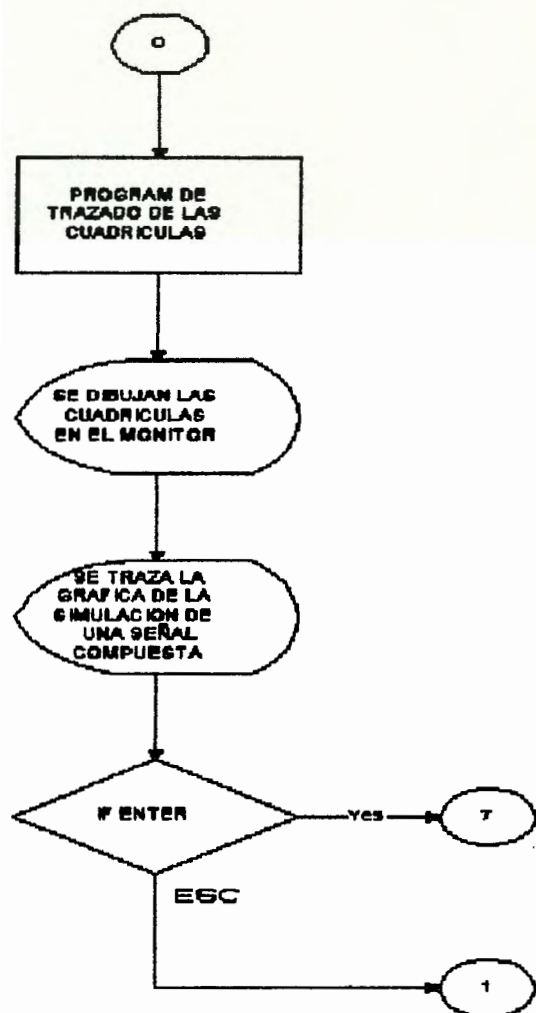
Ent. Otro Calculo

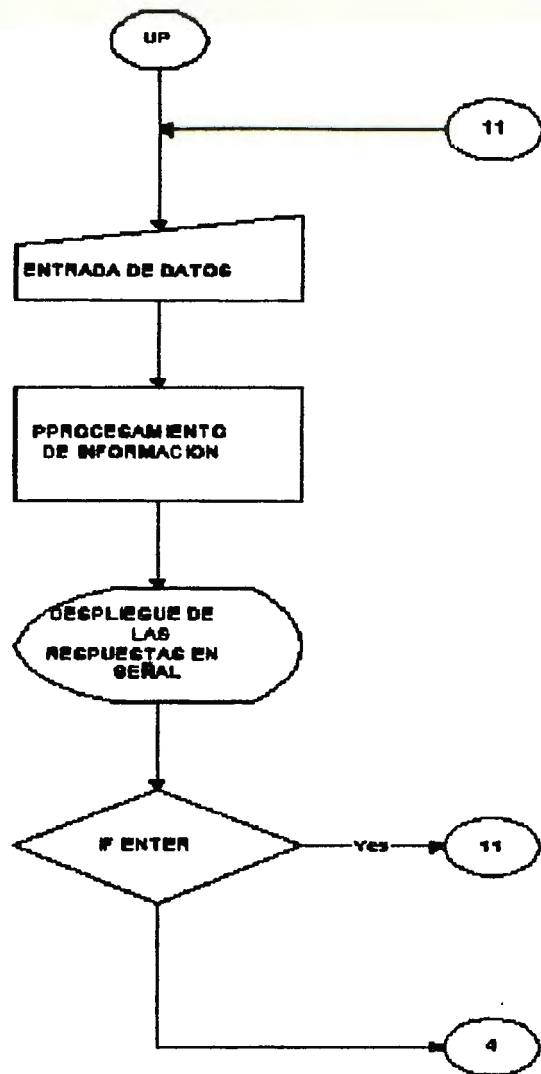
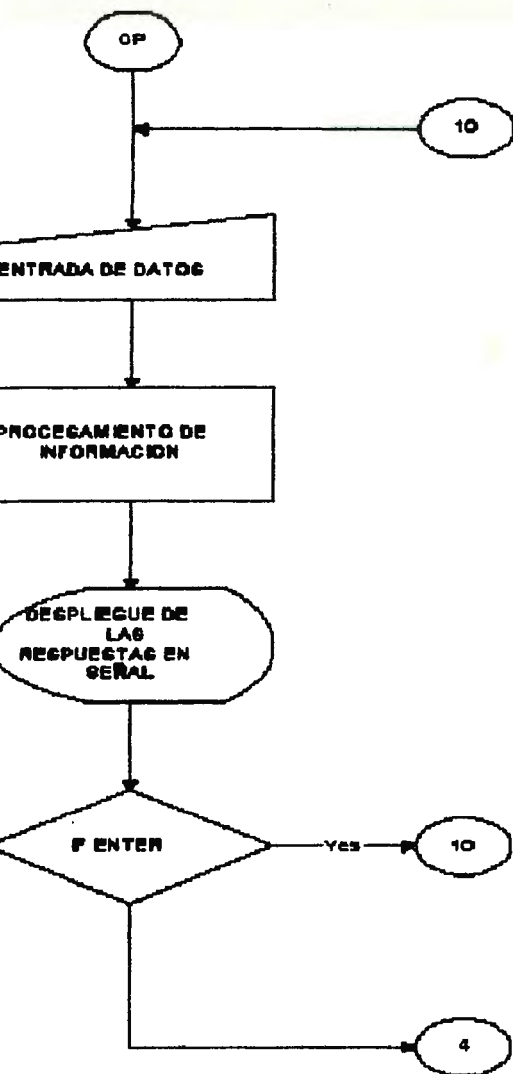
Señal Cuadrada (Práctica).











"PROGRAMA PARA EJECUTAR EL MENÚ DE SIMULACIONES"
"LENGUAJE TURBO C"

```
/*PROGRAMA PARA SIMULACIÓN DEL SISTEMA DELTA*/
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{
    int i,k,key;
    int driver,mode;
    char msg[80];
    driver=VGA;
    mode=VGAHI;
    initgraph(&driver,&mode,"");

    menu:
    cleardevice();
    /*MENU*/
    setcolor(7);
    rectangle(140,90,500,390);
    setfillstyle(SOLID_FILL,7);
    floodfill(141,91,7);
    setcolor(3);
    rectangle(145,95,495,385);
    setfillstyle(INTERLEAVE_FILL,3);
    floodfill(146,96,3);
    setcolor(7);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
    sprintf(msg,"MENU DE SIMULACION");
    outtextxy(200,110,msg);
    sprintf(msg," <F1>  SENOIDAL      ");
    outtextxy(160,160,msg);
    sprintf(msg," <F2>  COSENOIDAL     ");
    outtextxy(160,195,msg);
    sprintf(msg," <F3>  RAMPA           ");
    outtextxy(160,230,msg);
    sprintf(msg," <F4>  CUADRADA        ");
    outtextxy(160,265,msg);
    sprintf(msg," <F5>  COMPUESTA       ");
    outtextxy(160,300,msg);
    sprintf(msg," <Esc> Menu Principal");
    outtextxy(160,335,msg);
    key=bioskey(0);
    if (key==0x3b00) goto lazo1; /*SENO*/
    if (key==0x3c00) goto lazo2; /*COSENO*/
    if (key==0x3d00) goto lazo3; /*RAMPA*/
    if (key==0x3e00) goto lazo4; /*CUADRADA*/
    if (key==0x3f00) goto lazo5; /*COMPUESTA*/
    if (key==0x11b)  goto fin;
    goto menu;
    return(0);
}
```

```

/*1°CUADRANTE*/
setcolor(WHITE);
rectangle(44,34,276,206);
setfillstyle(SOLID_FILL,WHITE);
floodfill(5,5,WHITE);
setcolor(BLACK);
setcolor(2);
settextstyle(SMALL_FONT,HORIZ_DIR,6);
sprintf(msg,"Senal de Entrada");
outtextxy(83,225,msg);
/*2°CUADRANTE*/
setcolor(WHITE);
rectangle(364,34,596,206);
setfillstyle(SOLID_FILL,WHITE);
floodfill(5,5,WHITE);
setcolor(BLACK);
setcolor(2);
sprintf(msg,"Senal Modul/Ent.");
outtextxy(403,225,msg);
/*3°CUADRANTE*/
setcolor(WHITE);
rectangle(44,274,276,446);
setfillstyle(SOLID_FILL,WHITE);
floodfill(5,5,WHITE);
setcolor(BLACK);
setcolor(2);
sprintf(msg,"Senal Modulada");
outtextxy(83,463,msg);
/*4°CUADRANTE*/
setcolor(WHITE);
rectangle(364,274,596,446);
setfillstyle(SOLID_FILL,WHITE);
floodfill(5,5,WHITE);
setcolor(BLACK);
setcolor(2);
sprintf(msg,"Senal Digital");
outtextxy(403,463,msg);
setcolor(1);
rectangle(80,0,215,20);
setfillstyle(SOLID_FILL,1);
floodfill(1,1,1);
rectangle(400,0,560,20);
floodfill(1,1,1);
setcolor(15);
sprintf(msg,"Esc Menu Ppal.");
outtextxy(83,0,msg);
sprintf(msg,"Enter Menu Simul.");
outtextxy(403,0,msg);
setcolor(2);
/*COSENO*/
x=y=x1=y1=0;
for (x=0;x<=11.75;x=x+0.01);
{
y=-35*cos(0.75*x);/*SEÑAL MENSAJE 1°CUADRANTE*/

```

```

setfillstyle(SOLID_FILL,WHITE);
floodfill(45,35,WHITE);
setcolor(BLACK);
setcolor(2);
settextstyle(SMALL_FONT,HORIZ_DIR,6);
sprintf(msg,"Senal de Entrada");
outtextxy(83,225,msg);
/*2°CUADRANTE*/
setcolor(WHITE);
rectangle(364,34,596,206);
setfillstyle(SOLID_FILL,WHITE);
floodfill(365,35,WHITE);
setcolor(BLACK);
setcolor(2);
sprintf(msg,"Senal Modul./Ent.");
outtextxy(403,225,msg);
/*3°CUADRANTE*/
setcolor(WHITE);
rectangle(44,274,276,446);
setfillstyle(SOLID_FILL,WHITE);
floodfill(45,276,WHITE);
setcolor(BLACK);
setcolor(2);
sprintf(msg,"Senal Modulada");
outtextxy(83,463,msg);
/*4°CUADRANTE*/
setcolor(WHITE);
rectangle(364,274,596,446);
setfillstyle(SOLID_FILL,WHITE);
floodfill(365,275,WHITE);
setcolor(BLACK);
setcolor(1);
rectangle(80,0,215,20);
setfillstyle(SOLID_FILL,1);
floodfill(81,1,1);
rectangle(400,0,560,20);
floodfill(401,1,1);
setcolor(15);
sprintf(msg,"Esc Menu Ppal.");
outtextxy(83,0,msg);
sprintf(msg,"Enter menu Simul.");
outtextxy(403,0,msg);
setcolor(2);
sprintf(msg,"Senal Digital.");
outtextxy(403,463,msg);
x=y=x1=y1=0; /*SENAL MUESTRA 1° CUADRANTE*/
for (x=0;x<=2;x=x+0.01);
{
y=-20*x+sin(x);
line(20*x+4,11+y,20*x+4,11+y);
x1=20*x;
y1=y;
}
x=y=0;

```

```

}
x=y=0;
for (x=0;x<=2;x=x+0.01);
{
y=20*x+sin(x);
line(x1+20*x+44+78,121+y+y1,x1+20*x+44+78,121+y+y1);
}
    x=y=x1=y1=0;
for (x=0;x<=2;x=x+0.01);
{
y=-20*x+sin(x);
line(20*x+34+16,11+y,20*x+34+16,11+y);
y1=y;
}
x=y=0;
for (x=0;x<=1.9;x=x+0.01);
{
y=20*x+sin(x);
line(x1+20*x+44+156,121+y+y1,x1+20*x+44+156,121+y+y1);
}
    setcolor(BLUE);/*SENAL MODULADA 2° CUADRANTE*/
x=y=x1=y1=a=b=0;
for (x=0;x<=2;x=x+0.05)
{
d=-20*x+sin(x);
y=5*sin(60*x);
if (y<0)
if (y>0)
{
if (b<d) e=1;
else e=-1;
b=a
c=0;
}
a=b;
line(20*x+364,121+a,20*x+364,121+a);
x1=20*x;
y1=y;
}
    x=y=0;
a=b;
for (x=0;x<=2;x=x+0.001)
{
d=0;
y=5*sin(60*x);
if (y<0) c=1;
y=y*c;
if (y>0)
{
if (b<d) e=1;
else e=-1;
b=a+e*4;
c=0;
}
}

```



```

{
y=-35*sin(0.75*x);/*SEÑAL MENSAJE 1°CUADRANTE*/
line(x1*20+4,y1+11,x*20+4,y+11);
}
setcolor(8);
x=y=x1=y1=0;
for (x=0;x<=11.75;x=x+0.01);
{
y=-35*sin(0.75*x);/*SEÑAL MENSAJE 2°CUADRANTE*/
line(x1*20+34,y1+11,x*20+34,y+11);
}
setcolor(RED);/*SEÑAL MODULADA 2 CUAD.*/*
x=y=x1=y1=a=b=0;
for(x=0;x<=11.75;x=x+0.05);
{
d=-35*sin(0.75*x);
y=5*sin(30*x);
if (y<0) c=1;
x=y=x1=y1=a=b=0;
for(x=0;x<=11.75;x=x+0.01);
{
d=-35*sin(0.75*x);
y=5*sin(30*x);
if (y<0) c=1;
y=y*c;
if (y>0)
{
else e=-1;
b=a+e
c=0;
}
a=b;
line(20*x+4,31+a,20*x+4,31+a);
}
setcolor (RED);/*SEÑAL DIGITAL 4°CUADRANTE*/
x=y=x1=y1=a=b=0;
for(x=0;x<=11.75;x=x+0.01);
{
d=-35*sin(0.75*x);
y=5*sin(30*x);
if (y<0) c=1;
y=y*c;
if (y>0)
{
if (b<d) e=1;
else e=-1;
c=0;
}
a=b;
line(20*x+34,32+e*c*15,20*x+34,32+e*c*15);
}

lazo2:
cleardevice();

```

"PROGRAMA PARA EJECUTAR EL MENÚ DE PRÁCTICAS"
"LENGUAJE TURBO C"

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
main()
{
    int i,k,key;
    int driver,mode;
    char msg[80];
    driver=VGA;
    mode=VGAHI;
    initgraph(&driver,&mode,"");

    menu:
        cleardevice();
        /*MENU*/
        setcolor(7);
        rectangle(150,90,500,380);
        setfillstyle(SOLID_FILL,7);
        floodfill(1,1,7);
        setcolor(9);
        rectangle(145,95,495,385);
        setfillstyle(INTERLEAVE_FILL,9);
        floodfill(6,6,9);
        setcolor(7);
        setttextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
        sprintf(msg,"MENU DE PRACTICAS");
        outtextxy(200,110,msg);
        sprintf(msg,"  SENOIDAL      <F1>");
        outtextxy(160,160,msg);
        sprintf(msg,"  COSENO      <F2>");
        outtextxy(160,210,msg);
        sprintf(msg,"  CUADRADA    <F3>");
        outtextxy(160,260,msg);
        sprintf(msg,"  Salir      <Esc>");
        outtextxy(160,310,msg);
        key=bioskey(0);
        if (key==0x7f00) goto lazo1; /*lee la tecla F1*/
        if (key==0x7g00) goto lazo2; /*lee la tecla F2*/
        if (key==0x7h00) goto lazo3; /*lee la tecla F3*/
        if (key==0x151b) goto fin; /*lee la tecla Esc*/
        goto menu;
        return(0);
    lazo1:
        cleardevice();
        setcolor(1);
        rectangle(70,20,600,460);
        setfillstyle(INTERLEAVE_FILL,3);
        floodfill(71,21,3);
        gotoxy(25,3);
        printf("Datos de la señal Senoidal");
```

```

    goto amplitud2;
}
amplitud2:
gotoxy(20,7);
printf("Amplitud(V) :");
if (scanf("%f",&))
    goto muestreo2;
else
{
    fprintf(stderr, "Error reding an \
        an integer from stdin.\n");
}
muestreo2:
gotoxy(20,9);
printf("Frec. de Muestreo(Hz):");
if (scanf("%f",&))
    goto delta2;
else
{
    fprintf(stderr, "error reading an \
        integer from stdin. \n");
}
delta2:
gotoxy(20,11);
printf("Delta (0-25) : ");
if (scanf("%f",&))
    goto tercero;
else
{
    fprintf(stderr, "error reading an \
        integer from stdin. \n");
}
tercero:
cleardevice();
setcolor (11);
rectangle(70,20,566,396);
setfillstyle(SOLID_FILL,11);
floodfill(101,41,11);
setcolor (2);
setcolor(15);
settextstyle(SMALL_FONT,HORIZ_DIR,6);
sprintf(msg,"0");
outtextxy(50,127,msg);
sprintf(msg,"7v");
outtextxy(50,57,msg);
setcolor(3);
rectangle(70,405,256,425);
setfillstyle(SOLID_FILL,3);
floodfill(71,406,3);
rectangle(380,405,566,425);
setfillstyle(SOLID_FILL,3);
floodfill(381,406,3);
setcolor(15);
sprintf(msg,"Esc. Menu Practicas");

```

```

outtextxy(72,407,msg);
sprintf(msg,"Ent. Otro Calculo");
outtextxy(382,407,msg);
    setcolor(1);
    x=y=x1=y1=a=b=0; /*SEÑAL MUESTRA 1° CUADRANTE*/
    for(x=0;x<=25;x=x+0.01);
    {
        y=15*sin((f/1000)*x);
        y1=0;
        if (y>0) y1=-(h*10);
        line(20*x+70,137+y1,20*x+70,137+y1);
        x1=x;
        y1=y;
    }
        setcolor (RED); /*SEÑAL MODULADA 2 CUADR.*/*
        x=y=x1=y1=a=b=0;
        for(x=0;x<=25;x=x+0.01);
    {
        y1=0;
        if (d<0) y1=-(h*10);
        if (y<0) c=1;
        y=y*c;
        if (y>0)
    {
        if (b<y1) e=1;
        else e=-1;
        c=0;
    }
        a=b;
        line(20*x+70,139+a,20*x+70,139+a);
        x1=x;
    }
    setcolor(9);
    x=y=x1=y1=a=b=0; /*SEÑAL DIGITAL 4° CUADRANTE*/
    for(x=0;x<=25;x=x+0.01);
    {
        y1=0;
        if (d<0) y1=-(h*10);
        y=5*sin((m/1000)*x);
        if (y<0) c=1;
        if (y>0)
        {
            else e=-1;
        }
        a=b;
        line(20*x+70.302+e*c*15,20*x+70.302+e*c*15);
        x1=x;
    }
    fin:
    closegraph();
}

```


BIBLIOGRAFÍA.

- DR. JAMES T. SMITH. ADVANCED TURBO C. Mc. Graw Hill. USA 1989.
- B.P. LATHI. SISTEMAS DE COMUNICACIÓN. Edición en Español. Nueva Editorial Interamericana. México 1986.
- LUIS JOYANES AGUILAR. TURBO PASCAL 6.0 A SU ALCANCE. Mc. Graw Hill. México 1993.
- VILLALTA OSCAR. SISTEMA DIDÁCTICO DE ENSEÑANZA DE TRANSMISIÓN DIGITAL DEL SEÑALES ANALÓGICAS. Tesis Tecnológico Electrónica. Universidad Don Bosco. 1993.
- GREG VOSS and PAUL CHUI. TURBO C++ DISKTUTOR. Segunda Edición. Mc. Graw Hill. U.S.A. 1991.
- ROBERT LAFORE. ASSEMBLY LANGUAGE PRIMER FOR THE IBM® PC & XT. A Plume/Waite Book, New American Library, New York and Scarborough, Ontario 1984.