22

2-i)

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ARAGON"

DESARROLLO DE UN IGUALADOR ADOPTIVO PARA LA TRANSMISION DE DATOS

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A :

JOSE MARTIN VILLEDA GUTIERREZ

San. Juan de Aragón, Edo. de México 1986.

ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES ARAGON
UNIDAD ACADEMICA.

UNIVERSIDAD NACIONAL
AUTONOMA DE
MEXICO

ING. CLAUDIO C. MERRIFIELD CASTRO
COORDINADOR DEL AREA DE INGENIERIA,
P R E S E N T E .

En relación a la solicitud de la profesora ING. MA. DE JESUS L. - ORTEGA LARA, de fecha 8 de septiembre del año en curso, por la que se comunica que el alumno JOSE MARTIN VILLEDA GUTIERREZ, de la carrera de INGENIERIA MECANICA ELECTRICA, ha concluído su trabajo de investigación intitulado "DESARROLLO DE UN IGUALADOR ADAPTIVO PARA LA TRANSMISION DE DATOS", y como el mismo ha sido revisado y aprobado por dicho asesor, se autoriza su impresión, así como la iniciación de los trámites correspondientes para la celebración del - Examen Profesional.

Sin otro particular, le reitero las seguridades de mi atenta consideración.

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPIRITU"
San Juan de Aragón. Edo., de México., septiembre 10 de 1986.
EL JEFE DE LA UNIDAD

LIC. ARTURO MUÑOZ COTA PEREZ.

c.c.p. Sra. Gloria Bech Germán. Jefa del Departamento de Servicios Escolares.
Asesor de Tesis.
Interesado.

AMCP'lfc.

INDICE

Página

INTRODUCCION

DESARROLLO DE UN IGUALADOR ADAPTIVO PARA LA TRANSMISION DE DATOS

Actualmente la transmisión de datos digitales está creciendo a una velocidad extremadamente alta y las señales analógicas son, cada vez con mayor frecuencia, convertidas al formato digital antes de ser transmitidas. Esto se debe al aumento acelerado, en todas las áreas de las comunicaciones, del uso de la tecnología digital y del procesamiento de datos de las computadoras.

En estas transmisiones digitales, cada uno de los pulsos transmitidos a través de canales analógicos reales presenta, al llegar al receptor, algún tipo de distorsión en su forma, ocasionada por las características de no linealidad en frecuencia del canal y por el ruido que se adhiere a los pulsos durante el paso a través de éste.

Uno de los tipos de distorsión más común es la causada por la coincidencia de las "colas" de pulsos sucesivos en la señal recibida, resultando de esto una considerable mezcla de energía de los pulsos entre los espacios de tiempo adyacentes. A este tipo de interferencia se le conoce con el nombre de interferencia intersimbólica (ISI), y es uno de los mayores obstáculos a superar en las transmisiones de datos de alta velocidad a través de canales de banda limitada.

Para compensar las características de distorsión introducidas por el canal se utiliza un filtrado adicional en el receptor, de manera que se minimice en lo posible la interferencia intersimbólica [2].

Cuando el canal de transmisión cambia notablemente sus características con el tiempo, es necesario que el filtro en el receptor tenga parámetros ajustables al valor requerido para las condiciones de compensación de las variaciones del canal. Tradicionalmente a éste tipo de filtros se les conoce como igualadores. Este término fué aplicado a los primeros filtros diseñados para compensar las características de amplitud de cables de transmisión. En 1935, Wiener y Lee diseñaron el filtro transversal que es uno de los filtros de parámetros variables más conocidos actualmente en el desarrollo de igualadores [6].

Hoy en día a cualquier filtro cuyo propósito es alguna clase de compensación es conocido con el nombre de igualador; y cuando el ajuste de sus parámetros lo lleva a cabo el igualador automáticamente es llamado igualador adaptivo.

Un tipo de canal de transmisión utilizado frecuentemente para la transmisión de datos es la red telefónica. La línea telefónica diseñada con un ancho de banda para señal de voz produce distorsión de amplitud que se presenta en forma de

Nota: La notación [N] indica la referencia N de la lista localizada al final de este trabajo.

interferencia entre símbolos, y esta distorsión es proporcional a la velocidad de transmisión. Además, la característica en frecuencia exacta del canal telefónico es desconocida y variante en el tiempo.

El objetivo principal de este trabajo es desarrollar un igualador del tipo adaptivo que ayude a minimizar la interferencia intersimbólica y contribuya a alcanzar un valor óptimo en la velocidad de transmisión de datos a través de un canal de banda limitada. Su principio de funcionamiento es un filtro transversal programado digitalmente en un microprocesador con estructura especialmente configurada para el procesamiento digital de señales. Para ello se ha seleccionado el microprocesador TMS32010 de la compañía Texas Instruments.

El desarrollo se inicia haciendo un estudio de un modelo de sistema de comunicación digital, destacando la función principal de cada uno de sus componentes. También se define el concepto de capacidad del sistema a partir de la cantidad de información contenida en una señal enviada por tal sistema. Posteriormente se analizan los procesos de muestreo y digitalización de señales analógicas para la obtención de sus correspondientes versiones digitales. Al final del capítulo 1 se presenta en forma breve el tema de distorsión lineal y ruido en los sistemas de comunicación digital.

En el capitulo 2 se da un bosquejo de las condiciones en que se presenta interferencia intersimbólica en una señal, asi como las caracteristicas del filtro transversal necesarias para minimizar esta interferencia.

En el capitulo 3 se detallan algunos algoritmos de igualación adaptiva haciendo una comparación de sus propiedades de velocidad de convergencia, complejidad, etc.; de entre estos algoritmos se elige el que se considera más adecuado para la aplicación deseada y se realiza su programación en lenguaje de alto nivel; las pruebas en computadora realizadas para los diferentes algoritmos analizados se presentan en el capitulo 4.

Una vez que éste funciona en forma satisfactoria se procede a la programación del algoritmo en el lenguaje ensamblador del procesador de señales TMS32010 y se ejecuta la simulación de funcionamiento en tiempo real del igualador adaptivo utilizando el sistema de emulación formado por las tarjetas EVM32010 y AIB32010.

CAPITULO 1

CARACTERISTICAS GENERALES DE TRANSMISION DIGITAL

Introducción

Entenderemos por comunicación el proceso por medio del cual la información se transfiere de un punto llamado fuente, a otro punto que es el destino. Un sistema de comunicación es la totalidad de mecanismos que proporcionan el enlace para la información entre fuente y destino [2].

Existen innumerables tipos de sistemas de comunicación, sin embargo todos ellos se pueden incluir en un modelo típico consistente de un transmisor, un medio de transmisión y un receptor. El análisis de estos componentes para un sistema de comunicaciones digitales es el primer tema a tratar en este capítulo. Además se estudiarán los conceptos de información y capacidad del sistema de transmisión y se hará un análisis del teorema de muestreo. Al final del capítulo se tratarán los temas de distorsión lineal y ruido en los sistemas de comunicación.

1.1 MODELO DE UN SISTEMA DE COMUNICACION DIGITAL

Generalmente un sistema completo de comunicación incluye un transmisor, un medio de transmisión y un receptor, el cual debe producir a su salida una réplica reconocible de la información de la entrada.

Los elementos básicos de un sistema digital de comunicaciones son ilustrados por el diagrama de bloques mostrado en la figura 1.1. La fuente de información genera mensajes los cuales son transmitidos al receptor. En general las características del mensaje dependen del tipo de fuente de información que lo produce. Por ejemplo, el mensaje puede ser una señal de audio o de video. Nos referiremos a tales señales como señales analógicas y las fuentes que las producen como fuentes analógicas.

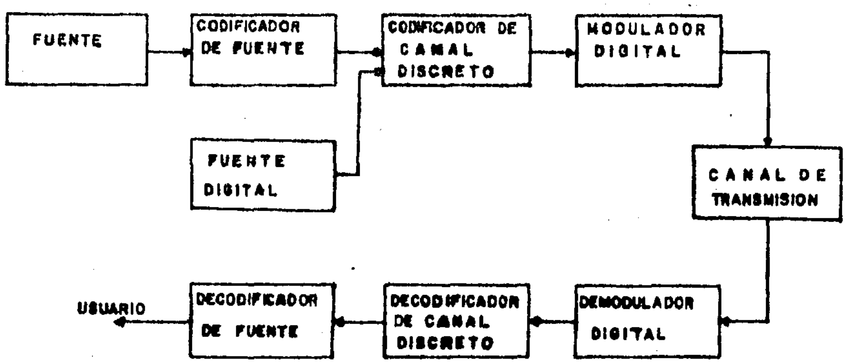


Figura 1.1 Diagrama de bloques de un sistema de comunicación digital.

En un sistema digital de comunicaciones, los mensajes producidos por la fuente son primero convertidos a forma digital, usualmente en una secuencia de digitos binarios (entendiendo por mensaje o señal binaria una secuencia de dos tipos de pulsos que se presentan a intervalos regularmente espaciados en el tiempo). El proceso de conversión eficiente de la salida de una fuente en una secuencia de digitos binarios es llamada codificación de fuente. En algunos casos la fuente de información puede ser de naturaleza digital, por tanto no es necesario la codificación de fuente. La secuencia de digitos binarios de la fuente codificada es transmitida a través de un canal al receptor.

En los canales continuos reales* utilizados en la transmisión (cable coaxial, fibra óptica, canal de radio, etc.) no es posible transmitir directamente la secuencia de digitos binarios de la fuente. Se requiere entonces un aparato que convierta la información digital en formas de onda que sean compatibles con las caracteristicas del canal. Tal aparato es llamado modulador digital o simplemente modulador.

---

* Un canal continuo es aquel canal de banda limitada en el cual los mensajes se presentan como formas de onda, es decir, funciones continuas del tiempo, y los parámetros apropiados son el ancho de banda B y la relación señal a ruido S/N.

Generalmente los canales reales no se comportan idealmente y por tanto presentan caracteristicas de respuesta en frecuencia no ideales asi como adición de disturbaciones de ruido e interferencia que corrompen la señal transmitida a través del canal. Tales disturbaciones causan error en la secuencia digital recibida.

En el extremo receptor del sistema de comunicación digital el demodulador procesa la forma de onda transmitida corrompida por el canal y reduce cada forma de onda a un solo número que representa una estimación del simbolo del dato transmitido. Esta estimación de la señal binaria obtenida a la salida del demodulador se alimenta al decodificador, el cual hace la reconstrucción de la secuencia de información original mediante el reconocimiento del código del codificador de canal.

Una evaluación del rendimiento del modulador y el decodificador es la frecuencia con la cual ocurren errores en la secuencia decodificada. Más precisamente, el promedio de probabilidad de error de bit a la salida del decodificador es una medida del rendimiento de la combinación demodulador-decodificador.

Como un paso final, cuando se desea una salida analógica, el decodificador de fuente acepta la secuencia de salida del decodificador de canal y, reconociendo el método de codificación de fuente usado reconstruye la señal de la fuente original. A la salida del decodificador de fuente se obtiene, por tanto, una aproximación de la salida de la fuente original.

En general, la probabilidad de error es una función de las características del código utilizado, los tipos de forma de onda empleadas al transmitir la información sobre el canal, la potencia de transmisión, las características del canal, la naturaleza del ruido y la interferencia en el canal [3].

1.2 CAPACIDAD DEL SISTEMA

El fundamento matemático para comunicaciones digitales fue establecido por Claude Shannon (1948) y Norbert Wiener (1949) de modo que tomadas juntas las ideas de Shannon y Wiener se obtienen los cimientos de la moderna teoría de la comunicación [1].

Shannon encontró que la probabilidad de que ocurra un error en la información transmitida puede idealmente reducirse tanto como se desee por medio de una adecuada codificación de la señal de entrada siempre que la velocidad de señalización binaria, en bits por segundo, sea menor que un número especificado que se determina por medio de la potencia del transmisor, el ruido del canal y la respuesta de tiempo del canal o ancho de banda.

Si se trata de introducir demasiados bits por segundo en un determinado canal, la cantidad de errores comenzará a aumentar rápidamente. La máxima velocidad de transmisión de señales por el canal se conoce con el nombre de capacidad del canal. Se ha elegido la palabra canal para expresar el medio físico donde la transmisión tiene lugar. Muchos autores incluyen en el canal varias otras partes del transmisor y también del receptor. Para evitar confusiones, se usará el término capacidad del sistema ya que todas las partes constituyentes del transmisor y del receptor contribuyen a determinar esta capacidad.

Para poder obtener la expresión matemática que expresa cuantitativamente la capacidad del sistema es conveniente primero analizar el concepto de información.

1.2.1 Información de un mensaje

La medida de la información es una indicación de la libertad de elección ejercida por la fuente en la selección de un mensaje. Si la fuente puede escoger libremente de entre muchos mensajes diferentes, el usuario tendrá una gran incertidumbre acerca de cual será el mensaje seleccionado. Pero si no hay posibilidad de elección, si solo hay un mensaje posible, no hay incertidubre y, en consecuencia, tampoco información. Sea que se prefiera el punto de vista de la incertidumbre o la interpretación de la libertad de elección, es evidente que la medida de la información incluye a las probabilidades. Los mensajes de gran probabilidad de elección de parte de la fuente conducen poca información y viceversa. Esta noción se formaliza definiendo a la información en términos de probabilidades.

Consideremos X y Y como dos variables aleatorias discretas con posibles salidas x_i, $i=1,2,...,n$ y y_j, $j=1,2,...,m$ respectivamente. Suponiendo que observamos alguna salida $Y=y_j$ y deseamos determinar cuantitativamente la cantidad de información que la ocurrencia del evento $Y=y_j$ provee acerca del evento $X=x_i$, observamos que cuando X y Y son estadisticamente independientes, la ocurrencia de $Y=y_j$ no provee información acerca de la ocurrencia del evento $X=x_i$. En caso contrario, cuando X y Y son totalmente dependientes, la ocurrencia de $Y=y_j$ determina la ocurrencia de $X=x_i$, el contenido de información es simplemente el provisto por el evento $X=x_i$. Por tanto, el contenido de información provista por la ocurrencia del evento $Y=y_j$ acerca del evento $X=x_i$ es definida como

$$I(x_i;y_j) = \log \frac{P(x_i/y_j)}{P(x_i)} \qquad (1.2.1)$$

$I(x_i;y_j)$ es entonces llamada la información mutua entre x_i y y_j. [3].

Consideremos ahora un canal discreto* que tiene un alfabeto de entrada

$$X=\{x_0,x_1,\ldots,x_{q-1}\}$$

un alfabeto de salida

$$Y=\{y_0,y_1,\ldots,y_{Q-1}\}$$

y el conjunto de probabilidades de trancisión $P(y_i/x_j)$ de la forma

$$P(Y=y_i/X=x_j) \equiv P(y_i/x_j) \qquad \begin{array}{l} i=0,1,\ldots,Q-1 \\ j=0,1,\ldots,q-1 \end{array}$$

las caracteristicas del canal determinan las probabilidades de trancisión $P(y_i/x_j)$, pero las probabilidades de los simbolos de entrada están bajo el control del canal discreto.

1.2.2 Capacidad del Canal

El valor de $I(X;Y)$ maximizado sobre el conjunto de probabilidades de simbolos de entrada $P(x_j)$ es una cantidad que depende de las caracteristicas del canal a través de las probabilidades condicionales $P(y_i/x_j)$. Esta cantidad máxima de $I(X;Y)$ es llamada capacidad del canal y denotada por C. Esto es, la capacidad del canal discreto es definida como

---

*Un canal discreto es aquel que transmite información en forma sucesiva, suponiendo diferentes estados eléctricos, frecuencia instantánea, disjuntos niveles de voltaje, etc.

$$C = \max_{P(x_j)} I(X;Y) = \max_{P(x_j)} \log \frac{P(x_i/y_j)}{P(x_i)} \qquad (1.2.2)$$

Las unidades de C son bits por símbolo de entrada en el canal cuando el logaritmo es base 2 y nats cuando el logaritmo es base e.

Para el caso de un canal continuo de banda limitada W con ruido blanco gaussiano aditivo (AWGN), la capacidad del canal por unidad de tiempo es definida formalmente como

$$C = \lim_{T\to\infty} \max_{p(x)} \frac{1}{T} I(X;Y) \qquad (1.2.3)$$

Además, la potencia promedio en x(t) es

$$P_{av} = \frac{1}{T} \int_0^T x^2(t)\,dt \qquad (1.2.4)$$

resolviendo la integral y considerando que la varianza de cada x_i es $\sigma_x^2 = TP_{av}/N$, entonces se obtiene que

$$\max_{p(x)} I(X;Y) = WT \log\left(1 + \frac{P_{av}}{WNo}\right) \qquad (1.2.5)$$

Finalmente la capacidad del canal por unidad de tiempo es obtenida como

$$C= W \log (1 + \frac{P_{av}}{WNo}) \qquad (1.2.6)$$

La ecuación (1.2.6) es la fórmula básica para la capacidad del canal continuo de banda limitada con AWGN y potencia promedio limitada a la entrada, y es conocida como la ley de Hartley-Shannon [2]. Entonces es posible transmitir hasta C bits por segundo por éste canal, obteniendose a la salida una cantidad de error tolerable.

Considerese la expresión de la capacidad (1.2.6) aplicada a un canal sin limites en el ancho de banda. Es evidente que si P_{av}/WNo se mantiene constante la capacidad del canal es directamente proporcional al ancho de banda. De modo que incrementando el ancho de banda podría obtenerse un mejoramiento de la capacidad de transmisión.

En particular, para un W grande, la capacidad se incrementa y tiende al valor limite C dado por

$$C_{\infty} = \frac{P_{av}}{No} \log_2 e = \frac{P_{av}}{No \ Ln\ 2}$$

$$= \frac{P_{av}}{.693} \qquad \text{bits/seg}$$

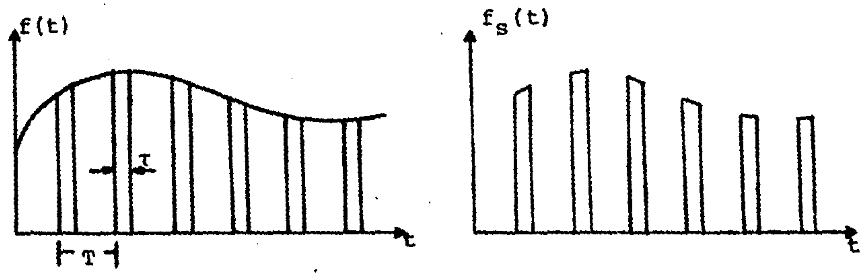
Entonces existe un limite para la velocidad a la que puede transmitirse libre de error por un canal de potencia limitada, donde se supone que el ancho de banda permisible es tan grande como se quiera. Aunque la transmisión a la velocidad de C_∞ bits/s no se puede obtener, la transmisión a la mitad de esta velocidad es un logro real [1].

1.3 MUESTREO Y DIGITALIZACION

Algunas señales de comunicación son digitales por naturaleza -por ejemplo, datos de teletipo, salidas de computadoras, señales pulsantes de radar, etc.- y otras son analógicas o funciones continuas en el tiempo. Si estas señales se van a transmitir en forma digital, deben primero ser muestreadas en forma periódica y posteriormente convertidas a muestras de amplitud discreta por medio de la cuantización.

1.3.1 Teorema de muestreo

Para comprender la operación de muestreo de una señal analógica primero consideremos una señal f(t) que varía continuamente, la cual se desea convertir a la forma digital. Esto se logra al muestrear en primer lugar a f(t) a una velocidad de fc muestras por segundo como se ilustra en la figura 1.2(a).



(a) Señal de entrada

(b) Salida muestreada

Figura 1.2 Proceso de muestreo (τ= tiempo de muestreo; T= 1/fc=intervalo de muestreo).

Se supondrá en primer lugar que la señal de entrada f(t) es de banda limitada a B Hertz, lo cual quiere decir que se encuentra libre de componentes de frecuencia por encima de f=B, aunque se sabe que con excepción de algunos casos especiales las señales reales contienen componentes de frecuencia para todas las frecuencias, sin embargo el contenido de frecuencia de las señales decae rápidamente después del ancho de banda definido, por lo tanto, la aproximación de limitación de banda de las señales reales no introduce un error significativo en el análisis.

Con la señal f(t) limitada en banda a B Hertz, no se destruirá ningún contenido de información al realizar el muestreo, siempre que la velocidad de muestreo $f_c \geqslant 2B$, lo que es posible demostrar mediante el análisis de Fourier de la siguiente forma

Como primer paso debemos introducir una función de conmutación S(t) tal que

$$f_s(t) = f(t)\, S(t) \qquad (1.3.1)$$

donde S(t) es un tren de pulsos periódicos de amplitud A_m, ancho τ y período $T = 1/fc$, como se muestra en la figura 1.3.

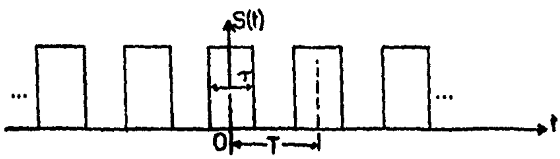


Figura 1.3 Función de conmutación

La función de conmutación periódica S(t) puede desarrollarse en su serie de Fourier, considerando $\omega_c = 2\pi n f_c$, como

$$C_n = \int_{-T/2}^{T/2} f(t)\, e^{-j\omega_c t}\, dt = \tau A_m \frac{\operatorname{sen} \omega_c \tau/2}{\omega_c \tau/2}$$

la expresión anterior generalmente se escribe en forma simplificada como

$$C_n = \tau A_m \operatorname{senc} x \qquad (1.3.2)$$

donde $\operatorname{senc}(x) = (\operatorname{sen} x)/x$ y $x = (\omega_c \tau)/2$. La gráfica de la función senc x se muestra en la figura 1.4.

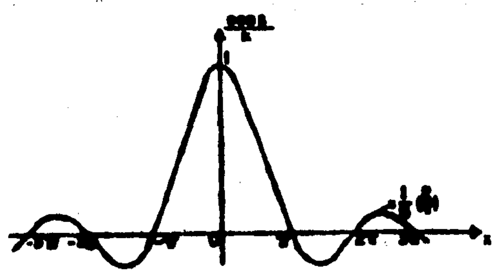


Figura 1.4 Función senc x= (sen x)/x

Por lo tanto, de la ecuación (1.3.2) se obtiene

$$S(t) = \frac{1}{T} \sum_{n=-\infty}^{\infty} C_n \, e^{j\omega_c t} \qquad (1.3.3)$$

como S(t) es una función par (simétrica respecto al eje t=0), además considerando Am=1 y que senc(0)=1 (como se observa en la figura 1.4) se puede reescribir la ecuación (1.3.3) como

$$S(t) = \frac{\tau}{T}\left[1 + 2\sum_{n=1}^{\infty} \text{senc } x \, e^{j\omega_c t}\right]$$

$$= d\left[1 + 2\sum_{n=1}^{\infty} \text{senc } \pi nd \cos 2\pi n f_c t\right] \qquad (1.3.4)$$

donde $d=\tau/T$ es llamado el ciclo de trabajo (duty cycle).

Sustituyendo S(t) de (1.3.4) en la ecuación (1.3.1) tenemos

$$f_s(t) = f(t)\, d\left[1 + 2\sum_{n=1}^{\infty} \text{senc}\ \pi n d \cos 2\pi n f_c t\right] \qquad (1.3.5)$$

debe notarse que el primer término de la ecuación (1.3.5) es el mensaje f(t) atenuado por el ciclo de trabajo d.

Para recuperar la señal original de la señal muestreada, solo será necesario quitar las componentes de frecuencia fuera de la que está en ω_m (que es la señal original $A\cos\omega_m t$ multiplicada por d). Esto se puede llevar a cabo con un filtro pasa-bajas, cuya frecuencia de corte W cumpla con la siguiente condición: $\omega_m < W < \omega_c - \omega_m$. Esto demuestra la posibilidad de recuperar f(t) sin distorsión a partir de su versión muestreada $f_s(t)$. Los sistemas que transmiten éstos valores muestreados de la señal f(t) se denominan comunmente sistemas de datos muestreados o de modulación de pulsos.

Consideremos ahora que la señal a muestrear tiene un espectro de banda limitada con una frecuencia máxima f_m como muestra la figura 1.5(a). Para extraer la señal original a partir de la onda muestreada, la primer banda lateral no debe superponerse a la segunda banda lateral, es decir

$$f_m \leqslant f_c - f_m$$

por tanto se debe cumplir

$$f_c \geqslant 2\, f_m \tag{1.3.6}$$

entonces la frecuecia de muestreo f_s debe ser por lo menos el doble de la máxima frecuecia de señal, para permitir la reconstrucción de la señal por filtrado. Este principio importante es denominado Teorema del Muestreo.

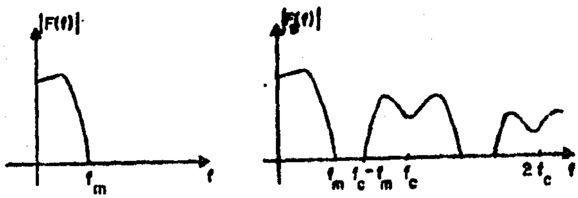
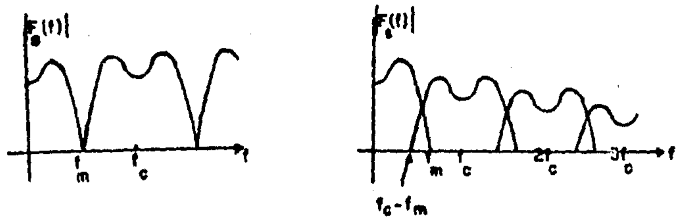


Figura 1.5 Espectros de una señal y su correspondiente muestreada.

La frecuencia mínima de muestreo es llamada **velocidad de muestreo de Nyquist.**

Si muestreamos la señal de la figura 1.5 a una velocidad menor que la velocidad de muestreo de Nyquist obtendremos los espectros indicados en la figura 1.6. El resultado es que las frecuencias que normalmente están fuera de la banda de frecuencias de la señal han sido desplazadas dentro de la banda de frecuencias y la señal será distorsionada. Este efecto es llamado interferencia de colas espectrales (aliasing) [4].



(a) (b)

Figura 1.6 Muestreo a velocidad (a) igual a la velocidad de muestreo de Nyquist; (b) menor que la velocidad de muestreo de Nyquist.

Ya que los filtros pasa-bajos en la práctica no son ideales y no tienen un punto de corte abrupto, la velocidad de muestreo necesaria en un sistema práctico es mayor que la velocidad de muestreo de Nyquist.

1.3.2 Cuantización y codificación

Antes de que las muestras sean transmitidas, éstas deben ser digitalizadas y codificadas. Los sistemas que implican la transmisión de señales digitalizadas y codificadas se denominan sistemas de modulación por codificación de pulsos (PCM), siendo los más comunes de estos los sistemas digitales binarios.

El proceso de digitalización de señales originalmente analógicas se conoce como cuantización, el cual consiste en la subdivisión de las amplitudes de las muestras en un predeterminado número de niveles discretos de amplitud.

Obviamente, el proceso de cuantización introduce algunos errores durante la reproducción de la señal. El efecto total es similar al que resultaría si se hubiera agregado un ruido adicional al sistema. Este ruido de cuantización puede reducirse disminuyendo la separación de niveles o aumentando el número de niveles empleados.

Si se envían las señales cuantizadas a diferentes niveles el sistema resultante es un sistema de modulación por amplitud de pulsos (PAM).

Después de que la señal ha sido muestreada y cuantizada es común que se codifique en un grupo equivalente de pulsos binarios de igual amplitud, obteniendo así una señal binaria. El número de dígitos binarios utilizados dependerá del número de niveles de cuantización usados. Como los dígitos binarios deben de ser transmitidos en el intervalo de muestreo dispuesto originalmente para una muestra cuantizada, la duración de los pulsos es menor y en consecuencia el ancho de banda crece, por tanto normalmente se escogen tan pocos niveles de cuantización como sea posible.

Hay, sin embargo casos en que es necesario tener más exactitud en algún rango de niveles de las muestras, es por esto que se requiere una distribución de niveles no uniforme que se presente más comprimida en unas lecturas que en otras. En la práctica, en vez de usar niveles de cuantización no uniforme para adaptar las características de la señal, éstas son comprimidas en amplitud, para obligar a que todas las señales caigan dentro de un intervalo específico, siendo la más común la característica de compresión de tipo logarítmico. El efecto producido es que para los niveles inferiores de la señal se proporcionan mas niveles de cuantización. En el receptor la señal se expande a su amplitud original por medio de la curva inversa del logaritmo. La combinación de los procesos de compresión y expansión se conoce con el nombre de compansión [1].

1.4 DISTORISION LINEAL

Se entiende por distorsión la alteración de la señal transmitida debida a la respuesta imperfecta del sistema a ella misma. En la práctica, todos los sistemas de comunicacion reales presentan una cierta cantidad de distorsión.

De manera más precisa, dada una señal de entrada x(t), se dice que la salida es sin distorsión si difiere de la entrada solo en una constante multiplicadora y en un retardo finito de tiempo. En forma analítica, se tiene una transmisión sin distorsión si

$$y(t) = kx(t-t_d) \tag{1.4.1}$$

donde k y t_d son constantes. El espectro correspondiente a ésta señal es

$$Y(f) = k\, e^{-j\omega t_d}\, X(f)$$

y la función de transferencia del sistema es por tanto

$$Y(f) = H(f)\, X(f)$$

tal que

$$H(f) = K\, e^{-j\omega t_d} \tag{1.4.2}$$

esto indica que en una transmisión sin distorsión se tiene una respuesta de amplitud constante y un corrimiento de fase lineal negativo. Estas características se deben cumplir para las frecuencias en las cuales el espectro de la señal de entrada es diferente de cero.

Para un estudio adecuado, la distorsión se ha clasificado en tres tipos principales que son:

1.- Distorsión de amplitud

2.- Distorsión de fase

3.- Distorsión no lineal

En este trabajo estudiaremos únicamente los dos primeros casos que estan dentro de la categoria de distorsión lineal.

1.4.1 Distorsión de amplitud

Se dice que una señal de salida esta distorsionada en amplitud cuando sus componentes de frecuencia no estan en la proporsión correcta. Esto indica que abs[H(f)] no es constante con la frecuencia, por tanto algunas veces se le designa como distorsión de frecuencia.

Las formas mas comunes de distorsión de amplitud son atenuación excesiva o levantamiento de los extremos de las altas o bajas frecuencias en el espectro de la señal. Otra forma de distorsión, aunque menos común es una respuesta desproporcionada a una banda de frecuencia dentro del espectro. En la figura 1.7 se muestra el efecto de la distorsión de amplitud y de fase sobre un pulso cuadrado.

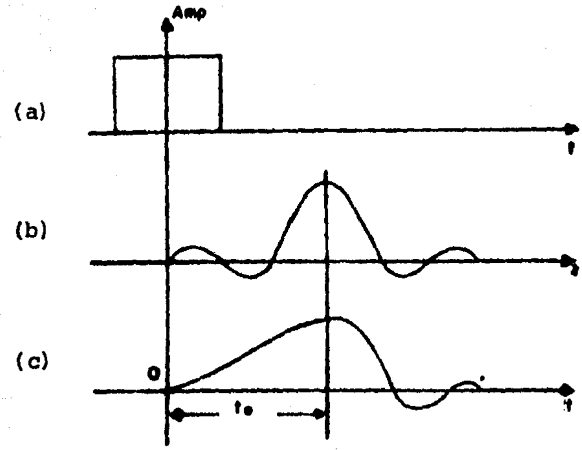


Figura 1.7 Distorsión de amplitud y fase provocada por la red. (a) Pulso de entrada. (b) Respuesta de salida con distorsión de amplitud, corrimiento lineal de fase (simétrico). (c) Distorsión de fase (no hay simetría).

1.4.2 Distorsión de fase

Un corrimiento de fase lineal ocaciona un retardo de tiempo constante para todas las componentes de frecuencia de la señal. Así, con una respuesta de amplitud constante, se tiene una salida sin distorsión. Si el corrimiento de fase no es lineal, las diferentes componentes de frecuencia sufren diferentes retardos de tiempo, y a la distorsión resultante se le designa distorsión de fase o distorsión por retardo.

Para un corrimiento de fase arbitrario, el retardo en tiempo es una función de la frecuencia y será independiente de la frecuencia solo si [H(f)] es lineal con ella.

Debe notarse que los valores de pico de la señal desplazada pueden ser diferentes que los de la señal de entrada, sin embargo, esta diferencia no es debida a la respuesta de amplitud, esto resulta así porque las componentes de la señal distorsionada alcanzan valores máximos o mínimos en tiempos diferentes a como se alcanzaban en la señal de entrada, provocando aumento o disminusión de los valores pico y otras alteraciones en la forma de onda [2].

El problema de la distorsión lineal es, en forma general, remediable con el uso de redes de igualación. Este tema se tratará en los capítulos posteriores cuando se estudie el igualador de linea de retardo variable o filtro transversal.

1.5 Ruido en los sistemas de transmisión digital

Durante la transmisión, las formas de onda sufren los efectos de ruido del canal. En los sistemas digitales, cuando la señal llega al receptor es necesario decidir cual de las n formas de onda conocidas posibles se ha transmitido. Una vez adoptada tal decisión, se recupera la onda transmitida sin ningún ruido. En este sentido, el ruido del canal no ejerce influencia alguna; sin embargo puede provocar algún error en las decisiones, de modo que algunas de ellas serán equivocadas y la probabilidad de error crecerá con el incremento de ruido.

Para un sistema binario con pulsos s(t) de duración T y espacios (ausencia de pulsos) también de duración T, el detector debe examinar el contenido de la señal de entrada cada T segundos y decidir si hay pulso presente o no. La decisión se puede facilitar si pasamos la señal por un filtro que acentúe la señal útil s(t) y suprima al mismo tiempo el ruido n(t). El filtro que realiza esto se conoce como filtro acoplado.

Sea s(t)+n(t) la señal de entrada a un filtro acoplado, en donde s(t) es la señal útil, n(t) es el ruido del canal y $s_o(t)+n_o(t)$ es la salida del filtro. Nótese que $n_o(t)$ es una señal aleatoria y no puede determinarse exactamente, por lo tanto es conveniente tomar su valor cuadrático medio $n_o^2(t)$. Así, se deseará optimizar la razón ρ en algún instante t=tm (instante de efectuar la decisión) dada por

$$\rho = \frac{s_o^2(tm)}{n_o^2(tm)} \qquad (1.5.1)$$

Sea $S(\omega)$ la transformada de Fourier de $s(t)$ y $H(\omega)$ la función de transferencia del filtro acoplado; entonces,

$$s_o(t) = \mathcal{F}^{-1}[S(\omega)\ H(\omega)]$$

y

$$s_o(tm) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) S(\omega)\ e^{j\omega tm}\ d\omega \qquad (1.5.2)$$

Si $S_n(\omega)$ es el espectro de densidad de potencia de la señal de ruido $n(t)$, entonces $|H(\omega)|^2 S_n(\omega)$ es el espectro de densidad de potencia de $n_o(t)$. Por lo tanto,

$$\overline{n_o^2(tm)} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S\ (\omega) |H(\omega)|^2\ d\omega \qquad (1.5.3)$$

y sustituyendo (1.5.3) y (1.5.2) en (1.5.1).

$$\rho = \frac{\overline{s_o^2(tm)}}{\overline{n_o^2(tm)}} = \frac{\left|\int_{-\infty}^{\infty} H(\omega) S(\omega) e^{j\omega tm}\, d\omega\right|^2}{\pi\gamma \int_{-\infty}^{\infty} |H(\omega)|^2\, d\omega} \qquad (1.5.4)$$

en que

$$S\,(\omega) = \frac{\gamma}{2}$$

es la densidad de potencia.

Empleando la desigualdad de Schwartz, (1.5.4) se puede expresar como

$$\left|\int_{-\infty}^{\infty} H(\omega) S(\omega) e^{j\omega tm} d\omega\right|^2 \leqslant \int_{-\infty}^{\infty} |H(\omega)|^2 d\omega \int_{-\infty}^{\infty} |S(\omega)|^2 d\omega \qquad (1.5.5)$$

sustituyendo (1.5.5) en (1.5.4) se obtiene

$$\frac{\overline{s_o^2(tm)}}{\overline{n_o^2(tm)}} \leqslant \frac{1}{\pi\gamma} \int_{-\infty}^{\infty} |S(\omega)|^2 d\omega$$

por tanto

$$\rho_{max} = \frac{1}{\pi\gamma} \int_{-\infty}^{\infty} |S(\omega)|^2 d\omega \qquad (1.5.6)$$

y como la energía E de la señal s(t) está dada por

$$E= \int_{-\infty}^{\infty} s^2(t)dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |S(\omega)|^2 d\omega$$

entonces

$$\rho = \frac{\overline{s_o^2(tm)}}{\overline{n_o^2(tm)}} = \frac{E}{\gamma/2} = \frac{2E}{\gamma}$$

$$= \frac{\text{energia de la se~al s(t)}}{\text{espectro de densidad de potencia de la se~al de ruido de entrada}} \qquad (1.5.7)$$

Es posible que la señal s(t) esté presente en la entrada, pero $n_o(t)$ tenga un valor negativo grande; esto haría la salida r(t) muy grande, causando que la salida r(t) sea grande. Debemos entonces encontrar una regla de decisión que minimice la probabilidad de error.

Una regla de decisión posible es: "presencia de señal" si r(T)>a y "ausencia de señal" si r(T)<a, siendo "a" el valor de umbral óptimo que minimiza la probabilidad de error de decisión. Pero debido al ruido, existen momentos en que r(T)>a aún en ausencia de señal; así como también r(T) pequeños en presencia de ésta. Por lo tanto, para un umbral dado, cometemos dos clases diferentes de error, el de señal presente y el de señal ausente.

Si la señal s(t) tiene la misma probabilidad de estar presente que ausente, entonces la probabilidad de error es mínima si elegimos el umbral.

$$a = \frac{E}{2} \tag{1.5.8}$$

Como el ruido es una señal aleatoria con amplitudes de distribución gaussiana, su función de densidad de probabilidad de la amplitud x está dada por

$$p(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-x^2/2\sigma_x^2} = \frac{1}{\sqrt{\pi\gamma E}} e^{-x^2/\gamma E} \tag{1.5.9}$$

en donde σ_x^2 es el valor cuadrático medio de la señal, que para $n_o(t)$ está dado por $\gamma E/2$.

Haciendo x=salida=r(T) en (1.5.9), y definiendo la probabilidad de error $P(\varepsilon)$ como

$$P(\varepsilon)= \int_a^\infty p(r)dr$$

entonces

$$P(\varepsilon)= \frac{1}{\sqrt{\pi\gamma E}} \int_a^\infty e^{-r^2/\gamma E}dr \tag{1.5.10}$$

y para la función de error

$$\operatorname{erf}(x)= \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2}\ dy \tag{1.5.11}$$

es posible expresar (1.5.10) en la forma

$$P(\varepsilon)= \operatorname{erf}\left(\frac{a}{\sqrt{\gamma E/2}}\right) \tag{1.5.12}$$

pero como a=E/2

$$P(\varepsilon) = \operatorname{erfc}\left(\sqrt{\frac{E}{2\gamma}}\right) \qquad (1.5.13)$$

Así, para N decisiones realizadas, el número total de decisiones incorrectas N_ε está dado por

$$P(\varepsilon) = \frac{N_\varepsilon}{N} \quad \rightarrow \quad N_\varepsilon = P(\varepsilon)N$$

Entonces, si $P(\varepsilon)=1/100$ en promedio por ejemplo, una de cada 100 decisiones será incorrecta [5].

CAPITULO 2

INTERFERENCIA INTERSIMBOLICA E IGUALACION LINEAL

Introducción

En los sistemas de comunicaciones digitales se desea transmitir a altas velocidades sobre canales de banda limitada W. Cuando el canal es ideal para f<W, una señal de pulsos g(t) puede ser diseñada permitiendonos transmitir a una velocidad igual (o superior) al ancho de banda W del canal. Se encontrará, en efecto, que hasta 2W pulsos por segundo pueden ser transmitidos por un canal cuyo ancho de banda es W hertz. Esta es llamada la velocidad de Nyquist y obviamente está relacionada con la velocidad de muestreo de Nyquist [1]. En el caso de que el canal no sea ideal, la señal de transmisión a una velocidad de símbolo igual o superior a W resulta en interferencia intersimbólica.

Este tipo de interferencia, así como los métodos empleados para disminuirla tales como la igualación, son los temas a tratar en este capítulo.

2.1 Análisis de Interferencia Intersimbólica

La interferencia intersimbólica es provocada por el efecto de filtrado del sistema, el cual ocaciona el ensanchamiento de los pulsos a medida que atraviesan la red. Tal ensanchamiento en los pulsos da origen al traslapamiento entre símbolos en los instantes de tiempo adyacentes y por tanto a una considerable mezcla de energía, la cual produce confusión en la interpretación de los símbolos y posibles errores a la salida del sistema.

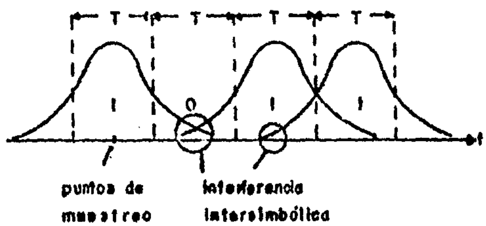


Figura 2.1 Interferencia entre símbolos en la transmisión digital

Esta interferencia puede minimizarse ampliando adecuadamente el ancho de banda de la transmisión tanto como sea debido, sin embargo de esto resulta un innecesario gasto del ancho de banda, además, si se exagera puede introducirse mucho ruido en el sistema.

En lugar de ello, se buscará una manera de diseñar las formas de onda y por tanto los filtros de transmisión utilizados con el objeto de minimizar o eliminar esta interferencia con el menor ancho de banda posible.

El equivalente pasa-bajos de la señal transmitida para diferentes tipos de técnicas de modulación digital tiene la forma común

$$u(t) = \sum_{n=o}^{\infty} In\ g(t-nT) \qquad (2.1.1)$$

donde {In} es una secuencia de M niveles formados al tomar bloques de K= $\log_2 M$ digitos binarios y mapeando cada bloque en uno de los M niveles o amplitudes. La señal g(t) es un pulso básico cuya selección (forma) la determina el filtro transmisor G_T y constituye un importante problema en el diseño de señales cuando hay limitación en el ancho de banda del canal.

Esta señal es transmitida sobre un canal que tiene respuesta en frecuencia $C(\omega)$ también limitada a $|f|<W$, como se observa en la figura 2.2.

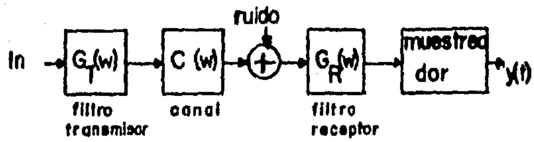


Figura 2.2 Sistema PAM con muestreador a 1/T muestras por segundo

Consecuentemente la señal recibida puede ser representada como

$$r(t) = \sum_{n=0}^{\infty} I_n\, h(t-nT) + z(t) \tag{2.1.2}$$

donde $h(t) = \int_{-\infty}^{\infty} g(\tau) c(t-\tau) g\ (\tau)\ d\tau$

y z(t) representa el ruido blanco gaussiano adicionado. Supongamos que la señal recibida es pasada primero a través de un filtro y después muestreada a una velocidad de 1/T muestras/seg. Denotamos la salida del filtro receptor como

$$y(t) = \sum_{n=0}^{\infty} In\ x(t-nT) + v(t) \qquad (2.1.3)$$

donde x(t) es el pulso de entrada h(t) representado a la salida del filtro receptor y v(t) es la respuesta del filtro receptor al ruido z(t).

Si y(t) es muestreada a tiempos $t = kT + \tau_o$, K=0,1,.... tenemos

$$y(kT+\tau o) \equiv y_k = \sum_{n=0}^{\infty} In\ x(kT-nT+\tau o) + v(kT+\tau o)$$

$$= \sum_{n=0}^{\infty} In\ x_{k-n} + v_k \quad k=0,1,.. \qquad (2.1.4)$$

donde τo es el retardo de la transmisión a través del canal. El valor muestreado puede ser expresado como

$$y_k = x_o(I_k + \frac{1}{xo}\sum_{\substack{n=o\\n\neq k}}^{\infty} In\, x_{k-n}) + v_k \quad k=0,1,.. \qquad (2.1.5)$$

xo representa la ganancia (o atenuación) de la señal al pasar a través del sistema y que, por conveniencia, lo consideraremos unitario. Entonces

$$y_k = I_k + \sum_{\substack{n=o\\n\neq k}}^{\infty} In\, x_{k-n} + v_k \qquad (2.1.6)$$

el término I_k representa el símbolo de información deseado en el k-ésimo instante muestreado y, los términos segundo y tercero de la ecuación (2.1.6) representan respectivamente la interferencia intersimbólica y el ruido gaussiano en el k-ésimo instante muestreado.

La interferencia intersimbólica proviene de la sobreposición de las colas de otros pulsos adicionadas al pulso particular $I_k g(t-kT)$ el cual es examinado en el k-ésimo instante muestreado. Cada término de interferencia es proporcional a una muestra de la respuesta impulso del canal espaciados un múltiplo iT del intervalo de símbolo T [3].

En forma práctica, la distribución de interferencia intersimbólica y ruido es fácilmente vista analizando el voltaje recibido y(t) con un osciloscópio cuya frecuencia de barrido horizontal sea igual a 1/T. La figura resultante en la pantalla del osciloscópio es conocida como diagrama de ojo [6]. Para comprender la interpretación de los diagramas de ojo examinemos la figura 2.3 donde se muestran dos formas de onda binarias, una distorsionada y otra no distorsionada.

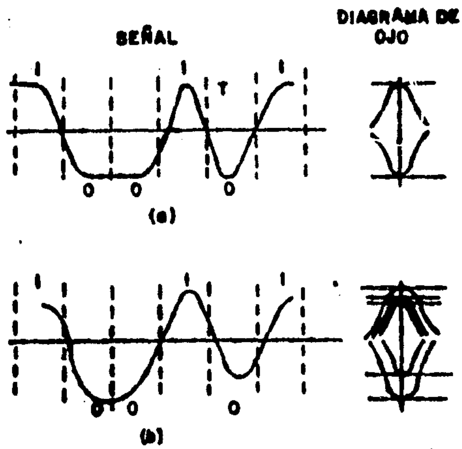


Figura 2.3 Señales binarias y sus correspondientes diagramas de ojo para señales (a) no distorsionada (b) distorsionada

En la onda no distorsionada, cuando todos los segmentos de y(t) son superpuestos el diagrama de ojo está totalmente abierto como en la figura 2.3(a). Dibujando una linea vertical a través del centro del diagrama de ojo, se nota que cuando los instantes de muestreo están ajustados en forma apropiada los valores muestreados serán +1 o -1.

En la figura 2.3(b) la onda es distorsionada por los efectos de la interferencia intersimbólica y ruido. Ahora y(t) no alcanza los valores ±1 en los puntos muestreados. La distorsión se ve claramente en el diagrama de ojo de la figura 2.3(b); el ojo está parcialmente cerrado y consecuentemente la detección es más difícil.

Los diagramas de ojo aportan gran cantidad de información acerca de las características de rendimiento de un sistema de datos. Si no hay demasiado ruido en la señal, se tendrá un diagrama de ojo bien definido, tal como se muestra en la figura 2.4.

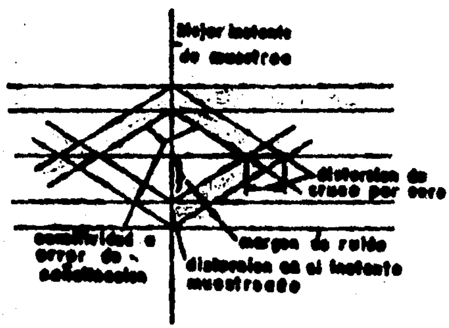


Figura 2.4 Características importantes de un diagrama de ojo

En la figura 2.4 puede notarse lo siguiente

1.- El mejor instante de muestreo debe ocurrir aproximadamente donde el ojo tiene la mayor apertura.

2.- La sensitividad del sistema a errores de señalización es evaluado claramente por la velocidad de cierre del ojo.

3.- La máxima distorsión de la señal es indicada por la anchura vertical de las dos ramas del diagrama en los instantes muestreados, y el margen mínimo contra error de ruido es encontrado como la distancia de la traza mas cercana al umbral en los instantes muestreados.

También pueden ser estudiados los diagramas de ojo de transmisiones multinivel en forma similar. Frecuentemente en estos diagramas de ojo es posible notar no linealidades de los canales de transmisión por asimetrías del diagrama.

2.2 Criterio de Nyquist para pulsos sin ISI

Asumiendo que el canal de banda limitada tiene características de respuesta ideales $C(f)=1$ para $f<W$. Entonces nos interesa saber las propiedades espectrales del pulso $x(t)$ y, el pulso transmitido con el cual resulta una interferencia intersimbólica nula. Esto es

$$x(t=kT) \equiv x_k = \begin{cases} 1 & k=0 \\ 0 & k\neq 0 \end{cases}$$

Supongase entonces que una señal f(t) limitada a W hertz ha sido muestreada a intervalos de 1/2W segundos. Tomando la transformada de Fourier de f(t) dada por

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-j\omega_c t}\, dt \qquad |\omega| < 2\pi W$$

y

$$F(\omega) = 0 \qquad\qquad |\omega| > 2\pi W$$

en función de que f(t) es de banda limitada. $F(\omega)$ puede desarrollarse en una serie de Fourier de período $4\pi W$, para ser usada dentro del intervalo $|\omega|<2\pi W$. Por lo tanto

$$F(\omega) = \frac{1}{4\pi W} \sum_{n=-\infty}^{\infty} Cn\, e^{j(2\pi n/4\pi W)\omega}$$

$$F(\omega) = \frac{1}{4\pi W} \sum_{n=-\infty}^{\infty} Cn \; e^{jn\omega/2W} \qquad |\omega| < 2\pi W$$

$$F(\omega) = 0 \qquad |\omega| > 2\pi W \qquad (2.2.1)$$

en que los Cn son

$$Cn = \int_{-2\pi W}^{2\pi W} F(\omega) \; e^{-jn\omega/2W} \, d\omega \qquad (2.2.2)$$

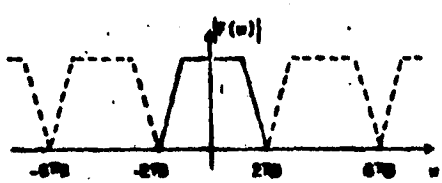


Figura 2.5 F(ω) representada como una función periodica.

pero como F(ω) es la transformada de Fourier de f(t), ésta puede escribirse

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}\,d\omega = \frac{1}{2\pi}\int_{-2\pi W}^{2\pi W} F(\omega)e^{j\omega t}\,d\omega \qquad (2.2.3)$$

por lo tanto en los instantes de muestreo $t=-n/2W$

$$f(-n/2W) = \frac{1}{2\pi}\int_{-2\pi W}^{2\pi W} F(\omega)e^{-jn\omega/2W}\,d\omega = \frac{Cn}{2\pi} \qquad (2.2.4)$$

Por lo que deducimos que si se tiene f(t) en esos instantes de muestreo, pueden encontrarse los correspondientes coeficientes de Fourier Cn.

Ahora, si sustituimos la ecuación (2.2.1) en la ecuación (2.2.3), la representación de f(t) por medio de la integral de Fourier se obtiene como

$$f(t) = \frac{1}{2\pi}\int_{-2\pi W}^{2\pi W} \frac{1}{4\pi W}\left[\sum_{n} Cn\ e^{jn\omega/2W}\right]e^{j\omega t}\,d\omega \qquad (2.2.5)$$

intercambiando la suma y la integral se obtiene

$$f(t) = \sum_{n} \frac{Cn}{2\pi} \frac{1}{4\pi W} \int_{-2\pi W}^{2\pi W} e^{j\omega(t+n/2W)} \, d\omega \qquad (2.2.6)$$

resolviendo (2.2.6) en forma similar que para la ecuación (1.3.2) tenemos

$$f(t) = \sum_{n} \frac{Cn}{2\pi} \frac{\text{sen } 2\pi W(t+n/2W)}{2\pi W(t+n/2W)} \qquad (2.2.7)$$

pero $Cn/2\pi = f(-n/2W)$ segun (2.2.4), entonces

$$f(t) = \sum_{n=-\infty}^{\infty} f\left[\frac{n}{2W}\right] \frac{\text{sen } 2\pi W(t-n/2W)}{2\pi W(t-n/2W)} \qquad (2.2.8)$$

debido a que todos los valores positivos y negativos de n se incluyen en la suma, se puede interpretar de la ecuación (2.2.8) que se toma cada muestra, se multiplica por senx/x y se suman los términos resultantes. Esto es lo que sucede exactamente en un canal con corte en W Hertz cuando pasan las muestras a través de él.

Si suponemos que la velocidad de transmisión para los símbolos $\{I_k\}$ es elegida a ser la velocidad de Nyquist (2W), con el intervalo de Nyquist ($T=1/2W$), entonces la ecuación (2.2.8) se convierte en

$$x(t) = \sum_{n=-\infty}^{\infty} x(nt) \frac{\text{sen } \pi(t-nT)/T}{\pi(t-nT)/T} \qquad (2.2.9)$$

Eliminando la interferencia intersimbólica, los coeficientes $x(nT)$ serán todos cero excepto cuando $n=0$. Consecuentemente el pulso $x(t)$ que da la interferencia intersimbólica nula es

$$x(t) = \frac{\text{sen } \pi t/T}{\pi t/T} \qquad (2.2.10)$$

cuya gráfica se muestra en la figura 1.4 y su espectro es una característica rectangular ideal.

$$x(f) = \begin{cases} T & |f| < 1/2T \\ 0 & |f| > 1/2T \end{cases} \qquad (2.2.11)$$

El problema con el que nos encontramos es el de realizar un pulso que tenga la caracteristica rectangular espectral X(f) dada en (2.2.11), es decir, X(f) no es fisicamente realizable; por lo tanto intentaremos ahora determinar si los pulsos existentes satisfacen las condiciones de interferencia intersimbólica nula.

Para esto

$$x(t) = x(f)\int_{-W}^{W} e^{j2\pi ft}\, df \qquad (2.2.12)$$

deseamos muestrear a x(t) a una velocidad 1/T, así que

$$x(kT) = \int_{-W}^{W} X(f)\ e^{j2\pi fkt}\, df \qquad k=0,\pm1,\pm2,\ldots \qquad (2.2.13)$$

la integral puede ser dividida en segmentos de ancho 1/2T quedando como

$$x(kT) = \sum_{n=-N}^{N} \int_{(2n-1)/2T}^{(2n+1)/2T} X(f)e^{j2\pi fkT}\, df$$

$$= \sum_{n=-N}^{N} \int_{-1/2t}^{1/2T} X(f+\tfrac{n}{T})\ e^{j2\pi fkT}\ df \qquad (2.2.14)$$

donde N es el entero [2TW]. Asumiendo que la integral y la suma pueden ser intercambiadas y, además, definiendo un canal equivalente de Nyquist Xeq(w) con banda limitada a la banda de Nyquist $[-\pi/T, \pi/T]$ rad por segundo, la ecuación (2.2.14) se puede escribir como

$$x(kT) = \int_{1/2T}^{1/2T} Xeq(f)\ e^{j2\pi fkT}\ df \qquad (2.2.15)$$

donde

$$Xeq(f) = \sum_{n=-N}^{N} X(f+\tfrac{n}{T}) \qquad |f| < \frac{1}{2T}$$

ahora, la condición para interferencia intersimbólica nula requiere que

$$
Xeq(f) = \begin{cases} T & |f| < 1/2T \\ 0 & |f| > 1/2T \end{cases} \qquad (2.2.16)
$$

El canal de ancho de banda equivalente de Nyquist es construido dividiendo $X(\omega)$ original en segmentos de ancho $2\pi/T$ y superponiendo todos los segmentos en el intervalo $[-\pi/T, \pi/T]$, tal como se ilustra en la figura 2.6. En tal figura el canal de ancho de banda equivalente de Nyquist es obtenido superponiendo las tres regiones x_{-1}, x_o y x_1. Ya que $x(t)$ es únicamente real, $X(\omega)=X^*(-\omega)$, y solo es necesario considerar frecuencias positivas. Colocando x_{-1} en x_o es equivalente a doblar $X(\omega)$ hacia atrás en sí mismo alrededor de la frecuencia de Nyquist π/T.

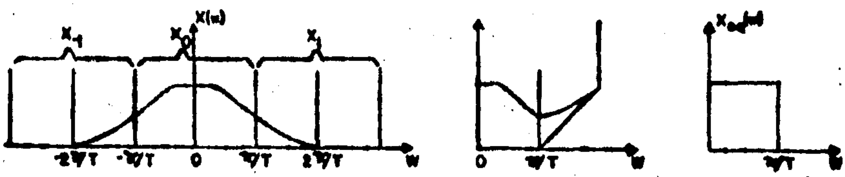


Figura 2.6 (a) Canal de ancho de banda equivalente de Nyquist; (b) Doblado de la característica de exceso del ancho de banda de Nyquist; (c) Canal equivalente perfecto.

También es posible usar una función rectangular mas alguna función arbitraria impar alrededor de π/T para eliminar la interferencia intersimbólica, ya que entonces Xeq(f) tiene la caracteristica espectral rectangular. Esto hace posible diseñar pulsos de banda limitada que dan interferencia intersimbólica nula en un canal ideal de ancho de banda W, condicionado a que la velocidad por simbolo no exceda 2W. Este es el <u>Criterio de Nyquist para diseño de señales sin interferencia intersimbólica</u> [6].

Prácticamente, si se selecciona la velocidad de modo que $W<1/T<2W$, hace posible diseñar una variedad de pulsos que tienen buenas caracteristicas espectrales y son libres de interferencia intersimbólica en un canal ideal.

Una forma de pulso que ha encontrado un amplio uso en transmisión digital sobre canales de banda limitada es la que tiene la caracteristica espectral coseno elevado, la cual es definida por

$$X(\omega)=\begin{cases} T & 0<\omega<\frac{\pi}{T}(1-\alpha) \\ \frac{T}{2}\{1-\mathrm{sen}[\frac{T}{2\alpha}(\omega-\frac{\pi}{T})]\} & \frac{\pi}{T}(1-\alpha)<\omega<\frac{\pi}{T}(1+\alpha) \end{cases}$$

y la respuesta a impulso correspondiente es

$$x(t) = \frac{\operatorname{sen} \pi t/T}{\pi t/T} \cdot \frac{\cos \alpha\pi t/T}{1-4\alpha^2 \quad t^2/T^2}$$

La caracteristica coseno elevado consiste de una porsión de amplitud plana y una porsión de "caida" la cual tiene forma sinusoidal. El espectro $X(\omega)$ es especificado en términos de un parámetro α el cual es la cantidad de ancho de banda usado en exceso del ancho de banda minimo de Nyquist dividido por el ancho de banda de Nyquist. Entonces, como se ve en la figura 2.7, a medida que α decrece, el ancho de banda usado disminuye.

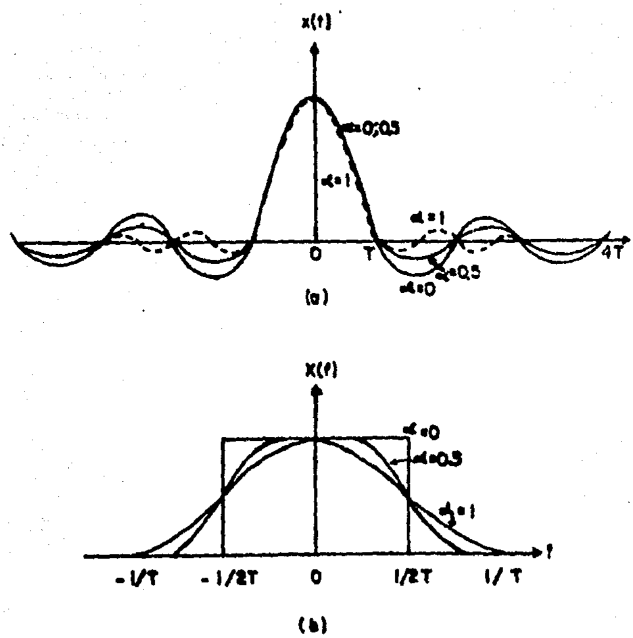


Figura 2.7 Pulsos que tienen un espectro coseno elevado.

Con el coseno elevado se tendrá entonces que un ancho de banda dado es más eficiente utilizando una caída tan pequeña como sea posible.

2.3 El filtro transversal en igualación lineal

La distorsión lineal (distorsión de amplitud y distorsión de retardo) es remediable mediante el uso de redes de igualación, es decir, es posible compensar la distorsión lineal introducida por el canal de transmisión mediante un igualador de parámetros ajustables colocado en cascada con el canal como se muestra en la figura 2.8.

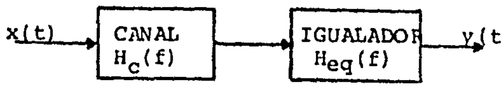


Figura 2.8 Canal con igualador para distorsion lineal.

entonces, la función de transferencia completa del sistema es

$$H(f) = H_c(f)\ H_{eq}(f)$$

con lo que la salida final estará libre de distorsión si se cumple que

$$H_c(f)\ H_{eq}(f) = Ke^{-j\omega td}$$, donde K y td son constantes

2.3.1 Estructura del filtro transversal

En la ecuación (2.1.6), que denota la salida del filtro receptor, se observa que un error ocurre en la detección de a_o cuando la suma de interferencia intersimbólica y ruido excede la distancia dx_o al umbral mas cercano, esto es, la probabilidad de error es

$$Pe = P\left[\left| \sum_{\substack{n=0 \\ n\neq k}}^{\infty} In\ x_{k-n} + v_k \right| > dx_0 \right] \qquad (2.3.1)$$

por tanto la minimización de Pe es entonces realizada usando el mismo tipo de desarrollo empleado en la derivación del criterio de Nyquist en términos del canal de ancho de banda equivalente de Nyquist. Con ello es posible demostrar [6] que las caracteristicas de amplitud y fase óptimas para cada uno de los segmentos $G_{Rk}(\omega)$ son alcanzadas cuando

$$G_{Rk}(\omega) = \left[G_{Tk}(\omega) C_k(\omega) \right]^{*} \sum_{n=-N}^{N} \frac{-\lambda n}{2No} \ e^{i\omega nT} \qquad (2.3.2)$$

donde el término entre paréntesis indica un filtro acoplado (matched filter) tal como el descrito en el capitulo 1.5, este tipo de filtro es óptimo si el canal es ideal; en caso contrario, este filtro se coloca en cascada con el filtro indicado por la sumatoria, formando el receptor lineal óptimo ilustrado en la figura 2.9. El filtro receptor es descompuesto en los dos filtros mencionados, el filtro acoplado $G_T(\omega)C(\omega)$ y el filtro periódico $T(\omega)$ de la forma

$$T(\omega) = \sum_{n=-N}^{N} Cn\ e^{-jn\omega T} \qquad (2.3.3)$$

el cual es llamado un filtro transversal y su representación se muestra en la figura 2.10.

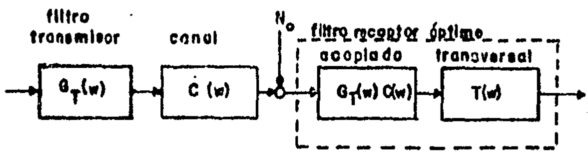


Figura 2.9 Receptor lineal óptimo

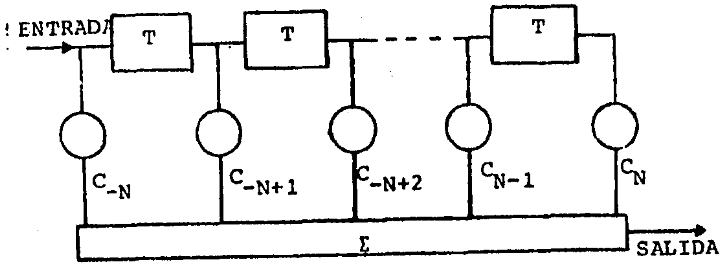


Figura 2.10 Filtro transversal

El filtro transversal consiste de una línea de retardo con ramas a intervalos de T segundos. Cada rama a lo largo de la línea de retardo es conectada a través de un amplificador a un sumador que provee la salida.

Concluyendo, es posible dividir al filtro receptor óptimo en dos partes que son: un filtro acoplado y un filtro transversal. El filtro acoplado puede ser interpretado como el tratamiento al ruido, y el filtro transversal corrige la interferencia intersimbólica. Cuando la relación señal a ruido es baja, la principal corrección debe venir del filtro acoplado; para relación señal a ruido alta, la interferencia intersimbólica debe ser eliminada por una corrección efectuada por el filtro transversal.

2.3.2 Igualación con filtro transversal

El problema que se presenta en el diseño de los filtros transmisor y receptor es que prácticamente no se puede tener un conocimiento exacto de las características del canal, por lo que no es posible producir las condiciones necesarias para el filtro óptimo. El único recurso para disminuir los efectos de la distorsión es incluir en el sistema de datos un filtro o filtros los cuales puedan ser ajustados lo mas cercano posible a las características requeridas para cualquier dato particular. Tradicionalmente a este tipo de filtros se les ha llamado igualadores (ecualizadores) [7].

La forma mas simple de redes variables usada para igualación ha sido la que emplea el filtro transversal con coeficientes ajustables como el que se muestra en la figura 2.11.

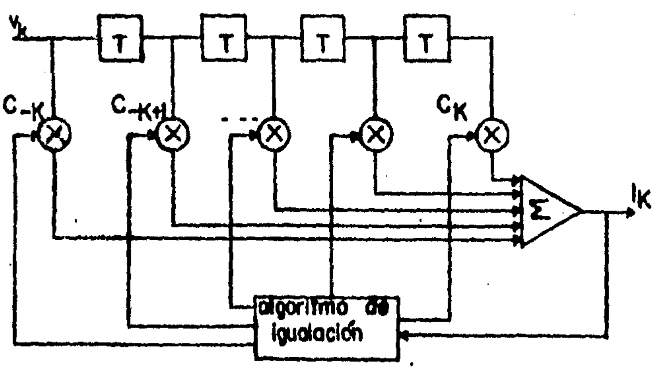


Figura 2.11 Filtro transversal lineal.

En este filtro, los últimos (2K+1) símbolos recibidos son linealmente multiplicados por los coeficientes de igualación (ganancias de rama) Cn, n=-K,-K+1,..,0,..,K, los cuales pueden ser elegidos tales que forcen a los símbolos de salida del igualador a tener valor cero en los instantes de muestreo de los símbolos adyacentes. El valor de Cn que cumple lo anterior es aquel con el que se logra que el efecto combinado del canal y la respuesta a impulso del igualador se cancelen mutuamente.

La estimación del k-ésimo símbolo es expresada como

$$\hat{I}_k = \sum_{j=-K}^{K} C_j \, v_{k-j} \tag{2.3.4}$$

donde C_j son los 2K+1 coeficientes de rama del filtro. La estimación I_k es cuantizada al símbolo de información mas cercano formando la decisión I'_k. Si I'_k no es idéntica al símbolo de información transmitida I_k, surge un error ε; este error tiene gran importancia en el criterio para optimizar los coeficientes Cn del filtro, ya que es deseable escoger los coeficientes que minimicen este índice de rendimiento.

2.3.3 Tipos de igualación automática

Es posible clasificar a los filtros para igualación en dos tipos. Uno es el de igualación preajustada (preset equalization), y el otro el de igualación adaptiva (adaptive equalization) [6].

En la igualación preajustada el igualador es ajustado previamente a, o durante interrupciones de la transmisión de datos usando una señal de prueba.

En la igualación adaptiva el igualador se ajusta por sí mismo en forma continua durante la transmisión de datos usando una señal de referencia u operando sobre la señal de datos misma.

En el igualador preajustado mas simple las componentes de error son medidas transmitiendo pulsos de prueba a través del sistema e inspeccionando la salida del filtro transversal en los instantes muestreados; tal igualador es mostrado en la figura 2.12. En este sistema es usado un método de incremento fijo por iteración. A la salida del filtro transversal los pulsos son muestreados en los tiempos seleccionados por un circuito de tiempo disparado por un detector de pico.

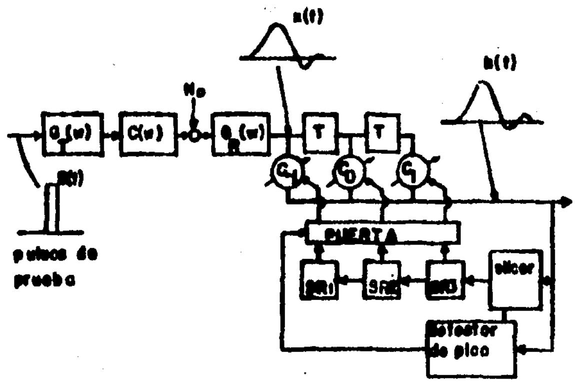


Figura 2.12 Diagrama de bloques de un igualador preajustado con ajuste de incremento fijo

La muestra central es elegida a un nivel 1 (u otro nivel de referencia deseado) para producir ε_0 y las otras muestras se eligen a tener nivel cero. Las polaridades del error son entonces almacenadas en un registro de corrimiento. Al fin de cada pulso de prueba es abierta una puerta y el contenido del registro de corrimiento es usado para incrementar la ganancia de los amplificadores C_i del filtro transversal por un factor de Δ.

En la igualación adaptiva los voltajes de error ε_n son continuamente estimados durante la transmisión normal de datos, y las correcciones del igualador son efectuadas conforme se requieren. Tal sistema tiene una ventaja de exactitud sobre los igualadores preajustados.

2.3.4 Igualador fraccionalmente espaciado

Hasta el momento se ha considerado que la linea de retardo del filtro transversal tiene retardos de T segundos, que es el tiempo de duración de los símbolos de la señal de entrada.

Un igualador transversal fraccionalmente espaciado tiene su linea de retardo fraccionada a intervalos τ menores que el intervalo de símbolo T [7]. El espacio τ entre las ramas es típicamente seleccionado tal que el ancho de banda de la señal a la entrada del igualador es $|f|<1/2\tau$, es decir, τ satisface el teorema de muestreo.

El valor de τ debe ser KT/M, donde K y M son enteros y M>K. En la práctica es mas conveniente elegir τ=T/M donde M es un entero pequeño.

Una propiedad importante del igualador fraccionalmente espaciado (FSE) es la insensitividad de su rendimiento a la elección de la fase muestreada, lo cual puede explicarse como sigue

Primero, muestreando a la velocidad de símbolo de la señal de entrada al igualador T-espaciado, se ocacionará un translape espectral (aliasing); cuando las fases de las componentes translapadas son iguales, se suman constructivamente, pero cuando tienen diferencia de fase de 180° se suman destructivamente, lo cual resulta como una cancelación o reducción de amplitud. El error cuadrático medio (MSE) mínimo de un igualador T-espaciado es por tanto una función de la fase muestreada.

En contraste, en el FSE no hay translape espectral. Entonces, la sensitividad del MSE mínimo con respecto a la fase muestreada es típicamente mucho menor que el de un T-espaciado.

Otro punto de vista es el siguiente: como se vió anteriormente, el filtro receptor óptimo en un sistema de modulación lineal es la cascada del filtro acoplado al canal actual con un filtro transversal T-espaciado. El igualador fraccionalmente espaciado, por virtud de su velocidad de muestreo, puede sintetizar la mejor combinación de las características de un filtro acoplado y un igualador T-espaciado. Un FSE puede compensar mas efectivamente distorsiones de amplitud y retardo mas severas que un T-espaciado.

CAPITULO 3

ALGORITMOS DE AJUSTE EN IGUALACION ADAPTIVA

Introducción

En los capitulos anteriores se ha observado que la forma más conveniente de minimizar la interferencia intersimbólica en una secuencia de datos transmitida por un canal es colocando una red de igualación con dicho canal. Tal red de igualación debe tener parámetros C_i que compensen las caracteristicas no lineales introducidas por el canal.

En este capitulo analizaremos algunos de los algoritmos desarrollados para la actualización de los C_i de acuerdo con los cambios ocurridos en el canal. Estos algoritmos tienen como objetivo fundamental la minimización de un índice de error mediante la solución de ecuaciones lineales.

3.1 Algoritmo Gradiente

El criterio del error cuadrático medio consiste en ajustar los pesos de los coeficientes de manera que se minimice el valor cuadrático medio del error

$$\varepsilon_k = I_k - \hat{I}_k \tag{3.1.1}$$

donde I_k es el símbolo de información transmitido en el k-ésimo intervalo de señalización e $I^\wedge_k$ es la estimación de ese símbolo a la salida del igualador, definida previamente en (2.3.4). El índice de rendimiento J para el criterio del error cuadrático medio es definido por

$$J(k) = E|\varepsilon_k|^2 = E|I_k - \hat{I_k}|^2$$

$$= E|I_k - \sum_{j=-K}^{K} C_j v_{k-j}|^2 \qquad (3.1.2)$$

La minimización de J(K) con respecto al peso de los coeficientes $\{c_j\}$ (o equivalentemente, forzar al error ε_k a ser ortogonal[1] a la señal $v_{k-1}{}^*$, $|1|<K)$ se puede obtener mediante un procedimiento iterativo que determine el vector de coeficientes optimo C_{opt} para tal minimización.

El método iterativo más simple es el método del error cuadrático medio, el cual se inicia eligiendo arbitrariamente el vector C_o. Esta elección de coeficientes corresponde a algún punto sobre la superficie del error cuadrático medio en

---

[1] El producto escalar de dos señales es una estimación del grado de similitud entre estas; si las señales son similares el producto escalar es máximo, y si son ortogonales el producto es cero.

el espacio de coeficientes $(2K+1)$-dimensional; entonces se calcula el vector gradiente G_o compuesto por los $(2K+1)$ componentes

$$\frac{1}{2}\frac{\delta J}{\delta C_o} \quad , \; k=-K,\ldots,-1,0,1,\ldots,K$$

sobre la superficie del error cuadratico medio y, después cada coeficiente de rama es cambiado en dirección opuesta a su correspondiente componente gradiente. Así, el valor del vector C de coeficientes es obtenido de acuerdo a la relación

$$C_{k+1} = C_k - \Delta G_k \qquad k=0,1,2,\ldots \qquad (3.1.3)$$

donde el vector gradiente es

$$G_k = \frac{dJ}{2dC_k} = -E(\varepsilon_k\, V_k^{*}) \qquad (3.1.4)$$

en el que C_k representa el conjunto de coeficientes en la k-ésima iteración, ε_k es la señal de error correspondiente definida anteriormente en (3.1.1), V_k es el vector de muestras de señal recibidas y Δ es un número positivo elegido suficientemente pequeño para asegurar la convergencia del procedimiento iterativo.

La dificultad básica de éste método es el desconocimiento del vector gradiente G_k, por lo que se usa una estimación de tal vector. De la ecuación (3.1.4) notamos que G_k es el negativo del valor de $\varepsilon_k V^*{}_k$; consecuentemente una estimación de G_k es

$$\hat{G}_k = - \varepsilon_k V_k^*$$

con lo cual la ecuación (3.1.3) se modifica de la forma

$$\hat{C}_{k+1} = \hat{C}_k + \Delta \varepsilon_k V_k^* \qquad (3.1.5)$$

este es el algorítmo básico del error cuadrático medio para el ajuste recursivo del peso de los coeficientes del igualador [3]; su representación en bloques es la que se muestra en la figura 3.1.

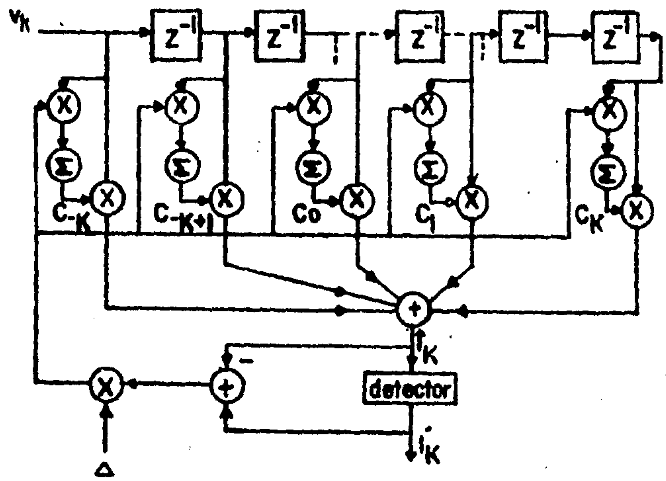


Figura 3.1 Igualador lineal adaptivo basado en criterio del error cuadrático medio

En la deducción de la fórmula anterior se consideró que el receptor tiene conocimiento de la secuencia de información transmitida al calcular la señal de error entre el símbolo deseado y su estimación. Tal conocimiento puede ser disponible durante un corto periodo de entrenamiento en el cual una señal con secuencia de información conocida es transmitida al receptor para el ajuste inicial de los coeficientes de rama. Durante este periodo de entrenamiento el ajuste de los coeficientes de rama es calculado comparando los símbolos de información de la secuencia transmitida conocida I_k con los símbolos estimados $\hat{I}_k$, resultando la señal de error ε_k.

En el modo de operación de decisión directa (operación permanente), la señal de error es $\varepsilon_k = I'_k - I\hat{}_k$ donde I'_k es la decisión de la recepción basada en la estimación $I\hat{}_k$.

Si el canal cambia su respuesta, este cambio es reflejado en la señal de error ε_k, y variarán entonces proporcionalmente los coeficientes de rama para compensar tal cambio y hacer que el error sea mínimo.

En la práctica, el valor de Δ es seleccionado para convergencia rápida durante el periódo de entrenamiento y entonces reducido para ajuste fino en la operación en estado estable del igualador.

Una alternativa para aumentar la eficiencia (velocidad) de convergencia de (3.1.5) es empleando dos tamaños de Δ, uno para cuando el error ε es grande y se requiere una corrección grande en el vector C y otra para cuando se requieren correcciones pequeñas [6].

La mayor ventaja del algoritmo analizado anteriormente es su simplicidad de cálculo, sin embargo, tal simplicidad trae consigo convergencia lenta. La baja velocidad en la convergencia de éste algoritmo es debida a que solo tiene un parámetro (Δ) para controlar la velocidad de convergencia.

Para lograr convergencia más rápida, se han desarrollado algoritmos mas complejos que contienen N parámetros, tales algoritmos utilizan el método de minimos cuadrados para la optimización del indice de rendimiento, por tanto hacen uso de la información disponible acerca de la información anterior de la señal.

Entre los algoritmos de minimos cuadrados más conocidos en igualación adaptiva se encuentran el algoritmo de Kalman, el algoritmo rápido de Kalman y el algoritmo de rejilla. Aquí analizaremos los dos primeros, ya que son los que requieren el máximo y mínimo número de operaciones por iteración respectivamente, además, presentan una velocidad de convergencia muy similar.

3.2 Algoritmo de Kalman

Durante el periodo inicial, en la igualación de prueba o de entrenamiento, una sucesión de simbolos de datos d(1), d(2),...,d(n), conocidos por el algoritmo de ajuste de igualación, es transmitido sobre un canal, resultando en una secuencia de entradas al igualador y(1),y(2),...,y(n). El igualador almacena las N entradas anteriores denotadas por un vector

$$x_N(t) = \begin{bmatrix} y(t-1) \\ y(t-2) \\ \vdots \\ y(t-N) \end{bmatrix} \qquad (3.2.1)$$

en el que los $y(t)$ son escalares y $x_N(t)$ es un vector N-dimensional.

Los coeficientes de rama en el tiempo t estan representados por el vector $C_N(t-1)$ y sus salidas son $C_N(t-1)^T x_N(t)$, la cual tiene una diferencia $\varepsilon(t)$ respecto a la salida ideal $d(t)$ determinada por

$$\varepsilon(t) = d(t) - C_N(t-1)\, x_N^T(t) \qquad (3.2.2)$$

Entonces se requiere que el algoritmo genere el vector de coeficientes $C_N(t)$ que minimice el error cuadrático acumulado

$$\sum_{k=1}^{t} \left[d(k) - C_N(t)^T\, x_N(k) \right]^2 \qquad (3.2.3)$$

y tal minimización se obtiene cuando

$$C_N(t)= R_N(t)^{-1}\left[\sum_{k=1}^{t} d(k)x_N(k)\right]= R_N(t)^{-1}v_N(t) \qquad (3.2.4)$$

y

$$R_N(t)=\sum_{k=0}^{t} \lambda^{t-k}\, x_N(t)x_N(t)^T + \lambda^t\delta I_N \qquad (3.2.5)$$

donde $R_N(t)$ es una estimación de la matriz de correlación[1] de la señal y $v(t)$ es una estimación del vector de correlación de cruce entre la señal deseada y la señal recibida.

δ es una pequeña constante positiva e I_N es la matriz identidad; el producto de ambas se incluye para asegurar una definición de $R_N(t)$ en todo momento. λ es una constante positiva cercana o igual a la unidad; cuando $\lambda=1$, toda la información pasada es considerada en igual proporción en la actualización del vector C_N; para $\lambda<1$ la información

---

[1] La correlación de dos señales se define como un producto escalar, en el cual la segunda señal presenta un desplazamiento o corrimiento de τ segundos. La correlación da la medida de similitud entre estas dos señales como una función del desplazamiento τ de una respecto a la otra. Cuando se hace la correlación de una señal consigo misma se obtiene la función de autocorrelación; esta función nos indicará la variación de esta señal en el tiempo en un sentido promediado.

pasada es atenuada geométricamente, consecuentemente la muestra presente tiene mayor influencia en la actualización de C_N que las muestras anteriores.

Para calcular el vector $C_N(t)$ es conveniente expresar la matriz $R_N(t)$ en la forma recursiva

$$R_N(t)= \lambda R_N(t-1) + x_N(t)\ x_N(t)^T \tag{3.2.6}$$

la cual nos da la actualización en tiempo de $R_N(t)$.

Debido a que en (3.2.4) se necesita la inversa de $R_N(t)$ para calcular $C_N(t)$, esta matriz invertida se puede calcular a partir de (3.2.6) obteniendose

$$R_N^{-1}(t)= \frac{1}{\lambda}\left[R^{-1}(t-1)- \frac{R_N^{-1}(t-1)x_N(t)x_N^T(t)R_N^{-1}(t-1)}{\lambda\ +x_N^T(t)R_N^{-1}(t-1)x_N(t)}\right] \tag{3.2.7}$$

esta ecuación se puede simplificar definiendo los vectores $P_N(t)$, $K_N(t)$ y $\mu_N(t)$ como

$$P_N(t) = R_N^{-1}(t) \tag{3.2.8}$$

$$K_N(t) = \frac{1}{\lambda + \mu_N(t)} P_N(t-1)\ x_N(t) \qquad (3.2.9)$$

$$\mu_N(t) = x_N^T(t)\ P_N(t-1)x_N(t) \qquad (3.2.10)$$

con las cuales (3.2.7) toma la forma

$$P_N(t) = \frac{1}{\lambda}\left[P_N(t-1)-K_N(t)x_N^T(t)P_N(t-1)\right] \qquad (3.2.11)$$

y multiplicando (3.2.11) por $x_N(t)$ es posible simplificar la ecuación resultante de tal forma que se obtenga

$$P_N(t)x_N(t) = \frac{1}{\lambda}\left\{P_N(t-1)x_N(t) - K_N(t)x_N^T(t)P_N(t-1)x_N(t)\right\}$$

$$= K_N(t) \qquad (3.2.12)$$

sustituyendo (3.2.8) en (3.2.4)

$$C_N(t) = P_N(t)\ v_N(t) \qquad (3.2.13)$$

en la que $v_N(t) = \lambda v_N(t-1) + I(t)\ x_N(t)$ (3.2.14)

Partiendo de (3.2.13) y considerando las ecuaciones (3.2.11) y (3.2.14), se encuentra la expresión recursiva para el cálculo del vector $C_N(t)$

$$C_N(t)=C_N(t-1)+K_N(t)\left[I(t) - x_N^T(t)C_N(t-1)\right] \qquad (3.2.15)$$

pero como $x_N^T(t)C_N(t)$ es la estimación de la señal en el tiempo t, la expresión recursiva anterior queda como

$$C_N(t) = C_N(t-1) + K_N(t)\ \varepsilon_N(t) \qquad (3.2.16)$$

El algoritmo formado por las ecuaciones (3.2.2), (3.2.9), (3.2.11) y (3.2.16) es llamado algoritmo de Kalman. En este algoritmo los coeficientes cambian con el tiempo en una cantidad igual al producto del error por el vector de Kalman. Como el vector $K_N(t)$ es N-dimensional, cada coeficiente c_i es controlado por un componente k_i, por lo que se obtiene convergencia rápida.

El algoritmo de Kalman tiene dos desventajas, una es su complejidad, ya que el número de operaciones necesarias para el cálculo de las variables por cada símbolo recibido es proporcional a N^2. La razón de tal cantidad de cálculos es la multiplicación de matrices en el cálculo de la ganancia de Kalman y en la actualización de $P_N(t)$. La otra desventaja de este algoritmo es la sensitividad al "ruido de redondeo" que se acumula en los cálculos recursivos y que causa inestabilidad.

3.3 Algoritmo Rápido de Kalman

Es posible modificar el algoritmo de Kalman de manera que se eliminen las multiplicaciones de matrices involucradas en el cálculo de $K_N(t)$ y de $P_N(t)$. La modificación está basada en el uso de relaciones de predicción directa y predicción de retraso. El algoritmo resultante tiene una complejidad proporcional a N y es conocido como el algoritmo ràpido de Kalman [9].

Las señales muestreadas a la velocidad igual o mayor de la velocidad de muestreo de Nyquist tienen una correlación significativa entre muestras sucesivas. Una forma de aprovechar esta propiedad es la de codificar las diferencias entre muestras sucesivas en vez de las muestras mismas, con lo cual se requieren menos bits por muestra. Si denotamos como y(t) a la muestra actual y y'(t) el valor de predicción de y(t) definido por

$$y'(t) = \sum_{i=1}^{p} a_i y(t-1)$$

entonces y'(t) será una combinación de p muestras pasadas y los $\{a_i\}$ son los coeficientes de predicción seleccionados para minimizar la función de error entre y(t) y y'(t).

Este algoritmo hace uso de la modificación de muestras relacionando las muestras nuevas y(t) y desechando las muestras n-anteriores $\rho(t)$ en los vectores $x_N(t)$ y $x_N(t+1)$ respectivamente. Estas relaciones involucran predicciones de minimos cuadrados directas y de retraso.

Primero definiremos una matriz $A_N(t-1)$ de coeficientes de predicción directa y un error de predicción directa $\varepsilon_d(t)$ asi que

$$y(t) = -A_N(t-1)x_N(t) + \varepsilon_d(t) \qquad (3.3.1)$$

el producto escalar de (3.3.1) es la predicción lineal del elemento y(t) que pertenece al vector $x_N(t)$; la matriz $A_N(t-1)$ es elegida a minimizar la suma cuadrática del error en el tiempo t dada por

$$\xi_d = \sum_{k=1}^{t} \lambda^{t-k} \left[y(k) + A_N(t-1)^T x_N(k) \right]^2 \qquad (3.3.2)$$

La matriz $A_N(t-1)$ que produce esta minimización es, en forma similar a (3.2.4)

$$R_N(t)\ A_N(t) = -B_N(t) \qquad (3.3.3)$$

donde

$$B_N(t) \equiv \sum_{k=0}^{t} \lambda^{t-k} x_N(k)\ y(k)$$

recursivamente

$$B_N(t) = \lambda B_N(t-1) + x_N(t) y(t)$$

y análogamente a (3.2.15), la ecuación (3.3.3) se puede expresar recursivamente como

$$A_N(t) = A_N(t-1) - K_N(t) \varepsilon_d(t) \qquad (3.3.4)$$

Así también es posible deducir las expresiones correspondientes a la matriz de predicción de retraso $D_N(t)$ y al error de predicción de retraso ε_r, así que

$$\rho(t) = -D_N(t-1)^T x_N(t+1) + \varepsilon_r(t) \qquad (3.3.5)$$

de manera que minimice

$$\xi_r = \sum_{k=1}^{t} \lambda^{t-k}\left[\ \rho(t) + D_N(t-1)^T x_N(k+1)\right]^2 \qquad (3.3.6)$$

y como en (3.3.3)

$$R_N(t+1)\ D_N(t) = -\ U_N(t) \qquad (3.3.7)$$

en la que

$$U_N(t) \equiv \sum_{k=1}^{t} \lambda^{t-k} x_N(k+1)\ \rho(k)$$

por lo tanto

$$D_N(t) = D_N(t-1) - K_N(t-1)\varepsilon_r(t) \qquad (3.3.8)$$

suponiendo que tenemos en la n-ésima iteración $(n>1)$ el vector $K_N(n)$ tal que satisface la ecuación (3.2.12), requerimos encontrar el vector $K_N(n+1)$ tal que

$$R_N(n+1)\ K_N(n+1) = x_N(n+1) \qquad (3.3.9)$$

para ello definiremos ahora, a partir de (3.2.6), una matriz extendida $R_M(n)$ que cumpla con la ecuación

$$R_M(n) = \lambda R_M(n-1) + x_M(n)\ x_M(n)^T \qquad (3.3.10)$$

y que contenga a los elementos de $R_N(n)$ y $R_N(n+1)$.

Si hacemos una partición de esta matriz mediante el uso de las matrices de permutación S_M y Q_M[1] podemos obtener

$$S_M R_M(n) S_M^T = \left[\begin{array}{c|c} \pi(n) & B_N(n)^T \\ \hline \bar{B}_N(\bar{n}) & \bar{R}_N(\bar{n}) \end{array} \right] \qquad (3.3.11)$$

---

[1] Existen matrices de permutación cuyos vectores y columnas contienen unicamente un 1 y tienen la propiedad de que $S_M^{-1} = S_M^T$. Estas matrices se utilizan para hacer particiones dentro de una matriz o vector.

y

$$Q_M R_M(n) Q_M^T = \left[\begin{array}{c|c} R_N(n+1) & U_N(n) \\ \hline U_N(n)^T & \pi'(n) \end{array} \right] \qquad (3.3.12)$$

en los que

$$\pi(n) = \lambda\pi(n-1) + y(n)\ y(n)^T$$

$$\pi'(n) = \lambda\pi'(n-1) + \rho(n)\ \rho(n)^T$$

entonces, para un vector extendido R_M corresponderán los vectores K_M y x_M que cumplan con la ecuación (3.3.9). Por lo tanto considerando los vectores extendidos y la ecuación (3.3.11) podemos encontrar el valor de K_M como

$$R_M(n)\ K_M(n) = x_M(n) \qquad (3.3.13)$$

al despejar $K_M(n)$ y recordando que $S_M^T = S_M^{-1}$,

$$K_M(n) = S_M^T (S_M R_M(n)\ S_M^T)^{-1} S_M x_M(n) \qquad (3.3.14)$$

así, haciendo uso de la ecuación (3.3.3) y las propiedades de

S_M obtenemos

$$K_M(n) = S_M^{-1} \begin{bmatrix} E(n)^{-1}\varepsilon_d(n) \\ \hline K_N(n)+A_N(n)E(n)^{-1}\varepsilon(n)^{-1} \end{bmatrix} \qquad (3.3.15)$$

donde

$$E(n) \equiv \pi(n) + A_N(n)^T B_N(n) \qquad . \qquad (3.3.16)$$

es llamado el error residual, y

$$\varepsilon_d(n)' \equiv y(n)+A_N(n)^T x_N(n)$$

es el error entre $y(n)$ y su valor estimado $-A_N(n)^T x_N(n)$.

Además es posible expresar $E(n)$ recursivamente en la forma

$$E(n) = \lambda E(n-1) + y(n)y(n)^T + A_N(n)^T B_N(n) - \lambda A_N(n-1)^T B_N(n-1)$$

$$E(n) = \lambda E(n-1) + \varepsilon_d(n)\ \varepsilon_d(n)'^T \qquad (3.3.17)$$

Finalmente, podemos encontrar el vector $K_N(n+1)$ a partir de $K_M(n)$ formando

$$Q_M K_M(n) \triangleq \begin{bmatrix} m_N(n) \\ \overline{\mu(n)} \end{bmatrix} \tag{3.3.18}$$

que con (3.3.12) se tiene

$$Q_M R_M(n) Q_M\, Q_M^T K_M(n) = \begin{bmatrix} R_N(n+1) m_N(n) + U_N(n)\mu(n) \\ U_N(n)^T m_N(n) + \pi(n)'\mu(n) \end{bmatrix} \tag{3.3.19}$$

sustituyendo (3.3.13) en (3.3.19) y haciendo una particion de $x_M(n)$ en la forma

$$Q_M x_M(n) = \begin{bmatrix} x_N(n+1) \\ \rho(n) \end{bmatrix} \tag{3.3.20}$$

tenemos

$$R_N(n+1) m_N(n) + U_N(n)\mu(n) = x_N(n+1) \tag{3.3.21}$$

pero como $U_N(n)$ esta dada como en (3.3.7), la ecuación (3.3.21) se puede escribir como

$$R_N(n+1)\left[m_N(n) - D_N(n)\mu(n)\right] = x_N(n+1) \qquad (3.3.22)$$

la cual tendrá la forma de (3.3.9) si se cumple que

$$K_N(n+1) = m_N(n) - D_N(n)\,\mu(n) \qquad (3.3.23)$$

que es la ecuación de actualización para el vector $K_N(t)$.

Para obtener la actualización de D_N sin depender del vector K_M, es conveniente sustituir (3.3.18) en (3.3.8), resultando la expresión

$$D_N(n) = [D_N(n-1) - m_N(n)\varepsilon_r(n)^T][I - \mu(n)\varepsilon_\rho(n)]^{-1} \qquad (3.3.24)$$

Entonces el algoritmo rápido de Kalman es el formado por las ecuaciones (3.3.1), (3.3.4), (3.3.15), (3.3.5), (3.3.24), (3.3.23), (3.2.2) y (3.2.16).

Al igual que con el algoritmo de Kalman, el algoritmo rápido de Kalman tiene la desventaja de que en el cálculo recursivo de los vectores que utiliza se introduce "ruido de redondeo" que se va acumulando, es decir, requiere de una gran precisión en los cálculos de las variables recursivas, ya que de lo contrario existirá un error acumulativo que lleva al algoritmo a volverse inestable fácilmente.

3.4 Comparación entre algoritmos

Considerando las divisiones como multiplicaciones, el número de operaciones necesarias para calcular cada salida del igualador es, para los tres algoritmos descritos anteriormente, la que se muestra en la siguiente tabla

Algoritmo	Multiplicaciones	Adiciones
Gradiente	2N	2N
Rapido de Kalman	10N + 4	12N + 5
Kalman	$3N^2 + 3N$	$2N^2+2N+1$

Tabla 3.1 Número de operaciones por iteración.

Para N grande, la complejidad del igualador con la actualización por el rápido de Kalman es alrededor de 5 veces la del gradiente, sin embargo, la actualización con el Kalman es alrededor de N veces esta última.

De lo anterior podemos hacer notar, entonces, que el algoritmo rápido de Kalman tiene una complejidad proporcional a N, justo como el algoritmo gradiente, pero ofrece la misma velocidad de convergencia que el Kalman cuya complejidad es proporcional a N^2. Por tanto el algoritmo de Kalman no es recomendable para aplicaciones prácticas. El algoritmo

Gradiente, a pesar de que presenta convergencia lenta, puede dar buenos resultados en la igualación de señales en tiempo real dada su simplicidad de procesamiento.

CAPITULO 4

PROGRAMACION Y PRUEBAS EN COMPUTADORA DE LOS ALGORITMOS DE IGUALACION

Introducción

De los algoritmos analizados en el capítulo anterior se puede observar que el gradiente y el rápido de Kalman son los que representan la menor complejidad y la mayor velocidad de convergencia respectivamente; por esta razón son los más convenientes para aplicaciones prácticas de igualación en canales reales.

En este capítulo se presentan las pruebas de ambos algoritmos realizadas en computadora, así como la comparación de estos resultados con los obtenidos con un algoritmo de igualación óptima fija (no adaptiva). Posteriormente se detallan las consideraciones hechas para la elección del algoritmo gradiente para su programación en lenguaje ensamblador en el desarrollo practico de un igualador adaptivo.

Para realizar en computadora las pruebas del comportamiento de estos algoritmos, fue necesaria la programación de filtros pasa-bajos para introducir interferencia en la secuencia de bits original. Los listados correspodientes a los programas utilizados en estas pruebas se pueden observar en el apéndice A.

Para las pruebas realizadas se consideró que la señal transmitida es bipolar sin retorno a cero (NRZ).

El sistema utilizado para la simulación en computadora de los algoritmos de igualación es el mostrado en la figura 4.1.

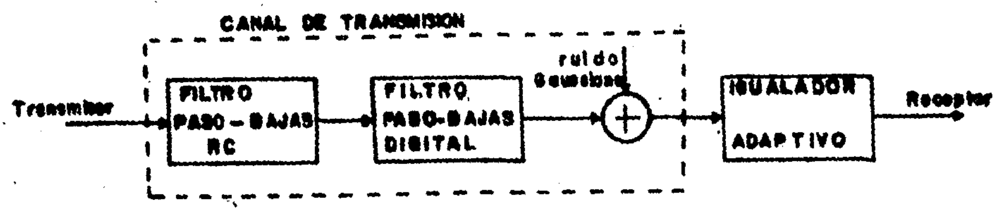


Figura 4.1 Sistema usado para la simulación en computadora de los algoritmos de igualación.

4.1 Igualador óptimo de coeficientes fijos

Para obtener una referencia en el comportamiento de los algoritmos de igualación adaptiva se programó un igualador de parámetros fijos que realiza la "deconvolución" de la señal distorsionada para calcular el vector de coeficientes.

El principio de funcionamiento de este igualador parte de la existencia de un filtro transversal con coeficientes óptimos para un canal determinado, además de que se debe conocer en todo momento la secuencia de símbolos transmitida y cumplirse que $x_i=0$ para $t<0$.

La salida P_i del filtro transversal será igual al símbolo transmitido siempre y cuando los coeficientes c_k sean los que compensen adecuadamente la distorsión introducida por el canal de transmisión. Entonces se tendrá que para la primera muestra (x_1), la salida correspondiente del filtro transversal (P_1) será

$$P_1 = c_1 x_1 \tag{4.1.1}$$

por lo que el primer coeficiente se puede encontrar despejando c_1 en la ecuación anterior. En igual forma, se pueden calcular los demás coeficientes óptimos del filtro transversal, uno por cada muestra de entrada. Así, es posible obtener las ecuaciones

$$P_2 = c_1 x_2 + c_2 x_1 \quad \rightarrow \quad c_2 = \frac{P_2 - c_1 x_2}{x_1} \tag{4.1.2}$$

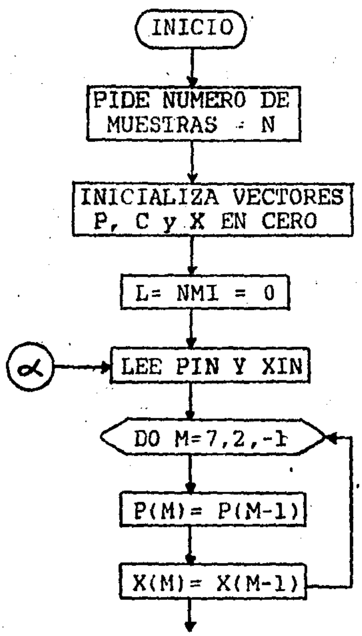
$$P_3 = c_1 x_3 + c_2 x_2 + c_3 x_1 \rightarrow \quad c_3 = \frac{P_3 - c_1 x_3 - c_2 x_2}{x_1} \tag{4.1.3}$$

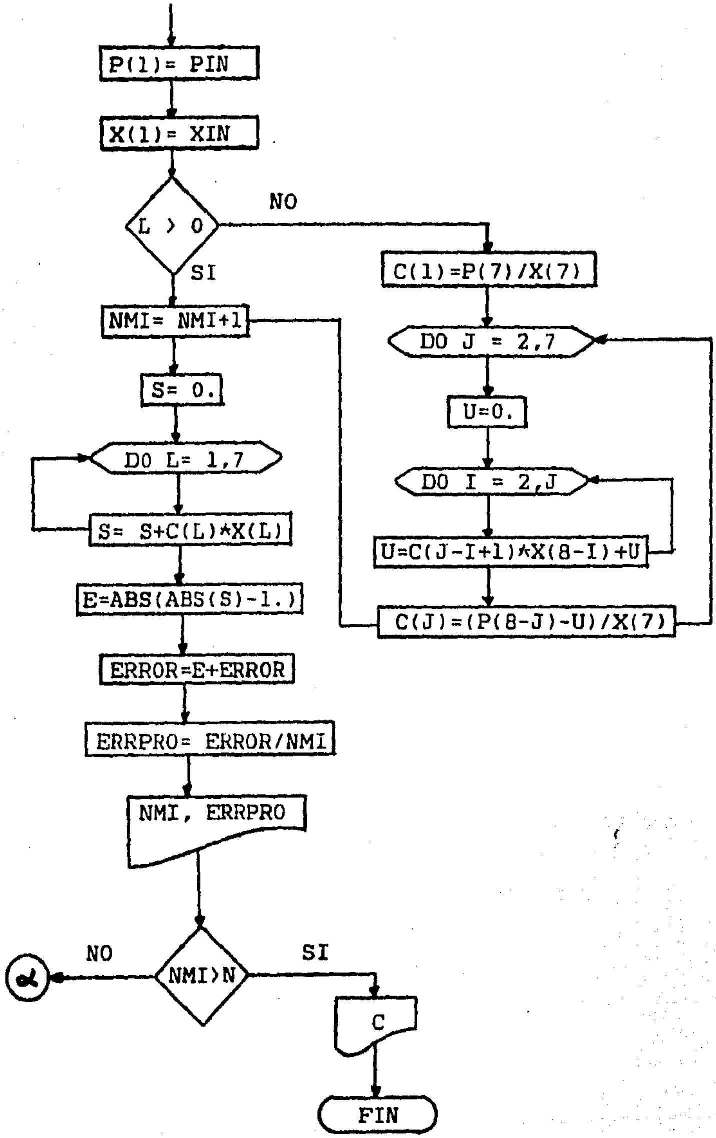
y así sucesivamente hasta c_n. Todas estas ecuaciones se pueden generalizar como

$$c_1 = \frac{P_1}{x_1} \qquad (4.1.4)$$

$$c_i = \frac{1}{x_1}\left[P_i - \sum_{k=1}^{i-1} c_k x_{i-k+1} \right] \quad i=2,3,\ldots,n \qquad (4.1.5)$$

El diagrama de flujo correspondiente al igualador fijo de coeficientes óptimos es el siguiente




P(1)= PIN
X(1)= XIN
L > 0
NO
SI
C(1)=P(7)/X(7)
DO J = 2,7
U=0.
DO I = 2,J
U=C(J-I+1)*X(8-I)+U
C(J)=(P(8-J)-U)/X(7)
NMI= NMI+1
S= 0.
DO L= 1,7
S= S+C(L)*X(L)
E=ABS(ABS(S)-1.)
ERROR=E+ERROR
ERRPRO= ERROR/NMI
NMI, ERRPRO
NO
NMI>N
SI
C
FIN

Para el caso que el canal no cambie sus caracteristicas, es posible continuar con estos coeficientes indefinidamente, obteniendo a la salida una señal muy parecida a la transmitida originalmente.

Definición de variables

P(I)= Vector de símbolos transmitidos

X(I)= Vector de muestras distorsionadas por el canal

C(I)= Vector de coeficientes de igualación

NMI = Numero de muestras a procesar

ITER= Contador de muestras procesadas de cada grupo de 50

ERROR=Error acumulado

E= Error por muestra

ERRPRO=Error promedio

S= Estimación resultante de la convolución C*X

MAX= Contador de grupos de 50 muestras procesadas

ITER= Contador de muestras en cada grupo de 50

U= Variable auxiliar

Listado del programa correspondiente al igualador fijo con coeficientes optimos

```
C      ------------------------------------------------------
C      ! PROGRAMA CORRESPONDIENTE AL IGUALADOR FIJO DE !
C      ! COEFICIENTES OPTIMOS. AL INICIO PIDE EL NUMERO DE !
C      ! MUESTRAS A PROCESAR (MENOR DE 99999), LEE MUESTRAS !
C      ! DISTORSIONADAS CON SU RESPECTIVO SIMBOLO TRANSMITI-!
C      ! DO NO DISTORSIONADO DE UN ARCHIVO DE DATOS LLAMADO !
C      ! SENAL.DAT;LOS RESULTADOS LOS DEPOSITA EN EL ARCHIVO!
C      ! DE DATOS COEFIDEAL.DAT. LA LECTURA DE LOS DATOS ES !
C      ! EN FORMA CICLICA DE 50 MUESTRAS DEL ARCHIVO !
C      ! SENAL.DAT HASTA COMPLETAR LAS NMI MUESTRAS.        !
C      ------------------------------------------------------
C
      DIMENSION C(18),P(18),X(18)

OPEN(91,NAME='COEFIDEAL.DAT',TYPE='NEW',RECORDSIZE=200)
      OPEN(77,NAME='SENAL.DAT',TYPE='OLD',RECORDSIZE=200)
C     PIDE NUMERO DE MUESTRAS A SER PROCESADAS
  253 TYPE 2
      ACCEPT 5,NMI
C     INICIALIZA LOS VECTORES C, P Y X
      DO 3 L=1,9
      C(L)=0.
      P(L)=0.
    3 X(L)=0.
C     INICIALIZA CONTADORES
      MAX=0
      ERROR=0.
      INIT=0
C     LEE LAS PRIMERAS 9 MUESTRAS Y LOS
C     PRIMEROS 9 SIMBOLOS
      DO 200 LL=1,9
    9 ITER=0
   10 ITER=ITER+1
      READ(77,12)IN,XIN
      DO 14 M=9,2,-1
      P(M)=P(M-1)
   14 X(M)=X(M-1)
      P(1)=IN
  200 X(1)=XIN
      IF(INIT.GT.0)GO TO 91
      INIT=1
C     CALCULA COEFICIENTES
      C(1)=P(9)/X(9)
      DO 20 J=2,9
      U=0.
      DO 15 I=2,J
```

```
   15 U=C(J-I+1)*X(10-I)+U
   20 C(J)=(P(10-J)-U)/X(9)
C     CALCULA CONVOLUCION
   91 S=0.
      DO 25 L=1,9
   25 S=S+C(L)*X(L)
C     CALCULA ERROR POR MUESTRA Y ERROR ACUMULADO
      E=ABS(ABS(S)-1.)
      ERROR=E+ERROR
      NUMUES=ITER+((MAX)*50)
C     CALCULA ERROR PROMEDIO
      ERRPRO=ERROR/NUMUES
      PRINT 321,NUMUES,ERROR,ERRPRO
C     ESCRIBE NUMERO DE MUESTRAS PROCESADAS Y
C     ERROR PROMEDIO EN EL ARCHIVO DE DATOS
C     COEFIDEAL.DAT
      WRITE(91,*)NUMUES,ERRPRO
      IF(ITER.LT.50)GO TO 10
      MAX=MAX+1
      IF(MAX*50.GE.NMI)GO TO 32
C     CIERRA Y ABRE EL ARCHIVO DE DATOS SENAL.DAT
C     PARA LEER NUEVAMENTE LAS 50 MUESTRAS
      CLOSE(77)
      OPEN(77,NAME='SENAL.DAT',TYPE='OLD')
      GO TO 9
   32 DO 40 I=1,9
C     MUESTRA EN PANTALLA LOS COEFICIENTES CALCULADOS
      PRINT 30,I,C(I)
   40 CONTINUE
      CLOSE(91)
      CLOSE(77)
      STOP
    2 FORMAT(' NUMERO DE MUESTRAS A CONSIDERAR =', $)
    5 FORMAT(I5)
   12 FORMAT(I3,F8.4)
   30 FORMAT(' C(',I2,')=',F12.4)
  321 FORMAT(' MUESTRA NUMERO ',I5,'  E.AC=',F12.5,
     1'    E.PROM.=',F8.5)
      END
```

Mensajes a terminal del usuario:

Solicita el numero de muestras a procesar enviando el mensaje "NUMERO DE MUESTRAS A CONSIDERAR=", al cual se debe contestar tecleando un numero entero menor o igual a 99999 sin punto decimal.

Observaciones

Tanto en éste como en los programas de los algoritmos adaptivos se leen periodicamente 50 muestras del archivo de datos de la señal distorsionada, hasta procesar el total de iteraciones programado.

Es conveniente en este punto, analizar el comportamiento de este algoritmo para diferentes longitudes del vector C, y así poder determinar el número minimo de coeficientes que resulte en un error máximo tolerable a la salida.

Para este análisis, se procesaron con diferentes longitudes del vector C, 100 muestras de una señal distorsionada por filtros paso-bajas ideales (ver apéndice A) con $\eta/F=1/5$, $1/3$ y $1/2$, obteniendose en cada caso el error promedio (e.p.) mostrado en las tablas siguientes

n	7	8	9	11	17
e.p.	0.1156	0.0988	0.0830	0.0833	0.1156

(a) $\eta/F=1/5$

n	7	8	9	11	17
e.p.	0.1003	0.738	0.047	0.0458	0.0337

(b) $\eta/F=1/3$

n	7	8	9	11	17
e.p.	0.0978	0.0794	0.069	0.0532	0.0456

(c) $\eta/F=1/2$

Tabla 4.1 Error promedio e.p. para diferentes magnitudes del vector C.

Como se observa, el filtro con 9 coeficientes es el que tiene la longitud mínima del vector C sin introducir un incremento considerable en el error promedio; por esta causa los algoritmos de igualación adaptiva fueron programados con 9 coeficientes de rama.

4.2 Algoritmo gradiente

La programación en Fortran del algoritmo gradiente se hizo partiendo de las ecuaciones (2.3.4), (3.1.1) y (3.1.5) las cuales, por comodidad, se repiten a continuación

$$\hat{I}_k = \sum_{i=-K}^{K} C_i x_{k-i} \tag{4.1.6}$$

$$e_k = I_k - \hat{I}_k \tag{4.1.7}$$

$$c_{k+1} = c_k + \Delta e_k x_k^* \tag{4.1.8}$$

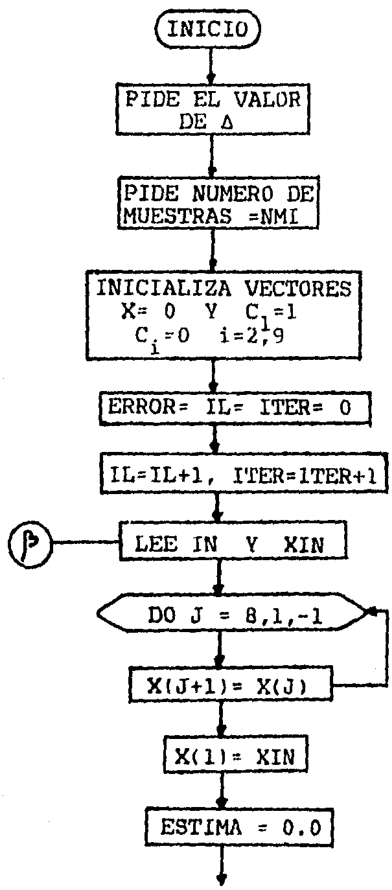
Las condiciones iniciales para este algoritmo son

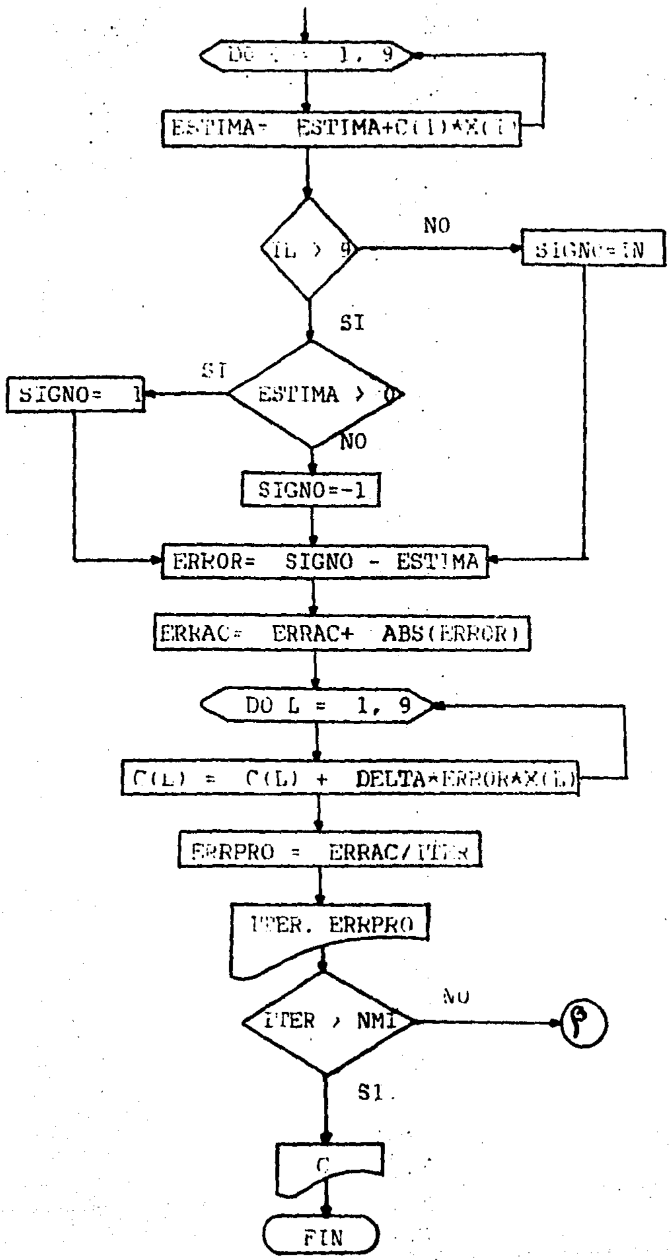
El vector $x_i = 0. \quad ;i=1,2,..,9$

El vector $c_j = 0. \quad ;j=2,3,..,9; \quad c_1=1.0$

Para el cálculo del error dado en (4.1.7) lo mas conveniente es tener un periodo inicial de entrenamiento en el cual se tenga conocimiento del simbolo transmitido correspondiente a la muestra leida, entonces el error se calcula restando a este valor la estimación resultante de la convolución. Posteriormente a éste periodo de entrenamiento, se considera correcto el valor de la estimación y se hace una cuantización al nivel del símbolo mas cercano (+1 o -1); finalmente el error resultará de la diferencia entre el valor estimado y el valor deseado.

El diagrama de flujo para este algoritmo es





DO I = 1, 9
ESTIMA= ESTIMA+C(I)*X(I)
IL > 9
NO
SIGNO=IN
SI
SI
SIGNO= 1
ESTIMA > 0
NO
SIGNO=-1
ERROR= SIGNO - ESTIMA
ERRAC= ERRAC+ ABS(ERROR)
DO L = 1, 9
C(L) = C(L) + DELTA*ERROR*X(L)
ERRPRO = ERRAC/ITER
ITER. ERRPRO
ITER > NMI
NO
SI
C
FIN

Definición de variables

DELTA= Δ

IN= Simbolo transmitido

XIN= Muestra distorsionada

X(I)=Vector de muestras distorsionadas

ESTIMA= Resultado de la convolución C*X

SIGNO= Valor al que se cuantiza basado en ESTIMA

C(I)= Vector de coeficientes de igualación

MAX= Contador de grupos de 50 muestras

ITER= Contador de muestras procesadas de un grupo de 50

NUMUES= Contador total de muestras procesadas

ERRPRO= Error promedio total

ERRAC= Error acumulado total

Listado del programa

ALGORITMO GRADIENTE

```
C     -----------------------------------------------------------
C     ! PROGRAMA  CORRESPONDIENTE  AL  ALGORITMO  GRADIENTE.!
C     ! PIDE VALOR DE  LA  CONSTANTE  DELTA (.003<Δ<.07) !
C     ! Y EL NUMERO  DE  MUESTRAS A  PROCESAR (N<99999). LEE!
C     ! MUESTRAS DEL ARCHIVO DE  DATOS SENAL.DAT  Y  ESCRIBE!
C     ! RESULTADOS  (NUMERO DE ITERACIONES Y ERROR PROMEDIO)!
C     ! EN EN ARCHIVO DE DATOS SALE.DAT.                     !
C     -----------------------------------------------------------
C
      DIMENSION X(9),C(9)
      OPEN(80,NAME='SALE.DAT',TYPE='NEW',RECORDSIZE=200)
      OPEN(77,NAME='SENAL.DAT',TYPE='OLD',RECORDSIZE=200)
C     INICAILIZA VECTORES C Y X
  253 DO 10 I=1,9
      C(I)=0.0
   10 X(I)=0.
      C(1)=1.0
      ERROR=0.0
      IL=0
C     PIDE VALORES DE Δ Y
C     NUMERO DE ITERACIONES
      TYPE 30
      ACCEPT 40,DELTA
      TYPE 50
      ACCEPT 60,NMI
      IL=0
      MAX=0
   70 ITER=0
   65 IL=IL+1
   80 ITER=ITER+1
C     LEE MUESTRA Y SIMBOLO
      READ(77,90)IN,XIN
C     RECORRE VECTOR X
      DO 100 J=8,1,-1
  100 X(J+1)=X(J)
      X(1)=XIN
C     CALCULA CONVOLUCION
      ESTIMA=0.0
      DO 110 I=1,9
  110 ESTIMA=ESTIMA+C(I)*X(I)
      IF(IL.GT.9)GO TO 220
```

```
C       SECUENCIA DE ENTRENAMIENTO
        SIGNO=IN
        PRINT 21,IL,SIGNO
        GO TO 130
C       FIN DE SECUENCIA DE ENTRENAMIENTO
C       CALCULA SIGNO DE LA ESTIMACION
    220 IF(ESTIMA.GE.0)GO TO 120
        SIGNO=-1.0
        GO TO 130
    120 SIGNO=1.0
C       CALCULA ERROR ACTUAL Y
C       ERROR ACUMULADO
    130 ERROR=SIGNO-ESTIMA
        ERRAC=ERRAC+ABS(ERROR)
C       ACTUALIZA COEFICIENTES
        DO 140 L=1,9
    140 C(L)=C(L)+DELTA*ERROR*X(L)
C       CALCULA NUMERO TOTAL DE MUESTRAS PROCESADAS
C       Y EL ERROR PROMEDIO
        NUMUES=ITER+((MAX)*50)
        ERRPRO=ERRAC/NUMUES
C       ESCRIBE RESULTADOS EN SALE.DAT
        WRITE(80,*)NUMUES,ERRPRO
    435 IF(ITER.LT.50)GO TO 65
        MAX=MAX+1
        IF(MAX*50.GE.NMI)GO TO 420
        CLOSE(77)
        OPEN(77,NAME='SENAL.DAT',TYPE='OLD',RECORDSIZE=200)
        GO TO 70
C       PRESENTA COEFICIENTES EN PANTALLA
    420 DO 200 K=1,9
    200 PRINT 210,K,C(K)
        CLOSE(77)
        CLOSE(80)
        STOP
     21 FORMAT(' ITERACION ',I2,' SIMBOLO= ',F8.4)
     30 FORMAT('  DELTA= ',$)
     35 FORMAT('    DELTA=',F10.7)
     40 FORMAT(F10.7)
     50 FORMAT(' NUMERO DE MUESTRAS= ',$)
     60 FORMAT(I5)
     90 FORMAT(I3,F8.4)
    210 FORMAT('     C(',I2,')= ',F8.4)
        END
```

Mensajes a terminal del usuario

Primero pide el valor de la constante Δ, a lo que hay que responder tecleando un numero entre 0.003 y 0.07.

Después pide el numero de muestras a procesar enviando el mensaje "NUMERO DE MUESTRAS=", a lo cual se debe teclear un numero entero maximo de 99999 sin punto decimal.

Observaciones

Los listados de los algoritmo Gradiente y Rápido de Kalman (este último será el siguiente programa a analizar) presentan el caso de una secuencia de entrenamiento de 9 muestras, pero es posible cambiarla colocando el número deseado de muestras de entrenamiento en la instrucción (IF) anterior al comentario "C SECUENCIA DE ENTRENAMIENTO".

Para la elección del valor de la constante Δ, se probó para una misma señal de entrada el comportamiento del algoritmo para diferentes valores de esta constante. El resultado de esta prueba es la diferencia de convergencia que se muestra en la figura 4.2. En esta figura se observa que para Δ=0.0007 el error promedio tiene una disminución lenta; para Δ=0.03 el error tiene cambios mas rápidos. Al probar el algoritmo con Δ mayor a 0.07, en algunos casos se perdió estabilidad.

Se puede concluir de lo anterior que el valor de Δ conveniente debe quedar en el intervalo $.003 \leqslant \Delta \leqslant .07$.

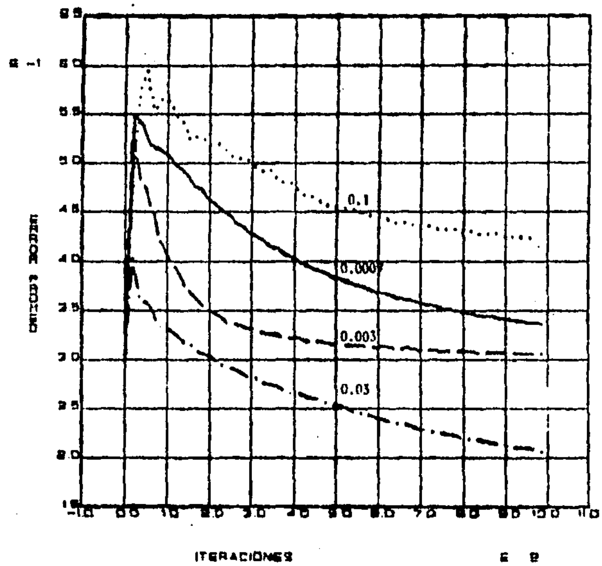


Figura 4.2 Convergencia inicial del algoritmo gradiente para diferentes valores de la constante Δ.

En la figura 4.3 se muestran las curvas de convergencia del algoritmo gradiente para diferentes niveles de ruido de la señal de entrada. El programa fue corrido con $\Delta=0.02$ y 18 iteraciones en la secuencia de entrenamiento.

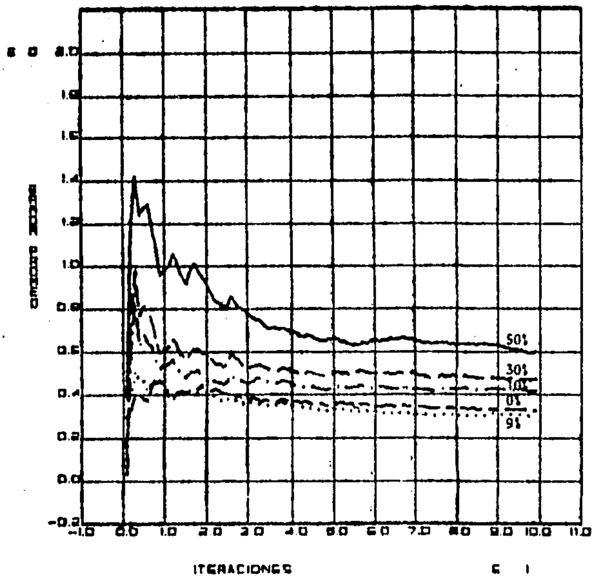


Figura 4.3 Convergencia del algoritmo gradiente para diferentes niveles de ruido de la señal de entrada.

De la figura 4.3 cabe hacer notar que para niveles de ruido menores al 10% de la señal de entrada, el algoritmo no presenta grandes cambios de convergencia respecto a la curva obtenida con la señal sin ruido. Sin embargo, para niveles de ruido mayores al 10% de la señal de entrada, se presenta un transitorio inicial y la convergencia del algoritmo disminuye proporcionalmente al nivel de ruido de la señal de entrada.

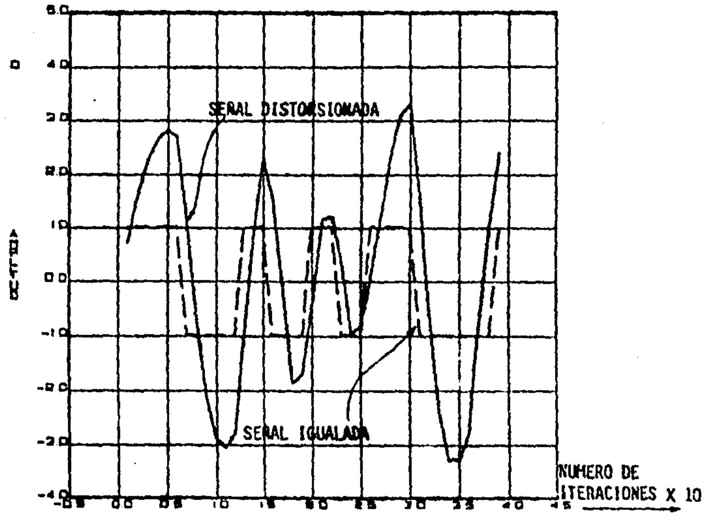


Figura 4.4 Señales de entrada y de salida del igualador adaptivo que funciona bajo el criterio del algoritmo gradiente.

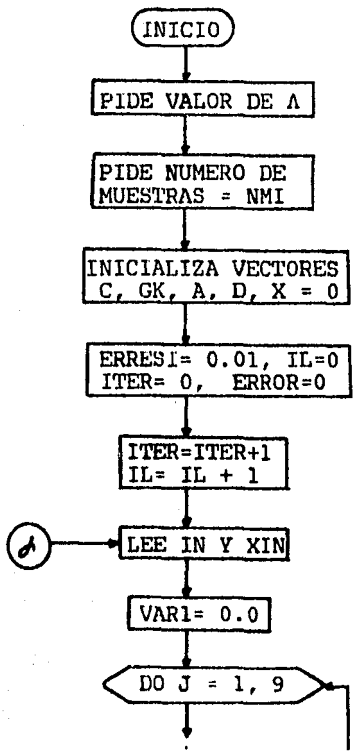
En la figura 4.4 se muestran las formas de onda de la señal distorsionada e igualada correspondientes a la entrada y salida del igualador que ejecuta el algoritmo gradiente con 9 coeficientes. Como se puede observar en esta figura, la señal igualada toma exactamente los valores +1 y -1, mientras que la señal distorsionada tiene grandes variaciones de nivel.

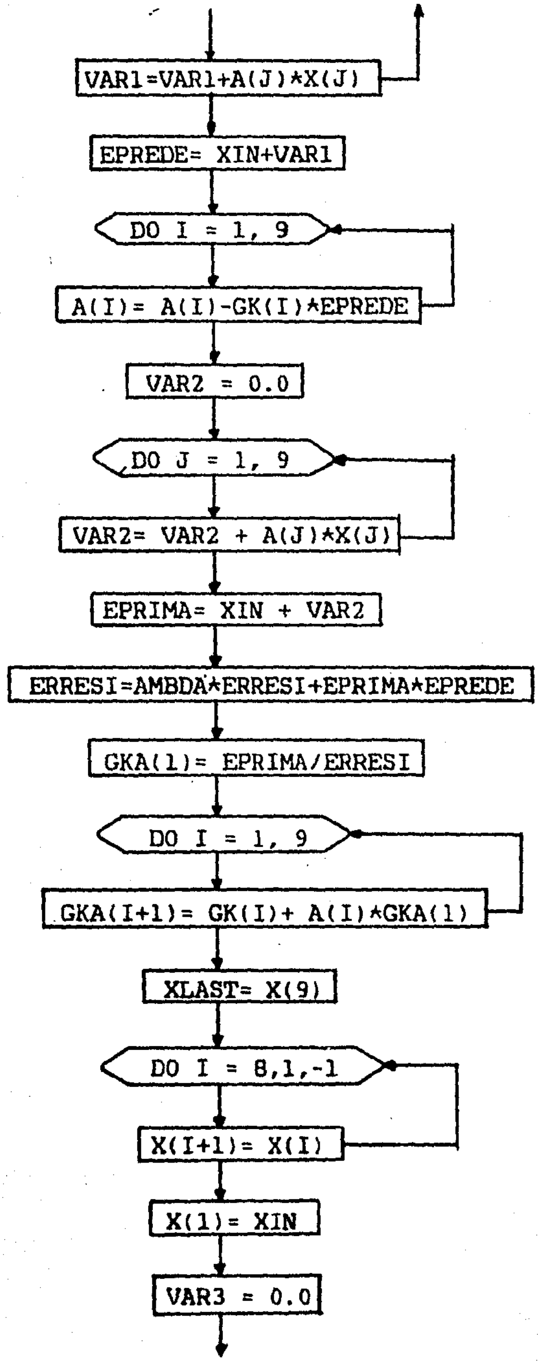
4.3 Algoritmo rápido de Kalman

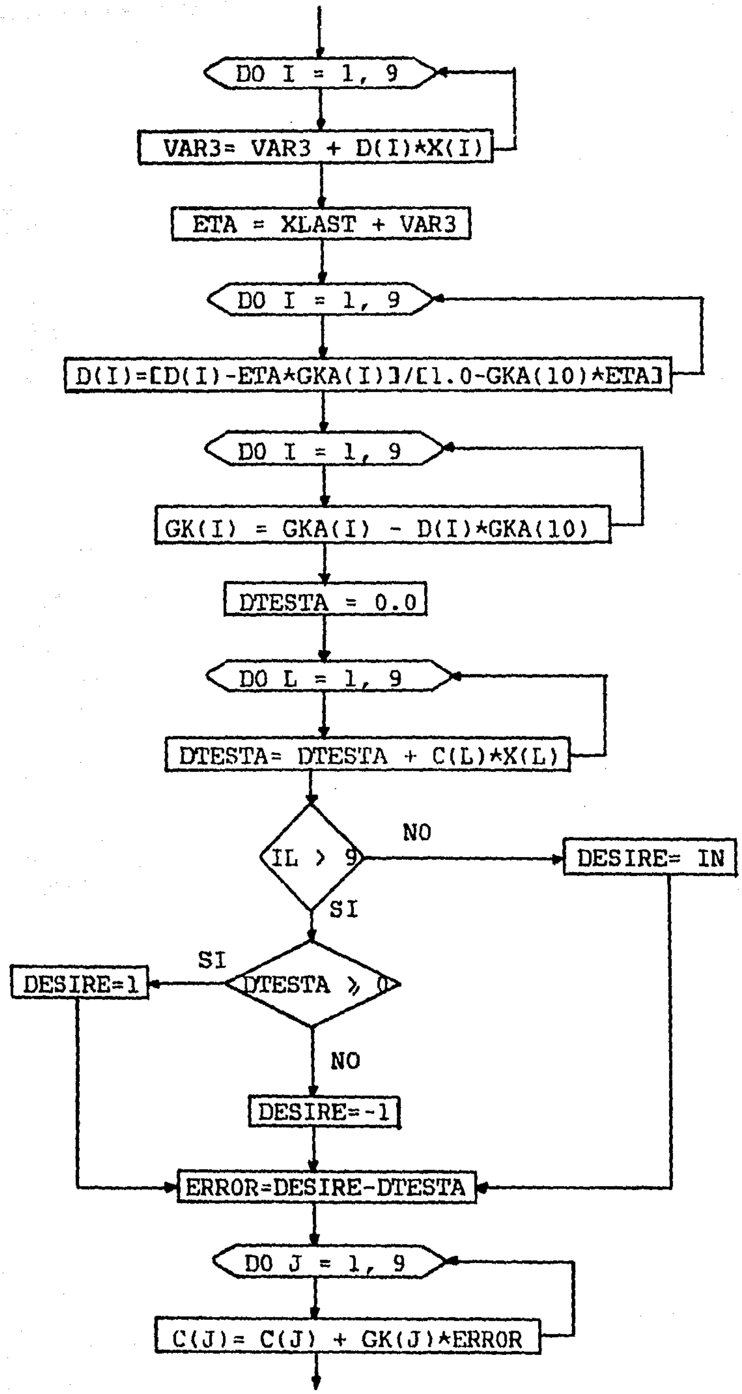
El programa correspondiente al algoritmo rápido de Kalman se obtiene mediante el uso de las ecuaciones (2.3.4), (3.2.2), (3.2.16), (3.3.1), (3.3.4), (3.3.5), (3.3.15), (3.3.17), (3.3.23) y (3.3.24).

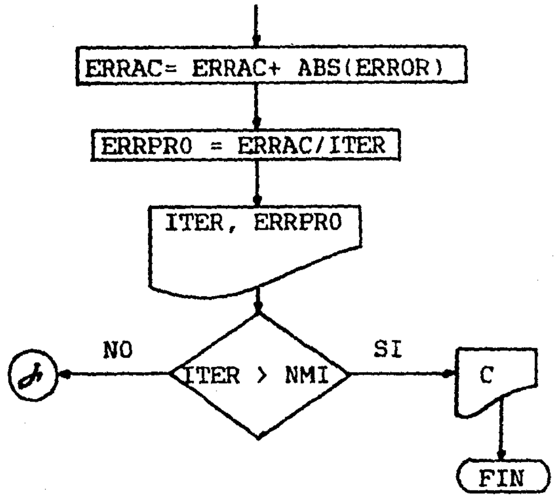
Las condiciones iniciales son con todos los vectores (C, GK, A, D y X) puestos a cero; el error residual ERRESI = .01, y considerando un valor de λ =0.999.

El diagrama de flujo para este algoritmo es




VAR1=VAR1+A(J)*X(J)
EPREDE= XIN+VAR1
DO I = 1, 9
A(I)= A(I)-GK(I)*EPREDE
VAR2 = 0.0
DO J = 1, 9
VAR2= VAR2 + A(J)*X(J)
EPRIMA= XIN + VAR2
ERRESI=AMBDA*ERRESI+EPRIMA*EPREDE
GKA(1)= EPRIMA/ERRESI
DO I = 1, 9
GKA(I+1)= GK(I)+ A(I)*GKA(1)
XLAST= X(9)
DO I = 8,1,-1
X(I+1)= X(I)
X(1)= XIN
VAR3 = 0.0


DO I = 1, 9
VAR3= VAR3 + D(I)*X(I)
ETA = XLAST + VAR3
DO I = 1, 9
D(I)=[D(I)-ETA*GKA(I)]/[1.0-GKA(10)*ETA]
DO I = 1, 9
GK(I) = GKA(I) - D(I)*GKA(10)
DTESTA = 0.0
DO L = 1, 9
DTESTA= DTESTA + C(L)*X(L)
IL > 9
NO
DESIRE= IN
SI
SI
DESIRE=1
DTESTA ≥ 0
NO
DESIRE=-1
ERROR=DESIRE-DTESTA
DO J = 1, 9
C(J)= C(J) + GK(J)*ERROR



Definición de variables

AMBDA= λ

IN= Simbolo transmitido

XIN= Muestra distorsionada

X(I)= Vector de muestras distorsionadas

C(I)= Vector de coeficientes de igualación

DTESTA= Estimación del simbolo transmitido a la salida del igualador

DESIRE= Valor al que se cuantiza suponiendo DTESTA como correcta

VAR1,2 y 3= Variables auxiliares

EPREDE= Error de predicción directa antes de actualizar A(I)

A(I)= Vector de predicción directa

EPRIMA= Error de predicción directa después de actualizar A(I)

ERRESI= Error residual

GKA(I)= Vector aumentado de Kalman

XLAST= Ultimo elemento del vector X(I)

ETA= Error de predicción de retraso

D(I)= Vector de predicción de retraso

GK(I)= Vector de Kalman

ERROR= Diferencia entre la estimación y su cuantización

ERRAC= Error acumulado total

ERRPRO= Error promedio total

ITER= Contador de muestras procesadas

El listado correspondiente a este algoritmo es

```
C     --------------------------------------------------------
C     ! ESTE PROGRAMA CALCULA LOS COEFICIENTES DEL IGUALADOR !
C     ! ADAPTIVO QUE UTILIZA  EL ALGORITMO  RAPIDO DE KALMAN.!
C     ! PIDE LA CONSTANTE λ (.9<λ<1)  Y  EL NUMERO DE  !
C     ! MUESTRAS A PROCESAR (N<99999).                       !
C     --------------------------------------------------------
C
      DIMENSION A(20),GK(20),GKA(20),X(20),EME(20)
      DIMENSION VA1(20),VA2(20),C(20),D(20),VA3(20)
      DOUBLE PRECISION ERRESI,DTESTA,EPRIMA,ETA,RMU
      DOUBLE PRECISION ERROR,VAR1,EPREDE,VAR2,DESIRE
      DOUBLE PRECISION A,C,GK,K,GKA,VA1,VA2,D,VAR3,XLAST
      OPEN(80,NAME='SALE.DAT',TYPE='NEW',RECORDSIZE=300)
      OPEN(77,NAME='SENAL.DAT',TYPE='OLD',RECORDSIZE=300)
C     INICIALIZA VECTORES C, X, GK, A, Y D
 253  DO 1 I=1,9
      C(I)=0.0
      GK(I)=0.0
      A(I)=0.0
      D(I)=0.0
      X(I)=0.0
    1 CONTINUE
C     DEFINE VALOR INICIAL DEL ERROR RESIDUAL
      ERRESI=0.01
C     PIDE LA CONSTANTE λ
      TYPE 4
```

```
      ACCEPT 2,AMBDA
C     PIDE NUMERO DE MUESTRAS A PROCESAR
      TYPE 141
      ACCEPT 122,NMI
      ERROR=0.0
      IL=0
      MAX=0
   33 ITER=0
C     INICIA CICLO DE IGUALACION
    3 IL=IL+1
      ITER=ITER+1
C     LEE MUESTRA DISTORSIONADA Y SIMBOLO
C     TRANSMITIDO   CORRESPONDIENTE
      READ(77,20)IV,VO
C     CALCULA ERROR DE PREDICCION DIRECTA
      VAR1=0.0
      DO 40 J=1,9
      VA1(J)=A(J)*X(J)
   40 VAR1=VAR1+VA1(J)
      EPREDE=VO+VAR1
C     ACTUALIZA VECTOR DE PREDICCION DIRECTA
      DO 50 I=1,9
   50 A(I)=A(I)-GK(I)*EPREDE
C     ACTUALIZA ERROR DE PREDICCION DIRECTA
      VAR2=0.0
      DO 60 J=1,9
      VA2(J)=A(J)*X(J)
   60 VAR2=VAR2+VA2(J)
      EPRIMA=VO+VAR2
C     CALCULA EL ERROR RESIDUAL
      ERRESI=AMBDA*ERRESI+EPRIMA*EPREDE
C     ACTUALIZA EL VECTOR AUMENTADO DE KALMAN
      GKA(1)=EPRIMA/ERRESI
      DO 70 I=1,9
   70 GKA(I+1)=GK(I)+A(I)*GKA(1)
      XLAST=X(9)
C     RECORRE EL VECTOR X
      DO 90 I = 16,1,-1
   90 X(I+1)=X(I)
      X(1)=VO
C     CALCULA EL ERROR DE PREDICCION DE RETRASO
      VAR3=0.0
      DO 110 I=1,9
      VA3(I)=D(I)*X(I)
  110 VAR3=VAR3+VA3(I)
      ETA=XLAST+VAR3
C     ACTUALIZA EL VECTOR DE PREDICCION DE RETRASO
      DO 120 J=1,9
  120 D(J)=(D(J)-ETA*GKA(J))/(1.0-GKA(10)*ETA)
      DO 130 I=1,9
C     ACTUALIZA EL VECTOR DE KALMAN
  130 GK(I)=GKA(I)-D(I)*GKA(10)
```

```
C     EJECUTA CONVOLUCION DE C*X
      DTESTA=0.0
      DO 155 L=1,9
  155 DTESTA=DTESTA+C(L)*X(L)
      IF(IL.GT.9)GO TO 190
C     SECUENCIA DE ENTRENAMIENTO
      DESIRE=IV
      GO TO 210
C     FIN DE SECUENCIA DE ENTRENAMIENTO.
C     DETERMINA SIGNO DE LA ESTIMACION
C     RESULTANTE DE LA CONVOLUCION
  190 IF(DTESTA-0.0) 200,300,300
  200 DESIRE=-1.0
      GO TO 210
  300 DESIRE=1.0
C     CALCULA EL ERROR POR MUESTRA Y EL
C     ERROR ACUMULADO
  210 ERROR=DESIRE-DTESTA
      ERRAC=ERRAC+ABS(ERROR)
C     ACTUALIZA COEFICIENTES
      DO 160 J=1,9
  160 C(J)=C(J)+GK(J)*ERROR
      NUMUES=ITER+(MAX*50)
      ERRPRO=ERRAC/NUMUES
C     ESCRIBE EN EL ARCHIVO SALE.DAT
C     EL NUMERO DE MUESTRAS PROCESADAS
C     Y EL ERROR PROMEDIO TOTAL
      WRITE(80,*)NUMUES,ERRPRO
  140 IF(ITER.LT.50)GO TO 3
      MAX=MAX+1
      IF(MAX*50.GE.NMI)GO TO 111
C     REINICIALIZA ARCHIVO DE DATOS DE ENTRADA
      CLOSE(77)
      OPEN(77,NAME='SENAL.DAT',TYPE='OLD',RECORDSIZE=200)
      GO TO 33
C     MUESTRA COEFICIENTES EN PANTALLA
  111 DO 1001 IMP=1,9
 1001 PRINT 1010,IMP,C(IMP)
      CLOSE(77)
      CLOSE(80)
      STOP
    2 FORMAT(F6.3)
    4 FORMAT(7H AMBDA=,$)
   20 FORMAT (I3,F8.4)
  122 FORMAT(I5)
  141 FORMAT(' NUMERO DE MUESTRAS=',$)
 1010 FORMAT('  C(',I2,')= ',F10.6)
      END
```

Mensajes a la terminal del usuario

Pide el valor de la constante λ, a lo que se debe contestar tecleando un número de preferencia en el rango $0.9 \leqslant \lambda \leqslant 1$.

También pide el número de muestras a procesar. Este debe ser, como en el caso de los algoritmos anteriores, un numero entero máximo de 99999.

La figura 4.5 muestra una gráfica comparativa de la convergencia inicial de los algoritmos gradiente, rápido de Kalman y de coeficientes óptimos fijos

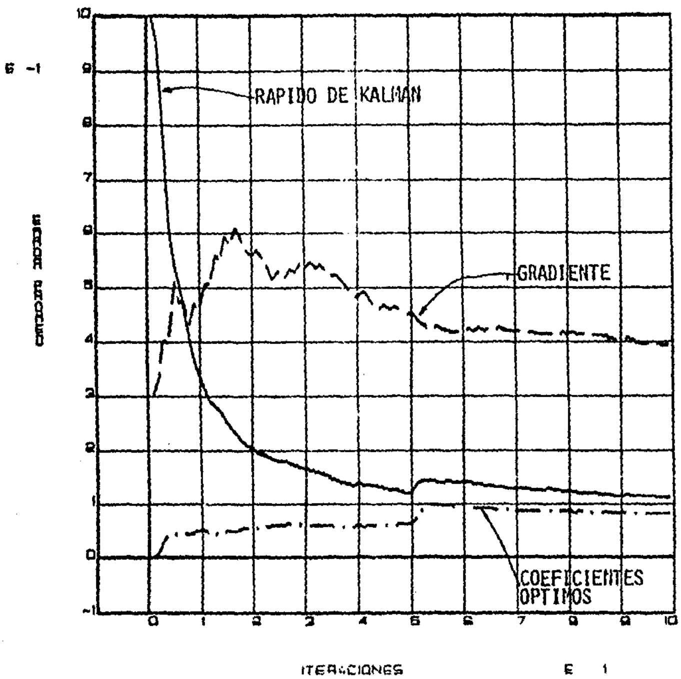


Figura 4.5 Comparación de la convergencia inicial de los algoritmos gradiente, rápido de Kalman y de coeficientes óptimos fijos utilizando nueve coeficientes.

Para distorsionar la señal de entrada al igualador se utilizaron dos filtros paso-bajas en serie, uno es un filtro RC y el otro un filtro digital con ciclo de trabajo de 0.5 (los programas de ambos filtros se encuentran en el apéndice A). Los algoritmos rápido de Kalman y gradiente se corrieron con una secuencia de entrenamiento de 18 muestras; el gradiente utilizando $\Delta= 0.02$ y el rápido de Kalman con $\lambda= 0.999$. Además se debe mencionar que para el algoritmo rápido de Kalman se utilizó aritmética de doble precisión, ya que al hacer pruebas para este algoritmo con $\lambda<1$ y precisión simple los elementos del vector C se incrementaron ilimitadamente produciendo inestabilidad del algoritmo.

En la figura 4.5 se puede observar que siempre existirá un pequeño error en la estimación resultante de la convolución de C con X, inclusive con el igualador de coeficientes óptimos; este error no es posible de eliminar prácticamente, ya que para lograrlo se requiere de una longitud infinita del vector C. Además se nota en esta figura que, como ya se había mencionado en el capitulo anterior, el algoritmo rápido de Kalman tiene una velocidad de convergencia inicial mayor que la del gradiente.

4.4 Elección del algoritmo de igualación para su aplicación práctica

Para la elección del algoritmo a emplear en el igualador adaptivo prácticamente se tomó en consideración lo siguiente

Como se puede calcular de la tabla 3.1, para un vector C con 9 coeficientes el algoritmo gradiente realizará 18 multiplicaciones y 18 adiciones; el algoritmo rápido de Kalman 94 multiplicaciones y 113 adiciones. Además, el rápido de

Kalman involucra divisiones, las cuales llevan mayor tiempo de procesamiento en su ejecución.

El microprocesador elegido para ser utilizado en el igualador es el TMS32010 (ver apéndice B) este microprocesador alcanza a ejecutar hasta 5 millones de operaciones por segundo (200 ns por operación) incluyendo multiplicaciones, por tanto tardará 7.2 µs en procesar las 36 operaciones del algoritmo gradiente y más de 41.4 µs en las del rápido de Kalman, es decir, este último tardará aproximadamente 6 veces lo que el gradiente en hacer una iteración.

Con lo anterior, el algoritmo gradiente compensa en cierta medida su lenta convergencia respecto al rápido de Kalman, pero con mayor simplicidad.

Además, debemos tomar en cuenta que el algoritmo rápido de Kalman requiere de mayor número de localidades de memoria de datos que el gradiente, ya que tiene que almacenar los vectores C, X, K_N, K_M, A y D lo cual involucra más ciclos de máquina al llevar y traer datos de memoria, aumentando así el tiempo total de procesamiento.

Finalmente, la última consideración es que el algoritmo rápido de Kalman necesita manejar una gran precisión en los cálculos, lo cual ocaciona que en un microprocesador se tenga que manejar aritmética de doble precisión, redundando así en mayor tiempo de procesamiento. Por todo lo descrito anteriormente, se eligió el algoritmo gradiente para el funcionamiento del igualador adaptivo real. El programa en el lenguaje ensamblador del microprocesador TMS32010 correspondiente a este algoritmo, así como las pruebas en tiempo real se detallan en el capítulo siguiente.

CAPITULO 5

PROGRAMACION Y PRUEBAS EN TIEMPO REAL DEL ALGORITMO GRADIENTE

Introducción

Habiendo elegido el algoritmo gradiente para el desarrollo del igualador adaptivo, el siguiente paso es su programación en lenguaje ensamblador del TMS32010 [10] y posteriormente la realización de las pruebas de funcionamento en tiempo real del igualador.

En este capitulo se presentan las partes principales del programa en lenguaje ensamblador, incluyendo una breve explicación de cada una de las instrucciones que ejecuta.

Se muestran también las figuras de las señales obtenidas a la entrada y la salida del igualador para las velocidades de transmisión de los datos de 4800 y 9600 bits/seg.

5.1 Documentación del programa

Para hacer la programación en lenguaje ensamblador, se asignó a cada una de las variables del algoritmo una localidad dentro de la memoria de datos del TMS32010. El mapa de memoria resultante es el que se muestra a continuación

localidad de memoria de datos (MD) decimal	dato
0	x_1
8	x_9
10	c_1
18	c_9
20	estimacion
21	simbolo
22	error
23	(delta)(error)
25	max=>1000
26	min=>F000
27	delta=0052
29	1
30	0
35	contador de muestras
36	error acumulado
37	No. de muestras=100

38	limite de error
40	error acumulado-limite
45	(delta)(error)(x_i)

Donde

x_1 a x_9	vector de muestras de entrada al igualador
C_1 a C_9	vector de coeficientes del igualador
estimación	resultado de la convolución de los vectores C y X.
símbolo	valor al que se cuantiza la estimación
error	error por muestra, resultado de la diferencia símbolo - estimación.
max	numero considerado como "+1" en Q_{12} (ver apéndice C para formatos Q).
min	numero considerado como "-1" en Q_{12}
delta	constante "Δ" del algoritmo.

contador de muestras
: es un numero que va contando las muestras procesadas hasta completar cien, en cuyo caso regresa a cero y empieza a contar nuevamente.

error acumulado
: es una localidad donde se acumula el error de 100 muestras en formato Q_5.

numero de muestras
: numero máximo al que llega el contador de muestras antes de comparar el error acumulado con el valor de error límite.

límite
: es el valor fijado como límite máximo del error acumulado sin que se considere necesario actualizar el vector de coeficientes.

Las instrucciones en lenguaje ensamblador que realizan las operaciones del algoritmo gradiente se explican a continuación [11]. Para esta explicación se consideran las siguientes abreviaciones: ACC =Acumulador; MDx=Localidad x de la memoria de datos; REGP=Registro P; REGT=Registro T; ARO,1= registros auxiliares 0 y 1.

Para proporcionar las condiciones iniciales y colocar las constantes 1 y 0 en las localidades 29 y 30

```
INICIO   LDPK   0       APUNTA A LA PAGINA CERO DE MD
         LACK   1       ACC= 1
         SACL   29      ACC → MD29
         ZAC            ACC= 0
         LARK   0,18    ARO= 18
         LARP   0       ARO= APUNTADOR ACTUAL
CEROS    SACL   *       ACC → MD APUNTADA POR ARO
         BANZ   CEROS   ARO=ARO-1; SI ARO≠ 0 SALTA A CEROS
         SACL   30      ACC → MD30
         LAC    25      ACC= +1 EN Q12
         SACL   10      HACE  C1= +1 EN Q12
```

Accesa una muestra por el puerto 2 en Q_{15} y la convierte a Q_{12} .

```
WAIT     IN     0,2     ENTRA MUESTRA POR PUERTO 2 A MDO
         LAC    0,13    ACC= MDO RECORRIDA 13 BITS
         SACH   0       ACC → MDO
```

Hace la convolución de los vectores C y X y recorre el vector X

```
         LARK   0,8     ARO = 8 (APUNTA A X9)
         LARK   1,18    AR1 = 18 (APUNTA A C9)
```

```
        ZAC           ACC = 0
        LARP   0      AR0 = APUNTADOR ACTUAL
        MPY    30     REGP = 0
LOOP    LTD    *,1    REGP+ACC→ACC; x(i+1)=x(i);
                      x(i)→T
        MPY    *-,0   x(i)c(i) → REGP
        BANZ   LOOP   AR0=AR0-1; SI AR0≠0 SALTA A LOOP
        APAC          REGP + ACC → ACC
        SACH   20,4   ACC → MD20 RECORRIDO 4 BITS
```

Determina el signo de la estimación y lo almacena en MD21

```
        LAC    20     ESTIMACION → ACC
        BGEZ   GAMA   SI ACC > 0 SALTA A GAMA
        LAC    26     -1 → ACC
        B      TETA   SALTA A TETA
GAMA    LAC    25     +1 → ACC
TETA    SACL   21     ACC → MD21
```

Calcula error y lo almacena en MD22

```
        SUB    20     SIGNO DE ESTIMACION-ESTIMACION→ACC
        SACL   22     ACC → MD22
```

Convierte signo de la estimación a Q_{14} y lo saca por el puerto 2

```
        LAC    21,2   MD21 → ACC CORRIDO 2 BITS
        SACL   21     ACC → MD21
        OUT    21,2   MD21 → PUERTO 2
```

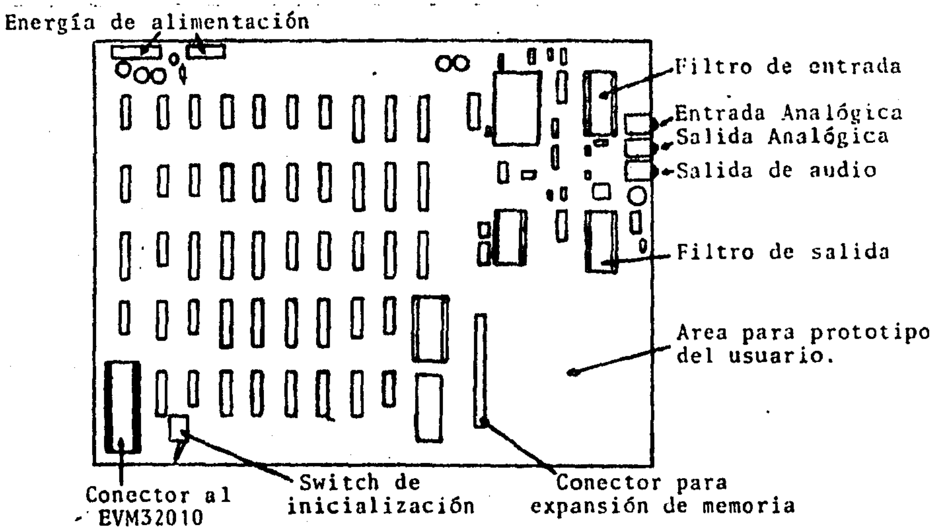
Actualiza coeficientes y regresa a tomar otra muestra

```
        LARK   0,9    AR0= 9  (APUNTA A X9)
        LARK   1,18   AR1= 18 (APUNTA A C9)
        LT     22     ERROR → REGT
        MPY    27     REGT*Δ → REGP
        PAC           P → ACC
        SACH   23,4   ACC → 23 CORRIDO 4 BITS
        LT     23     T= Δ*ERROR
        LARP   0      AR0=APUNTADOR ACTUAL
```

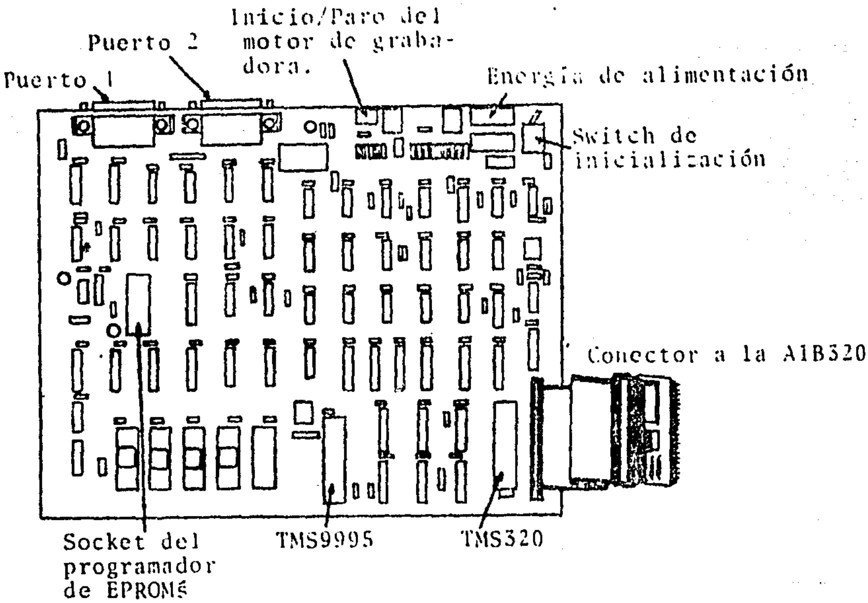
```
ALFA    MPY    *,1      T*Ci      →  REGP
        PAC              P  →  ACC
        SACH   45,4     ACC  →  MD45 CORRIDO 4 BITS
        LAC    45       ACC=  MD45
        ADD    *        ACC=   Ci Ci  +  MD45
        SACL   *-,0,0   ACC  →  Ci
        BANZ   ALFA     AR0=AR0-1; SI AR0=0  SALTA A ALFA
        B      WAIT     SALTA A TOMAR OTRA MUESTRA
```

5.2 Procedimiento de prueba

Este programa se probó empleando el sistema de emulación EVM32010 y la tarjeta de interface analógica AIB32010 que se muestran en la figura 5.1.



(a) Tarjeta de Interface Analógica (AIB)



(b) Módulo de Evaluación (EVM)

Figura 5.1 Tarjetas del sistema de emulación 32010

En esta figura, la AIB (Analogic Interface Board) es la tarjeta de interface analógica que permite tomar muestras de una señal (con una frecuencia programable entre 34.48 KHz y 76.29 Hz) y las convierte en un número binario que entrega al EVM para que sean procesadas por el TMS32010. Esta señal de datos debe tener niveles máximos de voltaje de ±10V [13].

El EVM (Evaluation Module) es un sistema de evaluación con el cual es posible simular el comportamiento del TMS32010 en condiciones reales. Esta tarjeta tiene dos puertos con conectores RS-232 mediante los cuales es posible enlazarse con una terminal de computadora (puerto P1), y con el CPU de algún sistema de cómputo mayor (puerto P2) [12].

El procedimiento seguido para la simulación en tiempo real es el siguiente:

Se conectan las tarjetas AIB y EVM a una terminal de computadora VT100 y a una computadora VAX/VMS como muestra la figura 5.2

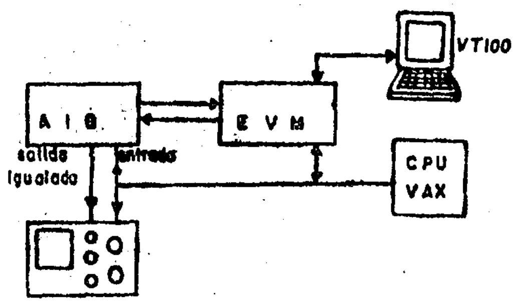


Figura 5.2 Sistema empleado para la simulación en tiempo real del algoritmo gradiente.

Este sistema permite por un lado, la programación del TMS32010 a través de archivos editados en la computadora central, y por otro, utilizar como señal de entrada al igualador señales provenientes de la VAX, distorsionadas por un canal específico de comunicación.

Mediante las instrucciones BAUD1 y BAUD2 se ajustan las velocidades de transmisión y recepción de los puertos 1 y 2 del EVM, haciendo así que sean compatibles las señales de la VAX (normalmente de 2400 bits/seg), las de la terminal y las del EVM.

El programa de igualación se editó en la computadora VAX haciendo uso del comando EDIT/EDT de tal sistema, ya que este editor es más completo que el incluido en el EVM. Este programa se dividió en tres archivos para poder ensamblarlo en forma adecuada. El primero de ellos, contiene las primeras instrucciones del programa y debe incluir como primer caracter el signo ">" para indicar al EVM que inicia el programa. El último archivo, que contiene las últimas instrucciones del programa debe contener como último caracter el signo "<" para indicar el fin del programa.

El procedimiento de ensamblado del programa es el siguiente:

1.- Estando comunicada la terminal VT100 con el EVM, se teclea el comando ASM 2, que le indica al EVM que ensamble la información que reciba por el puerto 2 (No presionar RETURN en este momento).

2.- Se presionan las teclas "CONTROL" y "C" simultáneamente (CTRL/C), con lo cual se establece la comunicación entre la terminal y la VAX.

3.- Se pide a la VAX que despliegue el primer archivo del programa mediante el comando TYPE (No presionar RETURN).

4.- Presionar nuevamente CTRL/C para que el EVM envie a la VAX el código equivalente a RETURN y al mismo tiempo inicie el ensamblado.

5.- Al terminar de ensamblar este archivo el TMS se detiene, entonces deben repetirse los pasos 2 a 4 para los otros dos archivos del programa.

Una vez que el programa se tiene en la memoria del EVM se procede a ejecutarlo utilizando el comando RUN cuando se desea ejecutar sin puntos de interrupción (en cuyo caso se puede detener la ejecución accionando el switch de reset del EVM), o con el comando EX para ejecutar solo una parte del programa.

Para colocar los puntos de interrupción en el programa se emplea el comando SB N (Set Breackpoints) donde N es un número entre 1 y 8 que corresponde a uno de los puntos de interrupción disponibles en el EVM. Para limpiar estos puntos se debe utilizar el comando CB N (Clear Breackpoints).

En caso de ser necesario conocer el contenido de la memoria de datos o de programa del TMS en algún instante determinado, es posible usar los comandos DDM M N (Display Data Memory) y DPM M N (Display Program Memory) respectivamente, con los que mostrará en la pantalla de la terminal el contenido de la memoria de datos o de programa desde la localidad M hasta la localidad N. Si se desea cambiar el contenido de alguna localidad de memoria, se tendrá que utilizar alguno de los comandos MDM N (Modify Data Memory) o MPM N (Modify Program Memory) con los cuales mostrará el contenido de la localidad N de memoria solicitada y dejará el cursor frente a éste por si se requiere hacer algún cambio.

Para obtener la señal de datos distorsionada por un canal real, se pide a la VAX que despliegue algún archivo y esta señal se alimenta a la entrada analógica de la AIB. Tanto la señal distorsionada de entrada como la señal procesada de salida son alimentadas a las entradas de un osciloscópio.

5.3 Resultados obtenidos

El tiempo que tarda el TMS32010 en ejecutar las instrucciones del programa es de aproximadamente 40 µs, por lo cual alcanzará a procesar 24500 muestras por segundo. Con lo anterior tenemos que para las velocidades de 4800 y 9600 bits/seg de la señal de datos recibida, podrá tomar varias muestras por bit, funcionando así como un igualador fraccionalmente espaciado; sin embargo, como la frecuencia con que toma las muestras no es múltiplo exacto de la frecuencia de la señal de datos, existe la posibilidad de que tome muestras en los instantes de transición de los datos (de +1 a -1 o de -1 a +1), registrando así errores incorrectos y consecuentemente reajustando el vector de coeficientes para compensar tal error. Esto ocacionaría que el vector C esté cambiando constantemente aún cuando el canal no varíe.

Se vieron dos formas posibles de evitar que este error influyera en la actualización de los coeficientes. Una es haciendo que el TMS32010 tome muestras a una frecuencia que sea múltiplo de la señal de datos. Los inconvenientes de esta solución son de que para tomar muestras a una frecuencia múltiplo de la señal, se debe sincronizar de manera que no se tomen muestras en el momento de trancisión de los datos. El otro inconveniente de esta solución es que el programa no podría aplicarse a velocidades que no sean submúltiplos de la frecuencia a la que éste toma muestras.

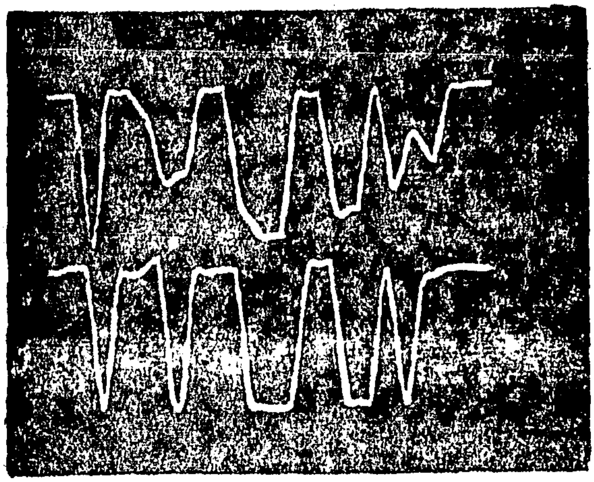
La otra solución al problema es la de considerar un error promedio en vez del error por muestra, de esta forma son compensados los picos de la señal de error ocurridos al tomar muestras en los instantes de trancisión de los datos.

Esta última es la solución que se eligió para este caso. Este cambio en el criterio de error se logró introduciendo un contador de muestras y un nivel máximo como limite del error acumulado; así, se procesan N muestras en cada una de las cuales se calcula el error y se acumula; una vez acumulado el error de las N muestras se compara con el valor limite y se decide si es necesario o no hacer algún cambio en el vector de coeficientes. El programa completo (incluyendo numero de instruccion, localidad de la memoria de programa, código de operación y el mnemónico de cada instruccion) se lista en el apéndice A.

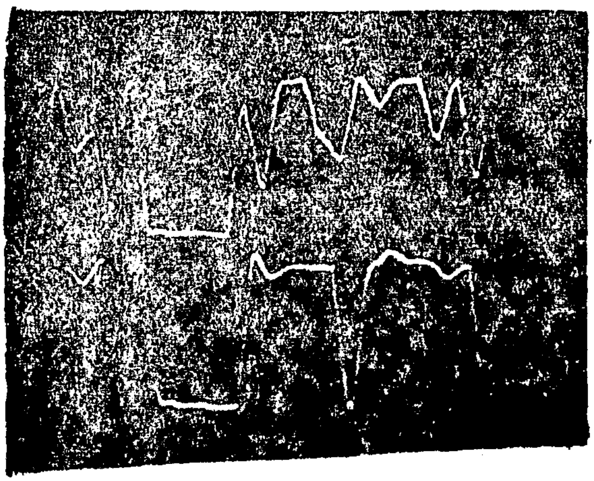
Las formas de onda obtenidas para las señales con velocidad de 4800 y 9600 b/seg se muestran en la figura 5.3.

Como se puede observar en estas figuras, las señales de entrada al igualador (señal de la parte superior de cada una de las figuras) tienen niveles muy diversos por lo cual, en algunos momentos, no es posible definir directamente si el símbolo transmitido correspondiente a la muestra tomada fue un "+1" ó un "-1" lógico. Por otro lado, se nota que la señal de salida del igualador (señal de la parte inferior de cada figura) presenta niveles bien definidos, por lo cual la decisión de "+1" ó "-1" se puede hacer fácilmente.

De lo anterior, se puede concluir que el algoritmo de igualación utilizado reduce considerablemente la interferencia entre símbolos de la señal recibida.



(a) 4800 b/seg



(b) 9600 b/seg

Figura 5.3 Señales de entrada y salida del igualador basado en el algoritmo gradiente.

Para la construcción física del igualador, se sugiere la configuración mostrada en la figura 5.4.

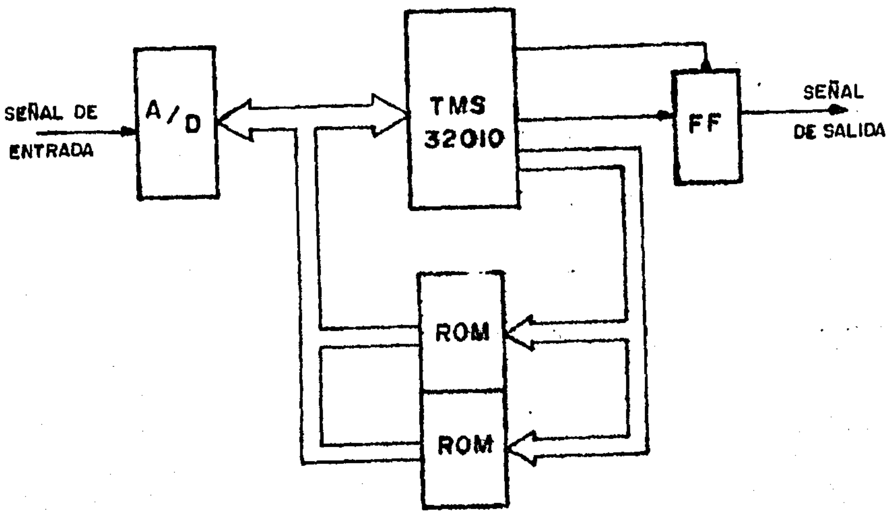


Figura 5.4 Configuración propuesta para la construcción física del igualador

En este caso, la señal de entrada es digitalizada por el convertidor analógico-digital (A/D) y mediante el bus de datos es llevado al TMS32010, el cual lee el dato y ejecuta las instrucciones del algoritmo almacenado en la memoria ROM (formada por los dos bloques inferiores de la figura); se utilizan dos circuitos integrados ROM debido a que el bus de datos del sistema es de 16 bits, y las memorias ROM, PROM y EPROM actales son de 8 bits. Algo que es importante destacar, es que los circuitos de memoria a utilizar deben de tener un tiempo de acceso menor a 100 nanosegundos.

La salida igualada se obtendrá enviando el valor adecuado al bus de datos, de manera que al tomar solo una de sus líneas a través de un flip-flop, la salida siga la secuencia de bits correspondiente a la estimación obtenida.

CONCLUSIONES

En este trabajo se ha analizado el hecho de que para poder aprovechar al máximo posible el ancho de banda de los canales en la transmisión de señales digitales en banda-base, resulta conveniente la colocación de redes de igualación en serie con el canal para eliminar la interferencia entre símbolos introducida por éste. Además se han analizado y comparado algunos de los algoritmos que se utilizan en igualación adaptiva, de entre los cuales se eligió el gradiente para el desarrollo práctico del igualador adaptivo.

Las instrucciones correspondientes a este algoritmo son ejecutadas por el microprocesador de señales digitales TMS32010. En el capitulo 5 se mostraron las formas de las señales de entrada y salida al igualador, en las cuales se notó la eficiencia del algoritmo empleado.

El programa presentado aquí, puede aún ser modificado para lograr un funcionamiento óptimo en alguna aplicación con una velocidad especifica. Algunas de las variaciones que se sugieren son:

1.- Aumentar el tamaño del vector C de coeficientes de igualación para obtener estimaciones más cercanas al valor correcto. Esta variante trae consigo el inconveniente de que al aumentar el número de coeficientes del igualador, se incrementa el numero de operaciones realizadas en el algoritmo y consecuentemente el tiempo de ejecución. Todo esto ocaciona que el número de muestras procesadas por unidad de tiempo se vea reducido. Pese a lo anterior, si la velocidad de transmisión de la señal a igualar es relativamente pequeña respecto a la frecuencia de muestreo del igualador, el aumento en el tamaño de C mejorará el rendimiento del igualador.

2.- Cambiar el número de muestras y el valor límite considerados en el criterio de error utilizado. Esta variante, al igual que la anterior, dependerá de la relación entre la velocidad de transmisión de la señal de datos y la frecuencia de muestreo del igualador.

3.- En caso necesario es posible cambiar el nivel de referencia de comparación de la estimación, que en este caso se ha tomado de cero considerando la señal de datos bipolar NRZ (No Retorno a Cero).

4.- Cambiar el nivel de salida del igualador según convenga, ya que en la forma actual será de aproximadamente 5V que corresponden al número digital en formato Q_{12} que se envía a la salida por el puerto 2.

5.- Muestrear de una manera síncrona para poder procesar menos muestras por bit (hasta 1 muestra por bit) logrando así ser aplicable para señales de frecuencia mayor a las analizadas aquí.

Una gran ventaja que presenta el igualador desarrollado en este trabajo es que no requiere secuencia de entrenamiento durante su período inicial de funcionamiento, es decir, no es necesario conocer la secuencia inicial de muestras cuando el igualador empieza a trabajar.

En la actualidad se siguen estudiando métodos y algoritmos aplicables a la igualación adaptiva. Simultáneamente, la tecnología VLSI continúa avanzando en forma acelerada en el área del procesamiento digital de señales (por ejemplo el surgimiento del TMS32020). Por lo anterior, El igualador desarrollado podrá ser modificado y actualizado para obtener mayor velocidad de procesamiento y consecuentemente un mejor rendimiento.

REFERENCIAS

1.- MISCHA SCHUARTZ
"Transmisión de información, Modulación y Ruido"
Mc Graw-Hill, México 1983

2.- A. BRUCE CARLSON
"Sistemas de Comunicación"
Mc Graw-Hill, México 1980

3.- JOHN G. PROAKIS
"Digital Communications"
Mc Graw-Hill, Tokyo Japan 1983

4.- DEGEM SYSTEM LTD
"Experimentos en Comunicaciones Modernas"
Curso COM-6 DEGEM SYSTEM

5.- B. P. LATHI
"Introducción a la Teoría y Sistemas de Información"
Limusa, México 1980

6.- R.W. LUCKY, J. SALZ AND E.J. WELDON Jr.
"Principles of data communication"
Mc. Graw Hill, New York 1968

7.- SHAID QURESHI
"Adaptive equalization"
IEEE Communications Magazine, pp. 9-16, March 1982

8.- M.S. MUELLER
"Least-Squares Algorithms for Adaptive Equalizers"
The Bell System Technical Journal, Vol. 60, No. 8
USA October 1981

9.- DAVID D. FALCONER AND LENNART LJUNG
"Aplication of Fast Kalman Estimation to Adaptive Equalization", IEEE Transactions on Communications, Vol.COM-26, No. 10, October 1978

10.-ANITA SEELING
"Telephone Management Systems Bring a New Range of Applications to Personal Computers"
Computers & Electronics, April 1984 pp. 56-59

11.-TEXAS INSTRUMENTS, INC
"TMS32010 Users Guide"
Dallas Texas, Mayo 1983

12.-TEXAS INSTRUMENTS, INC
"TMS32010 Evaluation Module"
Texas 1984

13.-TEXAS INSTRUMENTS, INC
"TMS32010 Analog Interface Board Users Guide"
Texas 1984

Bibliografiá

- P. BYLANSKI AND D.G.W. INGRAM
  "DIgital Transmission Systems"
  IEE Telecomunications series 4, Great Britain 1980

- CHRISTOPH MUNCH
  "Igualador Adaptivo para Transmisión Digital en Banda Base", Centro de Investigación y Estudios Avanzados del IPN, Departamento de Ingeniería Electrica, México Septiembre 1984

- THOMAS J. APRILLE
  "Filtering and Ecualization for Digital Transmission"
  IEEE Communications Magazine, pp. 17-29, March 1983

- W.R. BENNET
  "Data Transmission"
  Bell System, Tech.J.

- T.K. MILLER AND S.T. ALEXANDER
  "An Implementation of the LMS Adaptive Filter Using an SIMD Multiprocessor Ring Architecture"
  Center for Comunications & Signal Processing
  IEEE International Conference on Acoustics, Speech and Signal Processing, Vol 1, Tampa Florida 1985

- COLE ERSKINE, SURENDAR MAGAR, EDWARD CAUDEL AND DANIEL ESSIG
  "Architecture and applications of a second-generation digital signal processor"
  Texas Instruments Inc., Houston Texas 1985

- HIGH SPEED SIGNAL PROCESSORS
  Wordś Largest Computer Magazine
  Computers& Electronics pp.60-61 y 106, April 1984

- CHARLES ROGERS, KWANG-SHIK MIN, STEVEN SPEIER AND JHON WITSON, "A Transportable TMS32010 Signal Processing System", East Texas State University
  IEEE International Conference on Acoustics, Speech and Signal Processing, Florida 1985

- IEEE ACOUSTICS, SPEECH & SIGNAL PROCESSING SOCIETY
  "VLSI Signal Processing"
  United States 1984

- DAVID P. MORGAN AND HARVEY F. SILVERMAN
"An Investigation into the Efficiency of a Paralel TMS320 Architecture"
DFT and Speech Filterbanck applications
IEEE International Conference on Acoustics, Speech and Signal Processing, Florida 1985

- M. T. JONG
"Methods of Discrete Signal and Systems Analysis"
Mc Graw-Hill, E.U. 1982

- JOHN BELLAMY
"Digital Telephony"
John Wiley-& Sons. USA 1982

- DONALD E. MURPHY & STEPHEN A. KALLIS JR.
"Introducción a la Comunicación de Datos"
Publicaciones Telecomex, VIII-1979.

APENDICE A

PROGRAMAS

PROGRAMA QUE SIMULA UN FILTRO PASO-BAJAS RC

Este programa simula un filtro paso-bajas RC. Recibe muestras no distorsionadas (+1 y -1) y entrega a su salida muestras de una señal sin componentes de alta frecuencia (N muestras de salida por cada bit leido). La ecuación que rige su comportamiento es

$$v_o(t) = [v_o(t-1)-v_i]\, e^{-2\pi f k\Delta t} + v_i$$

donde

$v_o(t)=$ voltaje de salida

$v_i(t)=$ voltaje de entrada

$f=1/RC=4.775$ KHz

$k\Delta t=k(40\mu s)$, $k=1,2,..,N;$ $N<5$

El listado del programa se muestra a continuación

```
C       ------------------------------------------------------
C       !                                                    !
C       ! PROGRAMA CORRESPONDIENTE AL FILTRO PASO-BAJAS RC.  !
C       ! DEBE DARSELE COMO  DATOS  EL  NUMERO DE MUESTRAS   !
C       ! QUE DEBE DETERMINAR POR CADA BIT DE ENTRADA Y EL   !
C       ! NUMERO  TOTAL  DE  BITS QUE  LEERA. LOS DATOS DE   !
C       ! ENTRADA (VI)  LOS  TOMA  DEL ARCHIVO NODIS.DAT Y   !
C       ! LOS RESULTADOS (VO) LOS  DEPOSITA  EN EL ARCHIVO   !
C       ! BANDALIM.DAT                                       !
C       ------------------------------------------------------
C       ABRE ARCHIVOS DE DATOS PARA LECTURA Y ESCRITURA
        OPEN (50,NAME='NODIS.DAT',TYPE='OLD',RECORDSIZE=300)
        OPEN (70,NAME='BANDALIM.DAT',TYPE='NEW',RECORDSIZE=300)
C       PIDE NUMERO DE MUESTRAS QUE CALCULARA
C       POR CADA BIT DE ENTRADA
        TYPE 2
        ACCEPT 4,N
C       PIDE EL NUMERO DE BITS QUE PROCESARA
        TYPE 3
        ACCEPT 5,MU
        ITE=0
C       LEE DATO DE ENTRADA
      8 READ(50,106)IV
        ITE=ITE+1
        T=0.
        VV=VO
C       CALCULA LAS MUESTRAS DISTORSIONADAS
        DO 7 I=1,N
        T=T+.00004
        VO=(VV-IV)*(2.718282828**(-T*30000))+IV
C       ESCRIBE RESULTADOS EN EL ARCHIVO BANDALIM.DAT
      7 WRITE(70,10)IV,VO
        IF(ITE.EQ.MU)GO TO 14
        GO TO 8
C       CIERRA ARCHIVOS DE DATOS
     14 CLOSE(70)
        CLOSE(50)
        STOP
C       SECCION DE FORMATOS DE LECTURA Y ESCRITURA
      2 FORMAT(28H NUMERO DE MUESTRAS POR BIT=, $)
      3 FORMAT(' NUMERO DE MUESTRAS QUE LEERA=',$)
      4 FORMAT(I3)
      5 FORMAT(I4)
     10 FORMAT(I3,F8.4)
    106 FORMAT(I3)
        END
```

PROGRAMA CORRESPONDIENTE AL FILTRO PASO-BAJAS DIGITAL

Este programa está basado en el principio del filtro transversal para hacer la convolución de la señal de entrada con un vector de muestras de la respuesta a impulso del filtro.

Los valores de la respuesta a impulso utilizados fueron calculados suponiendo un filtro paso-bajas ideal, tal como se muestra en la figura A.1. Este filtro tiene como respuesta a impulso una señal de la forma

$$C_n = \tau' Am \frac{\operatorname{sen}(\omega_n \tau'/2)}{\omega_n \tau'/2}$$

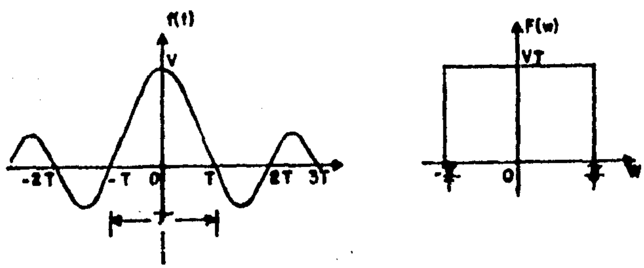


Figura A.1 Filtro paso-bajas ideal.

```
C      -----------------------------------------------------
C      ! ESTE PROGRAMA  SIMULA EL COMPORTAMIENTO DE UN FILTRO !
C      ! PASO-BAJAS DIGITAL;  HACE LA CONVOLUCION DE LA SEÑAL !
C      ! DE ENTRADA  CON  UN  VECTOR  C  QUE CORRESPONDE A LA !
C      ! RESPUESTA A IMPULSO DEL FILTRO.                      !
C      ! LEE LAS MUESTRAS A PROCESAR DEL ARCHIVO BANDALIM.DAT !
C      ! Y LOS VALORES DE LA RESPUESTA A IMPULSO  DESEADA DEL !
C      ! ARCHIVO  TRANSFUR.DAT.  LAS MUESTRAS  DE LA SEÑAL DE !
C      ! SALIDA DE ESTE FILTRO QUEDAN EN EL ARCHIVO SENAL.DAT !
C      -----------------------------------------------------
       DIMENSION X(20),C(20)
       OPEN(77,NAME='SENAL.DAT',TYPE='NEW',RECORDSIZE=300)
       OPEN(40,NAME='TRANSFUR.DAT',TYPE='OLD',RECORDSIZE=300)
       OPEN(70,NAME='BANDALIM.DAT',TYPE='OLD',RECORDSIZE=300)
       TYPE 320
       ACCEPT 322,NMI
C      LEE COEFICIENTES DEL ARCHIVO TRANSFUR.DAT
       DO 1 I=1,17
       X(I)=0
     1 READ(40,100)C(I)
       CLOSE(40)
       ITER=0
     5 READ(70,10)IDEAL,ENTRADA
       ITER=ITER+1
       DO 20 J=17,2,-1
    20 X(J)=X(J-1)
       X(1)=ENTRADA
       DISTOR=0.0
       DO 30 N=1,17
    30 DISTOR=DISTOR+C(N)*X(N)
       WRITE(77,25)IDEAL,DISTOR
       IF(ITER.EQ.NMI)GO TO 40
       GO TO 5
    40 CLOSE(70)
       CLOSE(77)
       STOP
    10 FORMAT(I3,F8.4)
    25 FORMAT(I3,F8.4)
   100 FORMAT(F8.4)
   320 FORMAT(' NUMERO DE MUESTRAS=',$)
   322 FORMAT(I4)
       END
```

```
>
*        ----------------------------------------
*        ! PROGRAMA CORRESPONDIENTE AL  ALGORITMO !
*        ! GRADIENTE  EN  EL LENGUAJE ENSAMBLADOR !
*        ! DEL  PROCESADOR   DIGITAL  DE   SEÑALES !
*      . ! TMS32010.  UTILIZA  UN  CODIGO  Q12  EN !
*        ! EL CALCULO DE TODAS SUS VARIABLES.      !
*        ! Δ=0.02 Y CONSIDERA 100 MUESTRAS PARA !
*        ! EL CALCULO DEL ERROR PROMEDIO, AL CUAL !
*        ! LE DA UNA  TOLERANCIA  DE HASTA 0.0043 !
*        ! SIN CAMBIAR COEFICIENTES.              !
*        ----------------------------------------
*
0001   0000   F900          B    INICIO
       0001   000A
0002   0002   0090   RATE   DATA   +144
0003   0003   00E1   MODO   DATA   >E1
0004   0004   1000   MAX    DATA   >1000
0005   0005   F000   MIN    DATA   >F000
0006   0006   000F   DELTA  DATA   >000F
0007   0007   F800   MEDIO  DATA   >F800
0008   0008   0064   NMUES  DATA   100
0009   0009   0708   LIM    DATA   1800
*      INICIALIZACION
*      APUNTA A LA PAGINA CERO DE LA MEMORIA DE
*      DATOS (MD) Y ALMACENA UN "1" Y UN "0" EN
*      LAS LOCALIDADES 29 Y 30 RESPECTIVAMENTE.
0010   000A   6E00   INICIO LDPK   0
0011   000B   7E01          LACK   1
0012   000C   501D          SACL   29
0013   000D   7F89          ZAC
0014   000E   501E          SACL   30
*      INICIALIZA TARJETA ANALOGICA
0015   000F   7E02          LACK   RATE
0016   0010   6700          TBLR   0
0017   0011   4900          OUT    0,1
0018   0012   7E03          LACK   MODO
0019   0013   6700          TBLR   0
0020   0014   4800          OUT    0,0
*      ALMACENA CONSTANTES CORRESPONDIENTES
*      A LOS NIVELES LOGICOS +1 Y -1 EN Q12
0021   0015   7E04          LACK   MAX
0022   0016   6719          TBLR   25
0023   0017   7E05          LACK   MIN
0024   0018   671A          TBLR   26
*      LLEVA A MEMORIA DE DATOS LOS VALORES
*      DE Δ, NUMERO DE MUESTRAS Y  LIMITE
0025   0019   7E06          LACK   DELTA
0026   001A   671B          TBLR   27
0027   001B   7E07          LACK   MEDIO
0028   001C   671C          TBLR   28
```

```
0029   001D   7E08           LACK   NMUES
0030   001E   6725           TBLR   37
0031   001F   6723           TBLR   35
0032   0020   7E09           LACK   LIM
0033   0021   6726           TBLR   38
*     INICIALIZA VECTORES X Y C EN CERO
0034   0022   7F89           ZAC
0035   0023   700A           LARK   0,18
0036   0024   6880           LARP   0
0037   0025   5088   CEROS   SACL   *
0038   0026   F400           BANZ   CEROS
       0027   0025
0039   0028   5024           SACL   36
0040   0029   2019           LAC    25
0041   002A   5028           SACL   40
0042   002B   500A           SACL   10
*     ACCESA MUESTRA EN Q15 Y LA RECORRE A Q12
0043   002C   4200   WAIT    IN     0,2
0044   002D   2D00           LAC    0,13
0045   002E   5800           SACH   0
*     EJECUTA CONVOLUCION Y RECORRE EL VECTOR X
0046   002F   7008           LARK   0,8
0047   0030   7112           LARK   1,18
0048   0031   7F89           ZAC
0049   0032   6880           LARP   0
0050   0033   6D1E           MPY    30
0051   0034   6B81   LOOP    LTD    *,1
0052   0035   6D90           MPY    *-,0
0053   0036   F400           BANZ   LOOP
       0037   0034
0054   0038   7F8F           APAC
0055   0039   5C14           SACH   20,4
*     DETERMINA SIGNO DE LA ESTIMACION Y
*     LO DEPOSITA EN LA LOCALIDAD 21
0056   003A   2014           LAC    20
0057   003B   FD00           BGEZ   GAMA
       003C   0040
0058   003D   201A           LAC    26
0059   003E   F900           B      TETA
       003F   0041
0060   0040   2019   GAMA    LAC    25
0061   0041   5015   TETA    SACL   21
*     CALCULA ERROR Y LO ALMACENA EN LA
*     LOCALIDAD 22
0062   0042   1014           SUB    20
0063   0043   5016           SACL   22
*     SACA VALOR ABSOLUTO AL ERROR, LO
*     LLEVA A Q5 Y LO ACUMULA
0064   0044   2916           LAC    22,9
0065   0045   7F88           ABS
0066   0046   6024           ADDH   36
0067   0047   5C24           SACH   36
```

```
*       LLEVA SIGNO DE ESTIMACION A Q14
*       Y LO SACA POR EL PUERTO 2
0068    0048    2215                    LAC     21,2
0069    0049    5015                    SACL    21
0070    004A    4A15                    OUT     21,2
*       DECREMENTA CONTADOR DE MUESTRAS
0071    004B    2023                    LAC     35
0072    004C    101D                    SUB     29
0073    004D    5023                    SACL    35
0074    004E    FC00                    BGZ     OMEGA
        004F    0057
*       REINICIALIZA CONTADOR DE MUESTRAS
0075    0050    2025                    LAC     37
0076    0051    5023                    SACL    35
*       COMPARA EL ERROR ACUMULADO CON
*       EL  LIMITE  MAXIMO  PERMITIDO
0077    0052    2024                    LAC     36
0078    0053    1026                    SUB     38
0079    0054    5028                    SACL    40
*       REINICIALIZA ERROR ACUMULADO
0080    0055    201E                    LAC     30
0081    0056    5024                    SACL    36
*       SI EL ERROR ACUMULADO ES MENOR QUE EL
*       LIMITE  PERMITIDO  SALTA A LEER  OTRA
*       MUESTRA  SIN ACTUALIZAR  COEFICIENTES
0082    0057    2028    OMEGA           LAC     40
0083    0058    FB00                    BLEZ    WAIT
        0059    002C
*       ACTUALIZA COEFICIENTES
0084    005A    7009                    LARK    0,9
0085    005B    7112                    LARK    1,18
0086    005C    6A16                    LT      22
0087    005D    6D1B                    MPY     27
0088    005E    7F8E                    PAC
0089    005F    5C17                    SACH    23,4
0090    0060    6A17                    LT      23
0091    0061    6880                    LARP    0
0092    0062    6D81    ALFA            MPY     *,1
0093    0063    7F8E                    PAC
0094    0064    5C2D                    SACH    45,4
0095    0065    202D                    LAC     45
0096    0066    0089                    ADD     *
0097    0067    5090                    SACL    *-,0,0
0098    0068    F400                    BANZ    ALFA
        0069    0062
*       SALTA A LEER OTRA MUESTRA
0099    006A    F900                    B       WAIT
        006B    002C
0100                                    END
```

APENDICE B

PROCESADOR DIGITAL DE SEÑALES TMS32010

B.1 Porqué se eligió el TMS32010

Las operaciones para el algoritmo gradiente pueden ser clasificadas genericamente como de entrada, lectura y escritura de memoria, adición signada y multiplicación signada. Las primeras cinco operaciones pueden ser ejecutadas por cualquier microprocesador cuyo bus permite un modo de operación síncrono. La multiplicación signada, sin embargo, presenta un gran problema. La mayoria de los microprocesadores de propósito general disponibles comercialmente, no tienen una instrucción de multiplicación signada, con excepción de algunos de los mas avanzados tales como el Motorola 68000 y el Intel 8086. Desafortunadamente la multiplicación signada en estos microprocesadores es ejecutada como un microprograma que no tiene un número fijo de ciclos de reloj para su ejecución. Esto significa que el procesador tendría que ser resincronizado después de cada operación de multiplicación para el caso de el igualador T-espaciado.

El problema puede ser resuelto mediante un microprocesador introducido por la compañía Texas Instruments cuyo conjunto de instrucciones incluye la multiplicación signada en un solo ciclo de reloj. Tal microprocesador es el TMS32010 que tiene además, instrucciones especiales para el procesado de señales a alta velocidad.

B.2 Caracteristicas del TMS32010

Ultimamente ha sido desarrollado un nuevo tipo de productos para varías computadoras, estos productos son llamados Sistemas de Manejo Telefónico (TMS) y su propósito es eslabonar la inteligencia y capacidad de memoria de una microcomputadora con la capacidad de comunicación del teléfono. En adición a manipular voz y comunicación de datos, un TMS puede ser programado para hacer tereas especificas en tiempos predeterminados asi como recibir mensajes telefónicos, o proveer palabras de paso de seguridad (password), etc.

El TMS32010 es el primer miembro de una nueva familia de procesadores de señales digitale (PSD) diseñados para aplicaciones de alta velocidad. La familia TMS320 contiene la primera microcomputadora MOS capás de ejecutar cinco millones de instrucciones por segundo.

El TMS32010 utiliza una modificación de la arquitectura Harvard en la cual la memoria de datos y la memoria de programa se encuentran separadas, permitiendo así un translape total de los ciclos de búsqueda y ejecución de cada instrucción. Además, en el TMS32010 se permite la transferencia de datos entre las memorias de programa y de datos.

El TMS32010 combina los siguientes elementos en un solo circuito integrado (ver figura B.1)

- Acumulador (ACC).- Acumulador de 32 bits.

- Unidad Aritmetica Logica (ALU).- ALU con dos puertos de 32 bits.

- Registros auxiliares (AR0 y AR1).- Dos registros auxiliares para direccionamiento indirecto de la memoria de datos y control del contador de ciclos. Los nueve bits menos significativos de cada registro son configurados como contadores bidireccionales.

- Apuntador de registro auxiliar (ARP).- Registro de un solo bit que contiene la dirección del registro auxiliar en función.

- Bus de datos (DBUS).- Bus de 16 bits conectado a la RAM de datos.

- Apuntador de página de memoria de datos (DP).- Registro de un solo bit que contiene la dirección de la página de RAM de datos (una página=128 palabras).

- RAM de datos.- 144 palabras de 16 bits de RAM para contener datos.

- Registro de bandera de interrupción (INTF).- Registro de un solo bit que indica cuando hay una solicitud de interrupción.

- Multiplicador.- Multiplicador hardware paralelo de 16X16 bits.

- Registro de bandera de sobreflujo (OV).- Registro de un solo bit que indica cuando ha ocurrido un sobreflujo en una operación aritmética.

- Registro del modo sobreflujo (OVM).- Registro de un solo bit que define el modo saturado o no saturado en operaciones aritméticas.

- Registro P (P).- Registro de 32 bits que contiene el producto de las multiplicaciones.

- Bus de programa (PBus).- Bus de 16 bits conectado a la memoria del programa.

- Contador del programa (PC).- Registro de 12 bits que contiene la dirección de memoria del programa.

- ROM del programa.- 1536 palabras de 16 bits que contienen el código del programa (solo en la versión TMS320M10).

- Registros de corriemiento.- Uno es un registro de corrimiento a la izquierda variable de 0-15 bits al mover datos de memoria a la ALU. El otro registro de corrimiento actúa en el acumulador cuando se almacenan datos a la RAM de datos; este registro puede tener corrimientos a la izquierda de 0,1 o 4 bits.

- Stack.- Registro de 4 por 12 bits para salvar el contenido del contador del programa en sub-rutinas o llamadas de interrupción.

- Registro T (T).- Registro de 16 bits que contiene el multiplicando durante las operaciones de multiplicación.

Para el caso del TMS32010, que no contiene memoria de programa interna, es posible direccionar hasta 4096 palabras de memoria de programa externa. Como el ciclo de máquina del TMS32010 es de 200 ns, la memoria de programa externa debe tener un tiempo de acceso máximo de 100 ns.

El grupo de instrucciones del TMS32010 es el que se muestra a continuación

INSTRUCCIONES DEL ACUMULADOR				
MNEMONICO	DESCRIPCION	No. CICLOS	No. PALABRAS	CODIGO DE OPERACION 15 … 0
ABS	VALOR ABSOLUTO AL ACUMULADOR	1	1	0 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0
ADD	SUMA ACUMULADOR CON CORRIMIENTO	1	1	0 0 0 0!-- S --!------ D ------!
ADDH	SUMA PARTE ALTA DEL ACUMULADOR	1	1	0 1 1 0 0 0 0 0!------ D ------!
ADDS	SUMA ACUMULADOR SIN EXTENSION DE SIGNO	1	1	0 1 1 0 0 0 0 1!------ D ------!
AND	AND CON ACUMULADOR	1	1	0 1 1 1 1 0 0 1!------ D ------!
LAC	CARGA ACUMULADOR CON CORRIMIENTO	1	1	0 0 1 0!-- S --!------ D ------!
LAC	CARGA ACUMULADOR CON DIRECCIONAMIENTO INMEDIATO	1	1	0 1 1 1 1 1 1 0!------ K ------!
OR	OR CON ACUMULADOR	1	1	0 1 1 1 1 0 1 0!------ D ------!
SACH	ALMACENA 16 BITS SUPERIORES DEL ACUMULADOR CON CORRIMIENTO	1	1	0 1 0 1 1!- X -!------ D ------!
SACL	ALMACENA 16 BITS INFERIORES DEL ACUMULADOR	1	1	0 1 0 1 0 0 0 0!------ D ------!
SUB	RESTA DEL ACUMULADOR CON CORRIMIENTO	1	1	0 0 0 1!-- S --!------ D ------!
SUBC	RESTA CONDICIONALMENTE	1	1	0 1 1 0 0 1 0 0!------ D ------!
SUBH	RESTA DE LOS 16 BITS SUPERIORES DEL ACUMULADOR	1	1	0 1 1 0 0 0 1 0!------ D ------!
SUBS	RESTA DEL ACUMULADOR SIN EXTENSION DE SIGNO	1	1	0 1 1 0 0 0 1 1!------ D ------!
XOR	OR EXCLUSIVA CON EL ACUMULADOR	1	1	0 1 1 1 1 0 0 0!------ D ------!
ZAC	CERO AL ACUMULADOR	1	1	0 1 1 1 1 1 1 1 1 0 0 0 1 0 0 1
ZALH	CERO AL ACUMULADOR Y CARGA LOS 16 BITS SUPERIORES	1	1	0 1 1 0 0 1 0 1!------ D ------!
ZALS	CERO AL ACUMULADOR Y CARGA LOS 16 BITS INFERIORES SIN EXTENSION DE SIGNO	1	1	0 1 1 0 0 1 1 0!------ D ------!

INSTRUCCIONES DE REGISTROS AUXILIARES Y APUNTADOR DE PAGINA DE DATOS

MNEMONICO	DESCRIPCION	No. CICLOS	No. PALABRAS	CODIGO DE OPERACION 15 ... 0
LAR	CARGA REGISTRO AUXILIAR	1	1	0 0 1 1 1 0 0 R!------ D ------!
LARK	CARGA REGISTRO AUXILIAR CON DIRECCIONAMIENTO INMEDIATO	1	1	0 1 1 1 0 0 0 R!------ K ------!
LARP	CARGA APUNTADOR DE REGISTRO AUXILIAR CON DIRECCIONAMIENTO INMEDIATO	1	1	0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 K
LDP	CARGA APUNTADOR DE PAGINA DE MEMORIA DE DATOS	1	1	0 1 1 0 1 1 1 1!------ D ------!
LDPK	CARGA APUNTADOR DE PAGINA DE MEMORIA DE DATOS CON DIRECCIONAMIENTO INMEDIATO	1	1	0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 K
MAR	MODIFICA REGISTRO AUXILIAR	1	1	0 1 1 0 1 0 0 0!------ D ------!
SAR	ALMACENA REGISTRO AUXILIAR	1	1	0 0 1 1 0 0 0 R!------ D ------!

OPERACIONES DE ENTRADA/SALIDA Y MEMORIA DE DATOS

MNEMONICO	DESCRIPCION	No. CICLOS	No. PALABRAS	CODIGO DE OPERACION 15 ... 0
DMOV	COPIA CONTENIDO DE UNA LOCALIDAD DE MEMORIA EN LA SIGUIENTE LOCALIDAD	1	1	0 1 1 0 1 0 0 1!------ D ------!
IN	ENTRADA DE DATO DESDE UN PUERTO	2	1	0 1 0 0 0!-PA -!------ D ------!
OUT	SALIDA DE DATO POR UN PUERTO	2	1	0 1 0 0 1!-PA -!------ D ------!
TBLR	LEE DATO DE MEMORIA DE PROGRAMA A MEMORIA DE DATOS	3	1	0 1 1 0 0 1 1 1!------ D ------!
TBLW	ESCRIBE DE LA MEMORIA DE DATOS A LA MEMORIA DE PROGRAMA	3	1	0 1 1 1 1 1 0 1!------ D ------!

INSTRUCCIONES DE CONTROL

MNEMONICO	DESCRIPCION	No. CICLOS	No. PALABRAS	CODIGO DE OPERACION 15 … 0
DINT	DESHABILITA INTERRUPCIONES	1	1	0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1
EINT	HABILITA INTERRUPCIONES	1	1	0 1 1 1 1 1 1 1 1 0 0 0 0 0 1 0
LST	CARGA REGISTRO DE ESTADOS	1	1	0 1 1 1 1 0 1 1 !------ D ------!
NOP	NO OPERACION	1	1	0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
POP	POP STACK AL ACUMULADOR	2	1	0 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1
PUSH	PUSH STACK DEL ACUMULADOR	2	1	0 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0
ROVM	QUITA MODO SOBREFLUJO	1	1	0 1 1 1 1 1 1 1 1 0 0 0 1 0 1 0
SOVM	PONE MODO SOBREFLUJO	1	1	0 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1
SST	ALMACENA REGISTRO DE ESTADOS	1	1	0 1 1 1 1 1 0 0 !------ D ------!

INSTRUCCIONES DE REGISTROS P, T Y MULTIPLICACION

MNEMONICO	DESCRIPCION	No. CICLOS	No. PALABRAS	CODIGO DE OPERACION 15 … 0
APAC	SUMA P AL ACUMULADOR	1	1	0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1
LT	CARGA REGISTRO T	1	1	0 1 1 0 1 0 1 0 !------ D ------!
LTA	COMBINA LAS INSTRUCCIONES LT Y APAC	1	1	0 1 1 0 1 1 0 0 !------ D ------!
LTD	COMBINA LAS INSTRUCCIONES LT, APAC Y DMOV	1	1	0 1 1 0 1 0 1 1 !------ D ------!
MPY	MULTIPLICA CON REGISTRO T Y PRODUCTO QUEDA EN P	1	1	0 1 1 0 1 1 0 1 !------ D ------!
MPYK	MULTIPLICA REGISTRO T CON OPERANDO INMEDIATO	1	1	1 0 0 !--------- K ------------!
PAC	CARGA REGISTRO P EN EL ACUMULADOR	1	1	0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0
SPAC	RESTA P DEL ACUMULADOR	1	1	0 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0

<table>
<tr><th colspan="5">INSTRUCCIONES DE SALTO</th></tr>
<tr><th>MNEMONICO</th><th>DESCRIPCION</th><th>No. CICLOS</th><th>No. PALABRAS</th><th>CODIGO DE OPERACION
15 ... 0</th></tr>
<tr><td>B</td><td>SALTO INCONDICIONAL</td><td>2</td><td>2</td><td>1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BANZ</td><td>SALTA SI REGISTRO AUXI-
LIAR NO ES CERO</td><td>2</td><td>2</td><td>1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BGEZ</td><td>SALTA SI ACUMULADOR ES
MAYOR O IGUAL A CERO</td><td>2</td><td>2</td><td>1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BGZ</td><td>SALTA SI ACUMULADOR ES
MAYOR QUE CERO</td><td>2</td><td>2</td><td>1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BIOZ</td><td>SALTA SI LA PIN BIO ES
IGUAL A CERO</td><td>2</td><td>2</td><td>1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BLEZ</td><td>SALTA SI ACUMULADOR ES
MENOR O IGUAL A CERO</td><td>2</td><td>2</td><td>1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BLZ</td><td>SALTA EL ACUMULADOR ES
MENOR QUE CERO</td><td>2</td><td>2</td><td>1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BNZ</td><td>SALTA SI ACUMULADOR ES
DIFERENTE DE CERO</td><td>2</td><td>2</td><td>1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BV</td><td>SALTA SI HAY SOBREFLUJO</td><td>2</td><td>2</td><td>1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>BZ</td><td>SALTA SI EL ACUMULADOR
ES IGUAL A CERO</td><td>2</td><td>2</td><td>1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>CALA</td><td>LLAMA SUBRUTINA DESDE
EL ACUMULADOR</td><td>2</td><td>1</td><td>0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0</td></tr>
<tr><td>CALL</td><td>LLAMA SUBRUTINA CON DI-
RECCIONAMIENTO INMEDIATO</td><td>2</td><td>2</td><td>1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 |-DIRECCION DEL SALTO-|</td></tr>
<tr><td>RET</td><td>REGRESA DE SUBRUTINA O
DE RUTINA DE INTERRUPCION</td><td>2</td><td>1</td><td>0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 1</td></tr>
</table>

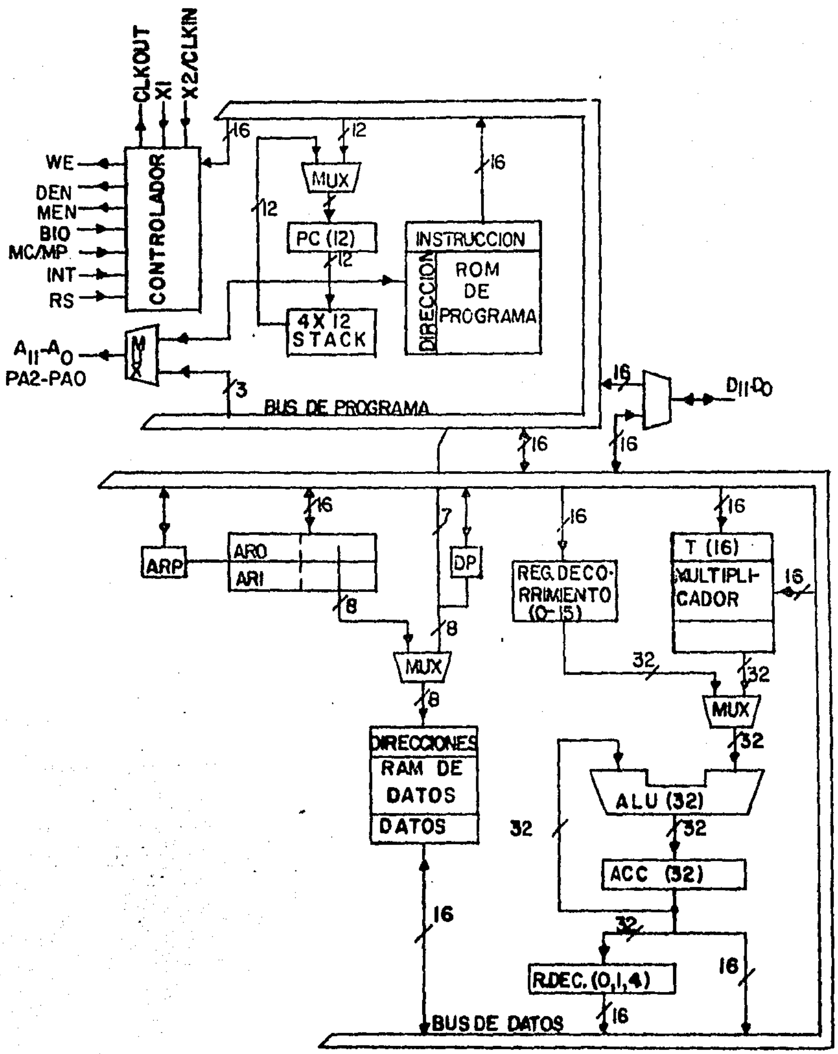


Figura B.1 Configuración interna del TMS32010

APENDICE C

ARITMETICA DE PUNTO FIJO

CON FORMATOS Q

Los cálculos en el TMS32010 están basados en la representación de los números en complemento a dos con punto fijo. Cada numero de 16 bits es evaluado con un bit de signo, i bits enteros y 15-i bits fraccionales. Entonces el número

00000010 10100000

punto decimal

tiene un valor de 2.625. Este numero particular se dice que está representado en un formato Q_8 (8 bits fraccionales). Su rango está entre -128 (1000000000000000) y 127.996 (0111111111111111). La precisión fraccional de un número en Q_8 es de 0.004 (2^{-8}).

En algunas ocaciones es necesario manejar simultáneamente números con muy diversas magnitudes (algunos solo enteros y otros únicamente fraccionales), entonces se debe tener cuidado al sumarlos cuando no están representados en el mismo formato Q. Además, se debe tomar en cuenta que el registro P del TMS32010 (donde queda el producto de las multiplicaciones) es de 32 bits y la memoria de datos es únicamente de 16 bits, así que, para guardar un producto en memoria se debe considerar el formato en que se desea guardar éste para dar el corrimiento de bits necesarios para llevar el producto a tal formato.

A continuación discutiremos algunas técnicas involucradas en el uso de este tipo de representación numérica.

a) Multiplicación.- Existe una gran variedad de situaciones que podemos encontrar cuando multiplicamos dos números.

Primer caso.- Fracción por fracción

Ejemplo 1: $Q_{15} * Q_{15}$

```
               0 100000000000000 = 0.5 en notación Q15
             * 0 100000000000000 = 0.5 en notación Q15
---------------------------------
00 010000000000000000000000000000 = 0.25 en Q30
 ↑
 └──── punto decimal
```

Para que el resultado se pueda almacenar en memoria en formato Q_{15}, es necesario recorrer 1 bit a la izquierda el producto obtenido y llevar a memoria solo los 16 bits mas significativos (0 010000000000000=0.25 en Q_{15}).

Ejemplo 2: $Q_{15} * Q_{12}$

```
              0 100000000000000 = 0.5 en Q15
          * 0000 100000000000 = 0.5 en Q12
----------------------------------
00000 01000000000000000000000000000 = 0.25 en Q27
   ↑
   └──── punto decimal
```

En este ejemplo, si se desea que el resultado quede en Q_{15} es necesario correr el producto 4 bits a la izquierda y almacenar en memoria los 16 bits mas significativos.

Segundo Caso.- Entero por entero

Ejemplo: $Q_0 * Q_0$

0000000000010001 = 17 en Q_0

* 1111111111111011 = -5 en Q_0

11111111111111111111111110101011 = -85 en Q_0

punto decimal

Para este caso el resultado deseado está completamente en la mitad inferior del producto, entonces, no es necesario ningún corrimiento al almacenar en memoria los 16 bits menos significativos del resultado. Debe notarse también de éste ejemplo que toda la mitad superior del producto son bits redundantes del bit de signo.

Tercer caso.- Enteros y fracciones

Ejemplo: $Q_{14} * Q_{14}$

01 10000000000000 = 1.5 en Q_{14}

* 00 11000000000000 = 0.75 en Q_{14}

0001 0010000000000000000000000000 = 1.125 en Q_{28}

punto decimal

Debe notarse en este caso que la magnitud máxima de un número en Q_{14} es muy próxima pero menor a 2, entonces la magnitud máxima del producto de dos números en Q_{14} es menor a 4. De lo anterior se puede obtener la siguiente regla

El producto de un número con i bits enteros y f bits fraccionales y un segundo número con j bits enteros y g bits fraccionales será un número con (i+j) bits enteros y (f+g) bits fraccionales. La mayor precisión posible para una representación de 16 bits de éste número tendrá (i+j) bits enteros y (15-i-j) bits fraccionales.

b) Adición.- Como ya se vió, durante el proceso de multiplicación sólo es necesario ajustar el punto decimal después de la operación. Para la adición,en cambio, se deben considerar dos cosas, una es que ambos operandos tienen que estar representados en la misma notación Q; y la segunda consideración es que se debe manejar el formato Q conveniente para evitar sobreflujos en el resultado, o bien, manipular en forma adecuada tales sobreflujos.