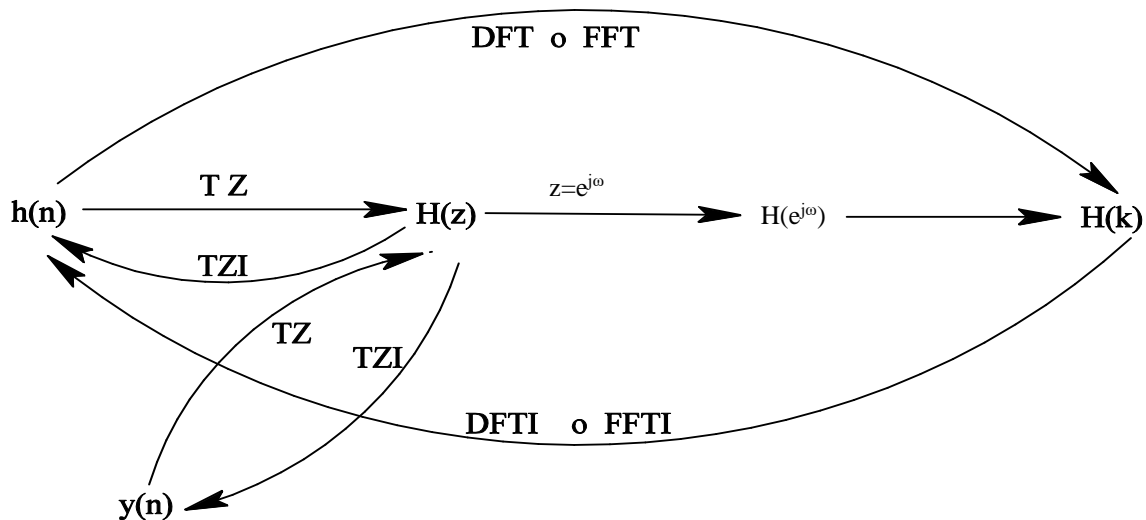


Conceptos Básicos de Procesamiento Digital de Señales



LARRY HIPÓLITO ESCOBAR SALGUERO

FACULTAD DE INGENIERÍA UNAM

SEPTIEMBRE DE 2008

Conceptos Básicos de Procesamiento Digital de Señales

LARRY HIPÓLITO ESCOBAR SALGUERO

FACULTAD DE INGENIERÍA UNAM

SEPTIEMBRE DE 2008

ÍNDICE

TEMA	PÁGINA
INTRODUCCIÓN	1
Elementos básicos de un sistema de procesamiento digital de señales	2
Diferencias entre el procesamiento digital y analógico	3
CAPÍTULO 1. SEÑALES	6
1.1 Definición de señales	6
1.2 Clasificación general	7
1.3 Clasificación por su descripción matemática	7
1.4 Señales continuas y determinísticas de interés.....	8
1.5 Propiedades de la función impulso	11
1.6 Resumen	11
1.7 Problemas propuestos.....	12
CAPÍTULO 2. CONVERSIÓN ANÁLOGO DIGITAL	13
2.1 Muestreo de una señal en el tiempo continuo.....	13
2.2 Teorema de Nyquist o Shannon o teorema del muestreo	15
2.3 Espectro de una señal muestreada.....	18
2.4 Cuantización.....	19
2.5 Modulación por codificación de pulsos (PCM)	20
2.6 Relación señal a ruido de cuantización	22
2.7 Compresión logarítmica	22
2.8 Recuperación de una señal analógica a partir de una señal muestreada.....	24
2.9 Resumen	25
2.10 Problemas propuestos.....	25
CAPÍTULO 3. SEÑALES DISCRETAS	26
3.1 Definición de señales discretas	26
3.2 Formas de representación de una señal o función discreta.....	27
3.3 Funciones discretas especiales o secuencias básicas.....	27
3.4 Operaciones básicas con secuencias	35
3.5 Resumen.....	38
3.6 Problemas propuestos.....	38
CAPÍTULO 4. SISTEMAS DISCRETOS	39
4.1 Sistemas.....	39
4.2 Caracterización de un sistema	40
4.3 Bloques básicos de sistemas discretos.....	41
4.4 Clasificaciones de los Sistemas Discretos.....	46
4.5 Sistemas lineales e invariantes en el tiempo discreto (SLITD).....	48
4.6 Convolución	50
4.7 Interconexión de sistemas	54

4.8	Correlación.....	56
4.9	Resumen.....	61
4.10	Problemas propuestos.....	61

CAPÍTULO 5. TRANSFORMADA Z.....62

5.1	Respuesta de un SLITD a una exponencial compleja.....	62
5.2	Definición de transformada Z.....	64
5.3	Deducción de la TZ por transformada de Laplace.....	64
5.4	Región de la convergencia de la TZ (ROC).....	66
5.5	Estabilidad de un sistema discreto.....	67
5.6	Secuencia geométrica.....	71
5.7	Propiedades de la TZ.....	72
5.8	Ejemplos de TZ de funciones elementales.....	75
5.9	Diseño de osciladores discretos.....	80
5.10	Resumen.....	81
5.11	Problemas propuestos.....	81

CAPÍTULO 6. TRANSFORMADA Z INVERSA (TZI)82

6.1	Métodos para obtener la TZI.....	82
6.2	Polos de orden superior.....	90
6.3	Polos conjugados complejos.....	90
6.4	Resumen.....	93
6.5	Problemas propuestos.....	93

CAPÍTULO 7. TRANSFORMADA DISCRETA DE FOURIER.....94

7.1	Espectro de señales y sistema discretos.....	94
7.2	La transformada discreta de fourier.....	96
7.3	Interpretación gráfica de la DFT.....	98
7.4	Desarrollo analítico en el tiempo.....	101
7.5	Transformada discreta de Fourier inversa (DFTI).....	110
7.6	Desarrollo matricial de la DFT.....	111
7.7	Propiedades de la DFT.....	111
7.8	Algoritmo de Goertzel.....	119
7.9	Resumen.....	122
7.10	Problemas propuestos.....	122

CAPÍTULO 8. TRANSFORMADA RÁPIDA DE FOURIER (FFT)..... 123

8.1	Desarrollo de la FFT.....	124
8.2	Desarrollo de la FFT radix 2 a partir de la definición de la DFT.....	128
8.3	Decimación.....	135
8.4	Desarrollo alternativo de la FFT.....	139
8.5	Algoritmo FFT de decimación en frecuencia radix 2.....	142
8.6	La FFT inversa.....	144
8.7	Resumen.....	157
8.8	Problemas propuestos.....	157

CAPÍTULO 9. LA TRANSFORMADA COSENO 158

9.1	Transformadas.....	159
9.2	Clasificación de las transformadas.....	160

9.3 Algoritmos optimizados para la DCT y DCTI.....	168
9.4 Ejemplo de aplicación.....	171
9.5 Resumen.....	175
9.6 Problemas propuestos.....	175
CAPÍTULO 10. FORMATOS NUMÉRICOS BINARIOS	176
10.1 Formato numérico de punto fijo.....	177
10.2 Truncamiento y redondeo.....	180
10.3 Operaciones binarias	181
10.4 Formato numérico de punto flotante	183
10.5 Estándar IEEE 754	185
10.6 Formato de Microsoft.....	186
10.7 Texas Instruments (TI).....	186
10.8 Otros formatos.....	189
10.9 Resumen	190
10.10 Problemas propuestos.....	190
BIBLIOGRAFÍA	191
GLOSARIO.....	194

CAPITULO 1

SEÑALES

En este capítulo se realiza una breve introducción al concepto de señales y funciones continuas, se inicia con la concepción física de la misma, se presentan algunas formas de clasificación de las señales y algunas funciones importantes a tratar en este texto para definir las señales discretas. En la mayoría de capítulos de este texto se involucran conceptos asociados con la teoría de Fourier, por tanto es necesario que el lector tenga presente éstas bases.

1.1 DEFINICIÓN DE SEÑALES

Una señal se puede considerar como una variable o cantidad física que provee información sobre el estado o evolución de un sistema o fenómeno [KUU05]. La variable física puede ser función del tiempo, el espacio o cualquier otra variable o variables, es decir, que una señal puede considerarse como un fenómeno físico (cantidad física) que experimenta cambios en el tiempo, el espacio u otra variable independiente. Una señal es una descripción de cómo un parámetro varía con otro parámetro.

El ser humano siempre ha tenido la tendencia a comprender y representar los fenómenos de su medio a través de abstracciones matemáticas que faciliten la comprensión y el estudio de tales fenómenos. Para su análisis formal una señal se representa como una función matemática de una o varias variables, y debido a que una de las variables de mayor importancia en la existencia humana es el tiempo, la caracterización de las señales y los sistemas están dadas en función del tiempo, es decir, que una función del tiempo representa un modelo de la señal. Algunas veces se utilizan indistintamente los términos de señal y función.

Las señales del mundo real por naturaleza son analógicas y se pueden representar en forma de funciones continuas, cuyo espectro de frecuencia está dado a través de su transformada de Fourier, que también es continua.

1.2 CLASIFICACIÓN GENERAL

Una clasificación muy general utilizada en el mundo de las señales indica que existen dos tipos: Continuas y Discretas [PRO92].

SEÑAL CONTINUA

Las señales continuas están definidas para cualquier valor del dominio de los números reales y toman valores en un intervalo continuo, matemáticamente se representan como funciones de variable continua. A este tipo de señal también se le suele llamar analógica y son las que se presentan en el mundo real.

SEÑAL DISCRETA

Está definida sólo para ciertos puntos del dominio de los reales y comúnmente se utiliza el dominio de los enteros, donde estos puntos no necesitan ser equidistantes. Este tipo de señales se presenta por secuencias de números, secuencias discretas o tablas de datos.

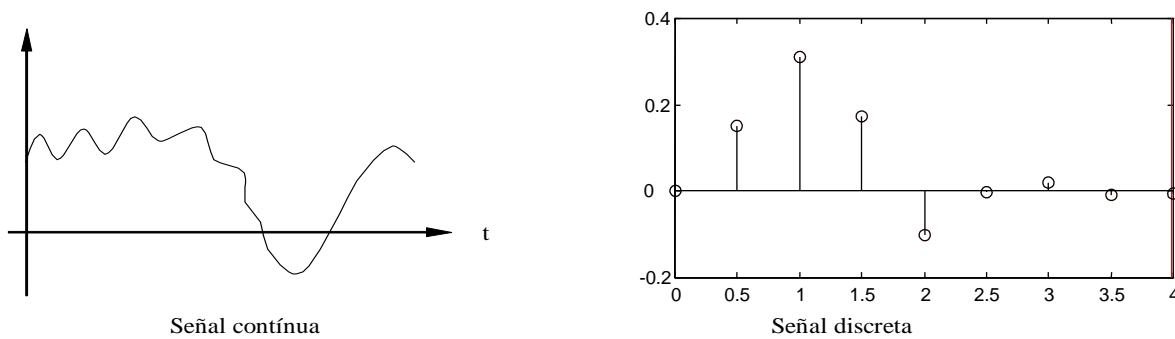


Figura 1.1 Formas de señal continua y discreta

1.3 CLASIFICACIÓN POR SU DESCRIPCIÓN MATEMÁTICA

Desde el punto de vista matemático las señales se pueden clasificar en Determinísticas y Aleatorias.

SEÑALES DETERMINÍSTICAS

Son señales que pueden ser descritas por una expresión matemática explícita, una tabla de datos, o una regla. Para estas señales los valores pasados, presentes y futuros de la señal pueden ser conocidos sin ninguna incertidumbre, es decir, que están completamente determinados.

SEÑALES ALEATORIAS

Se caracterizan porque sus valores en cada instante de tiempo no son predecibles. Las señales aleatorias son producidas por los fenómenos naturales tales como las ondas oceánicas, las ondas sísmicas, etc. Estas señales se pueden caracterizar por funciones de densidad de probabilidad, valores esperados, variancias y funciones de correlación. Para su análisis y descripción se utilizan técnicas estadísticas, teoría de probabilidad y procesos estocásticos en vez de fórmulas explícitas.

1.4 SEÑALES CONTÍNUAS Y DETERMINÍSTICAS DE INTERÉS

A continuación se enuncian algunas señales que son de mucha importancia en el PDS y se utilizarán continuamente en este texto. Dentro de estas funciones o señales tenemos las senoidales y cosenoidales que corresponden al espacio ortogonal más famoso que existe; por otro lado mencionamos funciones como el escalón y el pulso con el fin de llegar a obtener la función impulso que es de suma importancia en el proceso de convertir una señal continua a discreta y en la caracterización de un sistema.

SEÑAL PERIÓDICA

Se dice que una señal continua $f(t)$ es periódica si existe una constante $2p$ con la propiedad de que $f(t + 2p) = f(t)$ para toda t . Si $2p$ es el menor número para el que se cumple esta identidad, entonces $2p$ recibe el nombre de período de la función.

SEÑAL SINUSOIDAL O COSENOIDAL

Es una oscilación armónica simple descrita matemáticamente por $x(t) = A \cos(\omega t + \beta)$ y está completamente determinada por su amplitud A , su frecuencia ω y el ángulo de fase β .

Las funciones sinusoidales son de gran importancia, ya que de la teoría de Fourier, una función periódica $f(t)$ se puede expresar como una suma de funciones senoidales y cosenoidales dada por la serie de Fourier como se muestra en la figura 1.2 y expresada por la ecuación (1.1)

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{\pi n t}{p} + b_n \operatorname{sen} \frac{\pi n t}{p} \right) \quad (1.1)$$

donde

$$p = \frac{T}{2} \Rightarrow \frac{\pi n t}{p} = \frac{2\pi n t}{T} = 2\pi n t f = \omega_0 n t \quad \text{y} \quad f = \frac{1}{T}$$

$$b_n = \frac{2}{T} \int f(t) \sin \frac{2n\pi t}{T} dt \quad a_n = \frac{2}{T} \int f(t) \cos \frac{2n\pi t}{p} dt \quad a_0 = \frac{2}{T} \int f(t) dt$$

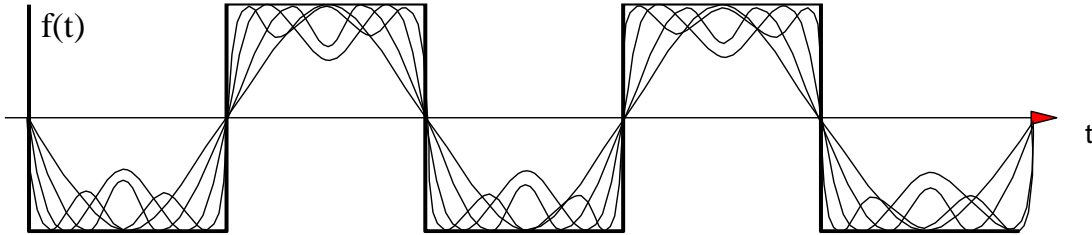


Figura 1.2 Descomposición de una Señal Periódica en Series de Fourier

SERIE DE FOURIER EN FORMA EXPONENCIAL

Si $f(t)$ es una función periódica con período T , entonces $f(t)$ se puede expresar como la serie

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t} \quad (1.2)$$

$$\omega_0 = \frac{2\pi}{T}$$

es decir, que ω_0 es la frecuencia fundamental de $f(t)$ y C_n son los coeficientes de la serie dados por

$$C_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jn\omega_0 t} dt \quad (1.3)$$

La transformada de Fourier en tiempo continuo de una función $x(t)$ se define

$$x(t) \leftrightarrow X(\omega)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega \quad X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \quad (1.4)$$

Función Escalón unitario

$$U(t) = \begin{cases} 1 & \text{si } t \geq 0 \\ 0 & \text{si } t < 0 \end{cases} \quad (1.5)$$

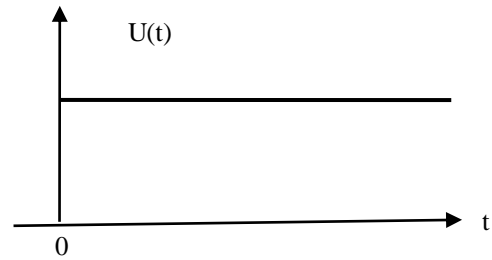


Figura 1.3 Función escalón unitario

Función Pulso

$$f(t) = \frac{1}{a}U(t) - \frac{1}{a}U(t-a) = P(t) \quad (1.6)$$

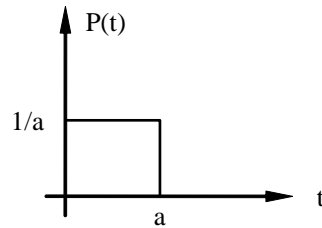


Figura 1.4 Función pulso

Función Delta de Dirac $\delta(t)$

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases} \quad (1.7)$$

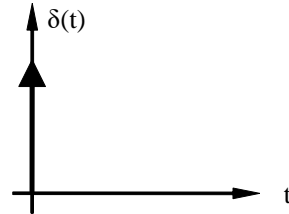


Figura 1.5 Función delta

Función Delta de Dirac retardada

$$\delta(t-t_0) = \begin{cases} 0 & t \neq t_0 \\ \infty & t = t_0 \end{cases} \quad (1.8)$$

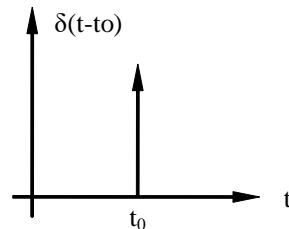


Figura 1.6 Función delta retardada

La función Delta de Dirac también se puede definir de la ecuación (1.6) como

$$\delta(t) = \lim_{a \rightarrow 0} P(t) \quad (1.9)$$

1.5 PROPIEDADES DE LA FUNCIÓN IMPULSO

1. Área bajo la función

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (1.10)$$

2. Esta función tiene la propiedad de aislar o reproducir un valor particular de una función $f(t)$, a esta propiedad también se le conoce como propiedad de muestreo

$$\int_{-\infty}^{\infty} f(t) \delta(t - t_0) dt = f(t_0) \quad \forall t_0 \in t \quad (1.11)$$

3. La función delta también tiene la propiedad de reproducir cualquier función $x(t)$

$$\int_{-\infty}^{\infty} x(\tau) \delta(t - \tau) d\tau = x(t) \quad (1.12)$$

esta propiedad también se le conoce como el teorema de la convolución. Esto significa que la convolución de una señal con la función impulso es la misma señal, es decir, $x(t) = x(t) * \delta(t)$, donde el símbolo $*$ representa a la operación convolución.

4. Derivada de la función escalón

$$\delta(t) = \frac{dU(t)}{dt} \quad (1.13)$$

1.6 RESUMEN

En este capítulo se han descrito algunos conceptos importantes de las señales y funciones continuas, éstos servirán para ir desarrollando los principios del procesamiento digital de señales. Aunque no se ha mostrado toda la teoría de señales continuas, consideramos que lo mencionado anteriormente es la base para introducir el mundo de las señales y sistemas discretos.

1.7 PROBLEMAS PROPUESTOS

- 1 Utilizando Fourier calcular el espectro de una señal muestreadora $S(t)$ que es un tren de impulsos separados un período T .
- 2 Comprobar analíticamente que de una sumatoria de Fourier se puede obtener la Transformada de Fourier.
- 3 Realizar el procedimiento analítico para obtener los coeficientes de la serie Exponencial de Fourier a partir de la serie Trigonométrica de Fourier.
- 4 Obtener analíticamente la función delta de Dirac a partir de otras funciones, mostrar al menos tres posibilidades.
- 5 Demostrar el teorema de la convolución.

CAPITULO 2

CONVERSIÓN ANÁLOGO DIGITAL

El muestreo es el proceso que implica estar adquiriendo muestras a intervalos específicos de tiempo de una señal analógica. Estas muestras (señal discreta) contienen diferentes magnitudes y son cuantizadas a valores digitales basados en la longitud de una palabra de L bits. A cada muestra discretizada se le asocia un valor digital de L bits, a este proceso se llama *cuantización*, el cual involucra errores que dependen del número de bits empleados por la palabra digital; para efectuar la cuantización hay que hacer una aproximación de los valores analógicos correspondientes, esta aproximación se hace por truncamiento o redondeo para obtener la palabra digital.

2.1 MUESTREO DE UNA SEÑAL EN EL TIEMPO CONTINUO

Sea una señal continua $x(t)$ limitada en banda, es decir, $|X_B(\omega)| = 0$ para $|\omega| > B$, si tomamos muestras a intervalos T espaciadas con respecto a la frecuencia más alta de la señal $x(t)$, entonces ésta se podrá reconstruir a partir de sus muestras. Para efectuar el muestreo se puede multiplicar la señal $x(t)$ por un tren de impulsos unitarios en el tiempo continuo

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2.1)$$

es decir

$$x_s(t) = x(t) \delta_T(t) = \sum_{n=-\infty}^{\infty} x(nT) \delta(t - nT) \quad (2.2)$$

Calculando la Transformada de Fourier

$$X_s(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s); \quad \omega_s = \frac{2\pi}{T} \quad (2.3)$$

donde: ω_s : es la frecuencia de muestreo
 T: el período de muestreo

$X_s(\omega)$: es una función periódica en el dominio de la frecuencia, compuesta por réplicas desplazadas de $X(\omega)$ y ponderada por $1/T$.

Utilizando el teorema de la convolución, se encuentra que la transformada de una secuencia discreta está dada por

$$X_s(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(\omega) \delta(\omega - n\omega_s) \quad (2.4)$$

La ecuación (2.4) significa que la convolución de $X(\omega)$ con un tren de impulsos corresponde a la suma de repeticiones de $X(\omega)$ centradas en cada impulso desplazado $n\omega_s$, el teorema de Nyquist se deduce de la figura 2.1, es decir, que las repeticiones del espectro no deben solaparse para evitar la superposición de frecuencias.

De la figura 2.1 se observa que si

$$|\omega_s - B| > B \Rightarrow \omega_s > 2B \quad (2.5)$$

La ecuación (2.5) corresponde al teorema del muestreo donde a la desigualdad $\omega_s > 2B$ se le conoce como frecuencia de *Nyquist o de Shannon*. El teorema de Nyquist garantiza que a partir de las muestras $x(nT)$ para $n = 0, \pm 1, \pm 2, \pm 3, \dots$ se podrá reconstruir a la señal original $x(t)$ de la siguiente forma:

- a) Generando un tren de impulsos cuyas amplitudes son las muestras sucesivas de $x(t)$.
- b) Alimentando el tren de impulsos a un filtro paso bajas con ganancia T y frecuencia de corte ω_s , donde $B < \omega_s < \omega_s - B$, entonces la salida resultante será igual a $x(t)$.

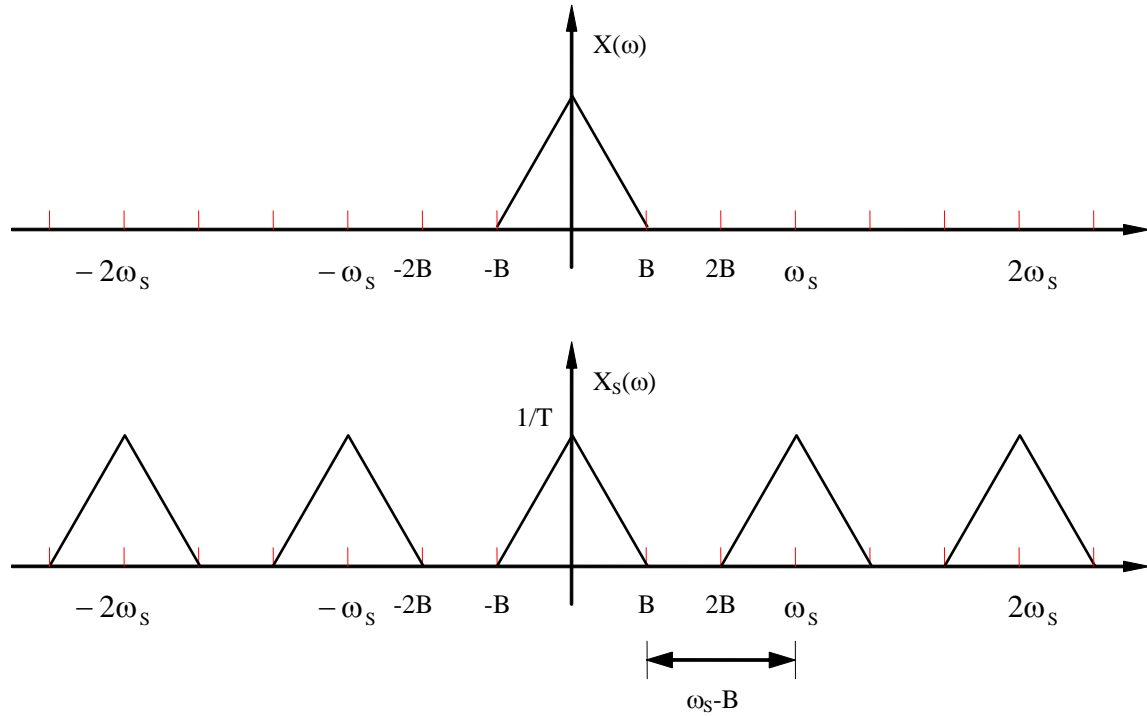


Figura 2.1 Espectro de una señal muestreada

De la figura 2.1, se puede deducir además que el espectro de una señal muestreada tiene las propiedades de ser: periódico, continuo y simétrico.

2.2 TEOREMA DE NYQUIST O SHANNON

Como se dedujo anteriormente, este teorema establece que una señal continua con un ancho de banda limitado a $f < f_s/2$ puede ser reconstruida de la señal muestreada si la razón de muestreo es al menos dos veces mayor que el ancho de banda de la señal de entrada.

En la figura 2.2 se muestra una señal continua $x(t)$ que va a ser muestreada; en la figura 2.3 se tiene la señal muestreada a intervalos de 0.5 unidades de tiempo y en la figura 2.4 el muestreo de $x(t)$ a 0.1 unidades de tiempo. En el primer caso se puede considerar que el muestreo cumple con el teorema de Nyquist, en el segundo caso también se cumple, sin embargo se ha excedido en muestras y es lo que se conoce como sobremuestreo.

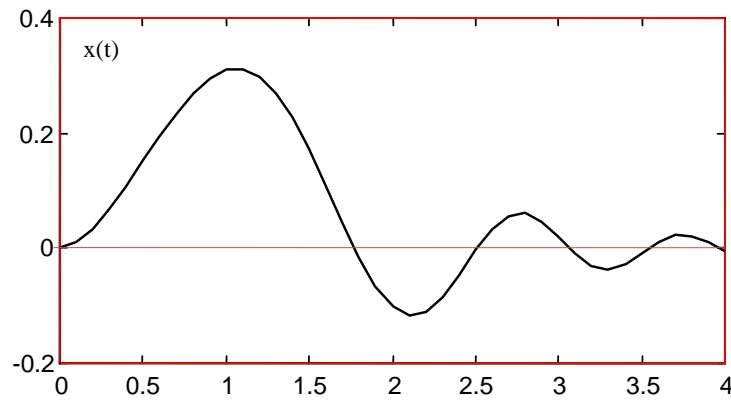


Figura 2.2 Señal analógica

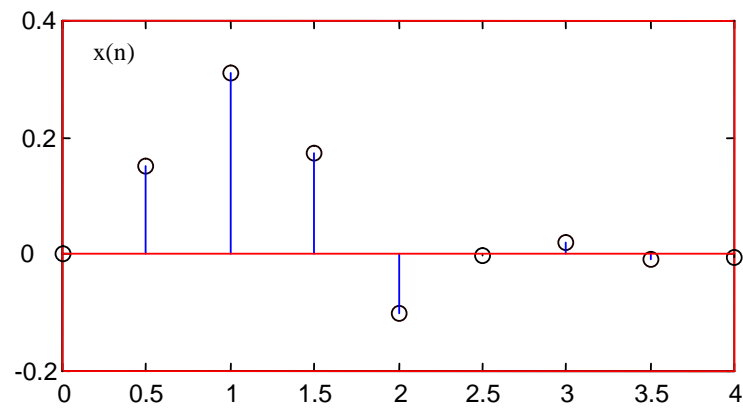


Figura 2.3 Señal muestreada cada 0.5 unidades en el tiempo

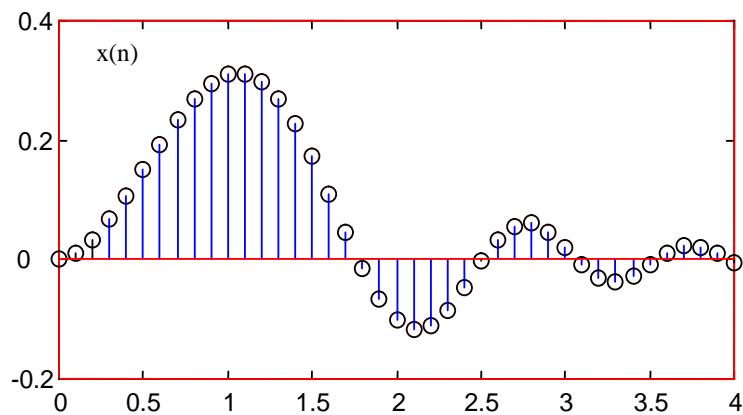


Figura 2.4 Señal muestreada cada 0.1 unidades en el tiempo

De las propiedades de la función impulso, cualquier señal continua $x(t)$ puede representarse como

$$x(t) = \int_{-\infty}^{\infty} x(\tau) \delta(t - \tau) d\tau \quad (2.6)$$

por el teorema de convolución

$$x(t) = x_1(t)x_2(t) \leftrightarrow \frac{1}{2\pi} X_1(\omega) * X_2(\omega) \quad (2.7)$$

recordando una función muestreadora, en el tiempo y la frecuencia se expresan en la ecuación (2.8)

$$\sum_{n=-\infty}^{\infty} \delta(t - nT) \leftrightarrow \omega_0 \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0) \quad \omega_0 = \frac{2\pi}{T} \quad (2.8)$$

muestreando una señal $x(t)$ y aplicando el teorema de convolución (2.7), se obtiene

$$X_t(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(\omega) * \delta(\omega - n\omega_0) \quad (2.9)$$

es decir

$$X_t(\omega) = X(\omega) \left(\frac{1}{T} \right) * \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0) \quad (2.10)$$

utilizando la propiedad de convolución de la función delta

$$x(t) = x(t) * \delta(t) \quad (2.11)$$

$$x(t - T) = x(t) * \delta(t - T) \quad (2.12)$$

entonces el espectro de la señal muestreada queda

$$X_t(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_0) \quad (2.13)$$

donde la ecuación (2.13) obtenida analíticamente muestra de nuevo que el espectro de una señal muestreada es continuo, periódico y simétrico.

2.3 ESPECTRO DE UNA SEÑAL MUESTREADA

Otra forma de obtener analíticamente el espectro de una señal muestreada es aplicar directamente la transformada de Fourier a la señal muestreada y posteriormente aplicar algunas propiedades. Dada la definición de la transformada de Fourier directa aplicada a una señal continua

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (2.14)$$

y transformada inversa

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad (2.15)$$

considerando una señal muestreada $x_s(t)$

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT) \delta(t-nT) \quad (2.16)$$

reemplazando en la Transformada de Fourier

$$X_s(f) = \int_{-\infty}^{\infty} \left[\sum_{n=-\infty}^{\infty} x(nT) \delta(t-nT) \right] e^{-j2\pi ft} dt = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT) e^{-j2\pi ft} \delta(t-nT) dt \quad (2.17)$$

aplicando las propiedades de la función delta se obtiene el espectro de la señal muestreada, es decir de la señal discreta

$$X_s(f) = \sum_{n=-\infty}^{\infty} x(nT) e^{-j2\pi fnT} \quad (2.18)$$

ésta es una función periódica con período $1/T$ cuyos coeficientes son

$$x(nT) = \frac{1}{T} \int_{-1/2T}^{1/2T} X_s(f) e^{j2\pi fnT} df \quad (2.19)$$

Cuando la señal continua $x(t)$ no es limitada en banda, el teorema del muestreo se burla, es decir, que se produce la superposición espectral de las repeticiones consecutivas de $X(\omega)$. A este fenómeno se le conoce como "aliasing", donde los componentes en frecuencia cercanos al límite de la banda se modifican. En la figura 2.5 se muestra un ejemplo gráfico del fenómeno de aliasing en el tiempo, el efecto "aliasing" para dos señales, se presenta para el muestreo cada $\pi/8$; en la figura 2.5 existen dos puntos comunes de muestreo que no se distingue a que señal corresponde.

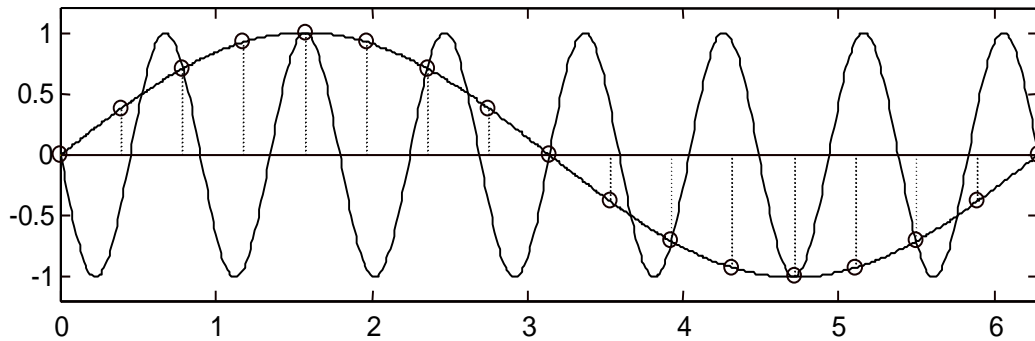


Figura 2.5 El fenómeno de “aliasing” en el tiempo

Para evitar el efecto de “aliasing”, se aplica un filtro paso bajas (FPB) a la señal de entrada para limitarla en banda. De la figura 2.5 se observa que un submuestreo ocasiona el efecto de “aliasing” y se presenta un solapamiento espectral, esto sucede cuando se consideran las muestras que coinciden con ambas señales. Por otro lado, un sobremuestreo (una frecuencia de muestreo muy alta) conduce a una cantidad excesiva de muestras ocasionando que un sistema digital sea insuficiente para almacenarlas.

2.4 CUANTIZACION

La magnitud de la amplitud de una señal muestreada o señal discreta pertenece al dominio de los reales, el proceso de cuantización consiste en representar la amplitud de la señal en un conjunto de valores no continuos, es decir, que no pertenecen a los reales sino que normalmente a los enteros, en general estos números se representa en forma binaria. Entonces cualquier magnitud de valor real se aproxima a un valor entero, trasladando valores de una escala infinita a una escala finita, en este proceso obviamente se produce un error de cuantización $e_q(n)$ definido por

$$e_q(n) = x(n) - x_q(n) \quad (2.20)$$

donde $x_q(n)$ es el valor cuantizado de $x(n)$.

Las muestras son cuantizadas en amplitud en forma binaria, también se dice que son codificadas. Para N niveles ($N < 2^L$) se necesitan $L = \log_2 N$ bits por muestra si N es potencia exacta de 2, ó $L = \text{Int}(\log_2(N)) + 1$ si N no es potencia exacta de 2. Si los niveles no son equiprobables y la probabilidad del nivel de salida es conocida, se puede utilizar la codificación de Huffman, llamada codificación de entropía, para mejorar la eficiencia de la codificación. En la cuantización de la amplitud de la señal muestreada resulta una compresión

de los datos, pero esto introduce alguna distorsión en la forma de onda o pérdida en la fidelidad.

2.5 MODULACIÓN POR CODIFICACIÓN DE PULSOS (PCM)

Sea $x(t)$ la función a muestrear, $x(n)$ es la muestra tomada a razón de $f_s \geq 2f$ y f es la frecuencia máxima del espectro de $x(t)$ o ancho de banda. En codificación PCM cada muestra de la señal es cuantizada a 2^L niveles de amplitud, donde L es el número de bits usados para representar a cada muestra. Entonces, la razón de codificación de la fuente $x(t)$ es Lf_s bits por segundo.

El proceso de muestreo puede modelarse matemáticamente de la siguiente forma

$$x(n) = x_q(n) + e_q(n) \quad (2.21)$$

donde $x_q(n)$: es la cuantización de la muestra

$e_q(n)$: error de cuantización que puede considerarse como ruido aditivo a L bits

En la figura 2.6 se muestran los niveles de cuantización utilizando tres bits.

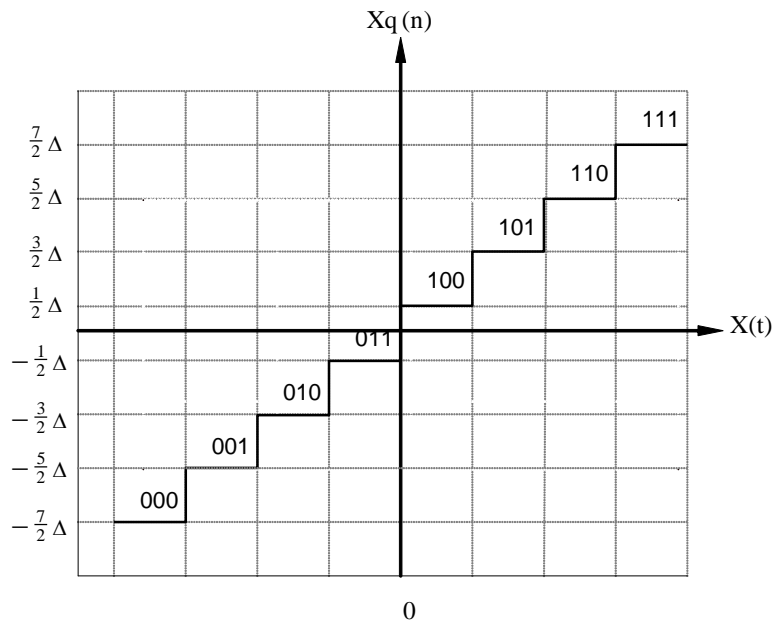


Figura 2.6 Cuantización utilizando tres bits

El ruido de cuantización es caracterizado estadísticamente por una distribución de densidad de probabilidad (PDF) uniforme, como se observa en figura 2.7, donde Δ es el paso de cuantización y depende del número de bits utilizados, $\Delta = 2^{-L}$

$$p(q) = \frac{1}{\Delta} \quad -\frac{\Delta}{2} \leq q \leq \frac{\Delta}{2} \quad (2.22)$$

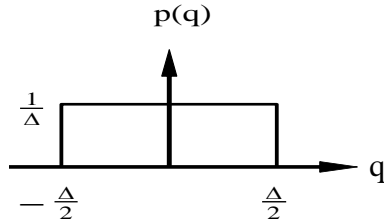


Figura 2.7 Función de densidad de probabilidad del ruido de cuantización

$$\int_{-\Delta/2}^{\Delta/2} p(q) dq = 1 = \int_{-\Delta/2}^{\Delta/2} \frac{1}{\Delta} dq = 1 \quad (2.23)$$

El error medio de cuantización se calcula como

$$E(q) = \int_{-\Delta/2}^{\Delta/2} q p(q) dq = \frac{q^2}{2} \left(\frac{1}{\Delta} \right) \Big|_{-\Delta/2}^{\Delta/2} = 0 \quad (2.24)$$

y el valor medio cuadrático

$$E(q^2) = \int_{-\Delta/2}^{\Delta/2} q^2 \left(\frac{1}{\Delta} \right) dq = \frac{q^3}{3} \left(\frac{1}{\Delta} \right) \Big|_{-\Delta/2}^{\Delta/2} = \frac{2}{3} \frac{\Delta^3}{8} \frac{1}{\Delta} = \frac{\Delta^2}{12} \quad (2.25)$$

donde Δ también corresponde a la resolución de la cuantización y si $\Delta = 2^{-L}$ para un ancho de palabra binaria de L bit, entonces

$$E(q^2) = \frac{2^{-2L}}{12} = \frac{2^{-2(L+1)}}{3} \quad (2.26)$$

midiéndolo en decibeles se tiene

$$10 \log(E(q^2)) = 10 \log \frac{\Delta^2}{12} = 10 \log \frac{2^{-2L}}{12} \approx -6L - 10.8 \text{ db} \quad (2.27)$$

2.6 RELACIÓN SEÑAL A RUIDO DE CUANTIZACIÓN

La calidad de salida de un convertidor análogo digital (A/D) se mide por la relación de la señal al ruido de cuantización (SQNR), el cual está dado por la *razón de potencia* de la señal x a la potencia del ruido q de cuantización:

$$\text{SQNR} = \frac{P_x}{P_q} \quad (2.28)$$

Para una señal que abarque el intervalo completo de entrada al convertidor, como una señal coseno de amplitud A , $P_x = A^2/2$ y $P_q = (A^2/3)2^{-2(L+1)}$, entonces:

$$\text{SQNR} = \frac{P_x}{P_q} = \frac{3}{2} 2^{2(L+1)} \quad (2.29)$$

en decibeles

$$\text{SQNR}_{\text{db}} = 10 \log(\text{SQNR}) = 1.76 + 6.02(L+1) \quad (2.30)$$

Se observa que SQNR se incrementa 6 db por cada bit agregado a la longitud de la palabra. Una SQNR de 70db es aceptable, por tanto en la mayoría de aplicaciones es suficiente utilizar 12 bits en la cuantización de una señal $x(n)$. La relación anterior es de mucha importancia ya que indica el número de bits para aplicaciones específicas, por ejemplo, el sistema de disco compacto muestrea a 44.1 kHz con resolución a 16 bits, lo que implica más de 96 db para SQNR.

Como el *error de cuantización* en el proceso de conversión A/D está dado por la diferencia entre el valor cuantizado y el valor verdadero de la muestra $e_q(n) = x_q(n) - x(t)$. El error máximo está dado por la mitad del paso de cuantización $e_q(n) = (\Delta/2)$, donde Δ es la distancia entre dos sucesivos niveles de cuantización, es decir, es la resolución de la conversión A/D.

2.7 COMPRESIÓN LOGARÍTMICA

Algunas señales como la voz, se caracterizan por presentar con más frecuencia amplitudes pequeñas que amplitudes grandes, ocasionando redundancia de información. En cuantización PCM cada muestra de la señal es codificada independientemente de las otras muestras, es decir, que una cuantización uniforme produce un mismo espaciado entre niveles sucesivos a través del intervalo dinámico de la señal. Para una cuantización de L bits una mejor aproximación consiste en hacer estos niveles más cerrados, es decir, menos espaciados en las amplitudes pequeñas y más espaciados en las amplitudes grandes. Esto produce una cuantización no uniforme del error, ocasionando un valor medio cuadrático menor del error que el indicado en la ecuación de $E(q^2)$. Una cuantización no uniforme es obtenida pasando la señal a través de un dispositivo no lineal que comprime la amplitud de la señal, como se observa en la figura 2.8 y utilizando la ecuación (2.31)

$$|y| = \frac{\log(1 + \mu|x|)}{\log(1 + \mu)} \quad (2.31)$$

donde:

$|x|$: magnitud de entrada

$|y|$: magnitud de salida

μ : parámetro seleccionado para las características de compresión

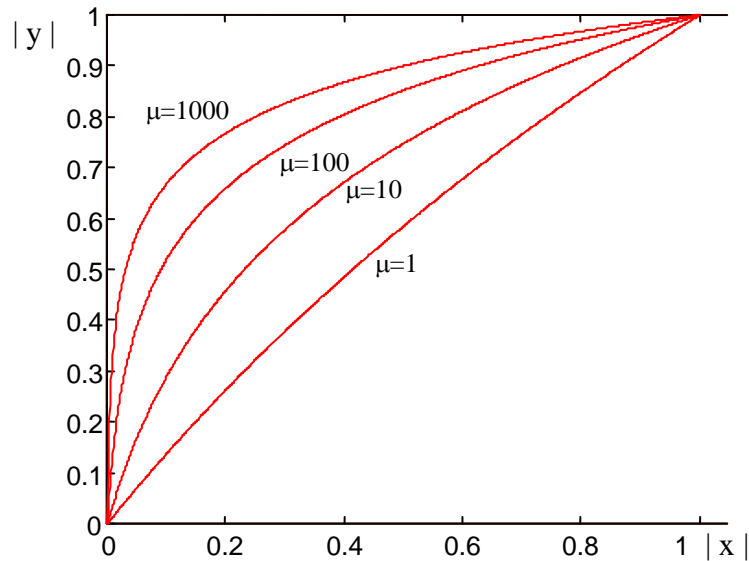


Figura 2.8 Relación de compresión logarítmica

Este tipo de codificador permite comprimir una señal de entrada de 13 bits a una señal de salida con intervalo dinámico de 7 bits. Para la codificación de señales de voz en USA, Canadá y Japón se ha adoptado el estándar de $\mu = 255$, resultando una reducción de 24 db en la cuantización de la potencia del ruido relativo a la cuantización uniforme [PRO95]. En la reconstrucción de la señal se utiliza una relación logarítmica inversa para expandir la señal. La relación compresor-expansor se conoce como *compandor*.

El estándar europeo es llamado Ley A, y está definido por la ecuación (2.32)

$$|y| = \begin{cases} \frac{A|x|}{1 + \text{Ln}(A)} & \text{para } 0 \leq x \leq \frac{1}{A} \\ \frac{\text{Ln}(1 + A|x|)}{\text{Ln}(1 + A)} & \text{para } \frac{1}{A} \leq x \leq 1 \end{cases} \quad (2.32)$$

donde $A=87.6$

En la figura 2.9 se muestran las gráficas entrada salida de la Ley μ y la Ley A, por lo que se observa que son muy similares.

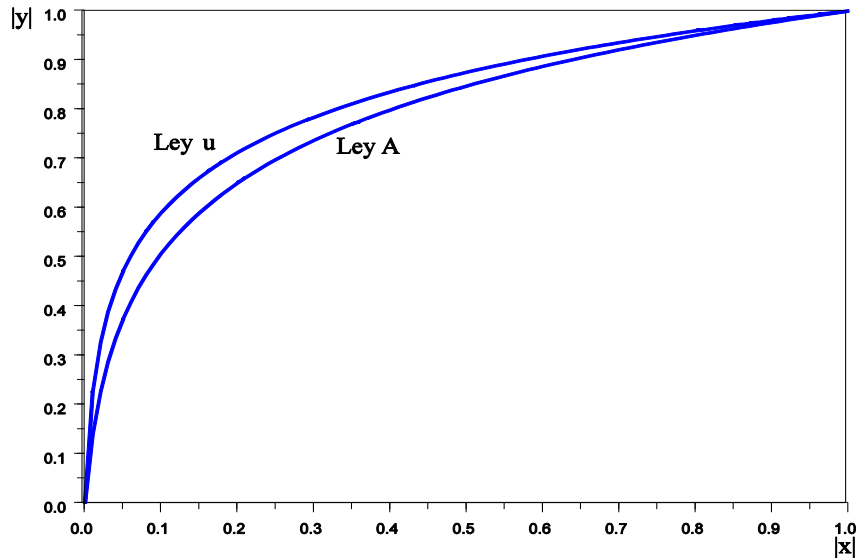


Figura 2.9 Ley μ y ley A

2.8 RECUPERACIÓN DE UNA SEÑAL ANALÓGICA A PARTIR DE UNA SEÑAL MUESTREADA

Si se tiene una señal discreta $x(n)$ que cumple con el teorema del muestreo, teóricamente podemos realizar el proceso contrario al muestreo para recuperar una señal continua $x(t)$ a partir de $x(n)$, haciendo pasar la señal $x(n)$ a través de un filtro ideal paso bajas (FIPB) cuya frecuencia de corte ω_c sea mayor o igual a la frecuencia de muestreo ω_s como se ilustra en la figura 2.10.

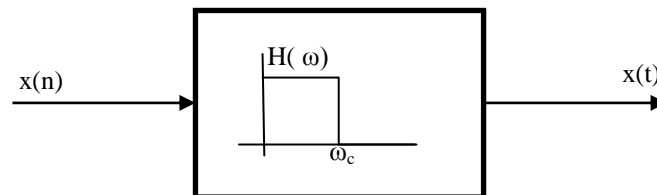


Figura 2.10 Filtro ideal paso bajas para recuperar una señal continua

Si se utiliza el análisis de Fourier se puede calcular la transformada inversa de la respuesta en frecuencia del FIPB $H(\omega)$ para obtener $h(t)$ y convolucionar la entrada $x(n)$ para obtener la salida analógica $x(t)$, este proceso se realiza a través de la ecuación (2.33)

$$x(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2\omega}\right) \frac{\sin\left(2\pi\omega\left(t - \frac{n}{2\omega}\right)\right)}{2\pi\omega\left(t - \frac{n}{2\omega}\right)} \quad (2.33)$$

Donde $x(n/2\omega)$ denota el muestreo de la señal $x(t)$ tomando muestras a razón de $\omega_s \geq 2\omega$.

2.9 RESUMEN

En este capítulo se han tratado algunos aspectos del muestreo de señales, principalmente el teorema del Nyquist el cual siempre se debe respetar, cuando se trabaja con señales discretas casi todos los procesos a realizar tienen una dependencia de la frecuencia de muestreo. Es importante mencionar que hasta esta parte de las notas se puede considerar que ya se está tratando con señales discretas y se puede empezar a realizarles algún proceso.

También se ha visto el proceso de cuantización donde se obtienen valores representados digitalmente en algún formato de representación numérica, si se quiere profundizar en esta parte se sugiere al lector que consulte el capítulo 10.

2.10 PROBLEMAS PROPUESTOS

1. Obtener analíticamente la función delta de Dirac a partir de otras funciones, mostrar al menos tres posibilidades.
2. Las señales de: temperatura ambiente, voz, audio y video se muestrean a la frecuencia de Nyquist, calcular la cantidad de información obtenida en cada caso si se muestrean a ocho, 12 y 16 bits.
3. Si una señal continua $x(t)$ limitada en banda se muestrea a una frecuencia $f_s > B$ se muestrea con un tren de pulsos unitarios con ciclo de servicio de $0.25T$, determinar su espectro.
4. Describir y explicar el funcionamiento de al menos dos tipos de convertidores AD.
5. Mostrar y explicar la función de transferencia de los convertidores AD con error de medio bit, un bit y dos bits.
6. Enumerar y explicar todos los errores que se producen en el proceso de conversión de una señal analógica a digital.

CAPITULO 3

SEÑALES DISCRETAS

Con base a la teoría estudiada en los capítulos anteriores, en este capítulo se introducen los conceptos relacionados con las señales discretas, se definen las formas de representarlas, algunas características importantes, se dan a conocer algunas señales de interés en el procesamiento digital de señales con las que podemos definir otras señales, las operaciones básicas que se pueden realizar entre señales y algunas funciones o señales discretas.

3.1 DEFINICIÓN DE SEÑALES DISCRETAS

Una señal discreta se define como una función que mapea un conjunto de números enteros en una línea o escala real para señales con valores reales o en un plano complejo para señales con valores complejos [KUU05]. Las señales del dominio discreto n son típicamente versiones muestreadas de señales analógicas obtenidas mediante la discretización, el dominio n es el dominio de los enteros. Una señal discreta significará una secuencia de números o muestras, es decir, dada una señal analógica $x(t)$ a través del proceso de discretización se puede obtener la señal discreta $x(n)$ (figura 3.1) si el período de muestreo $T=1$.

Una vez realizada la discretización, la señal discreta no es más que una secuencia de números o muestras de $x(t)$, donde el proceso de discretización es conocido como muestreo. En el PDS las variables o señales discretas son representadas como una secuencia de números que pueden manipularse matemáticamente.

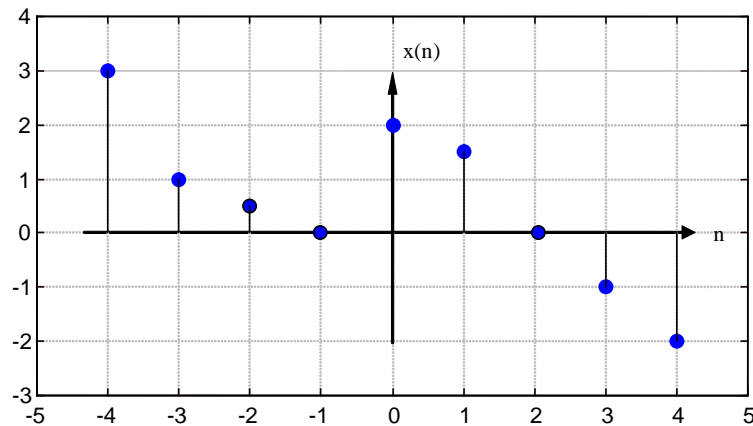


Figura 3.1 Señal discreta

3.2 FORMAS DE REPRESENTACIÓN DE UNA SEÑAL O FUNCIÓN DISCRETA

a) Como una función de n: $x(n)$ se define para varios valores de n o intervalos de n

$$x(n) = \begin{cases} A & \text{para } n = 1, 2, 3 \\ B & \text{para } n = 4, 5, 6 \\ C & \text{para } n > 6 \end{cases}$$

b) Como una tabla o representación tabular

n	...	-2	-1	0	1	2	3	4	...
x(n)	...	0	0	1	2	1	1	1	...

c) Como una secuencia donde el origen está marcado en $n = 0$

$$x(n) = \{ \dots 0, 0, 1, 4, 1, 0, 0, 1, 3, \dots \}$$

3.3 FUNCIONES DISCRETAS ESPECIALES O SECUENCIAS BÁSICAS

A continuación se definen algunas funciones discretas que se consideran de mucha importancia en el PDS y que son de utilidad en desarrollos posteriores.

IMPULSO UNITARIO O FUNCIÓN DELTA

La función impulso unitario es una secuencia que vale la unidad sólo cuando su argumento es cero

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (3.1)$$

Para retraso de tiempo igual a k

$$\delta(n-k) = \begin{cases} 1 & n = k \\ 0 & n \neq k \end{cases} \quad (3.2)$$

Para retraso de tiempo igual a uno

$$\delta(n-1) = \begin{cases} 1 & n = 1 \\ 0 & n \neq 1 \end{cases} \quad (3.3)$$

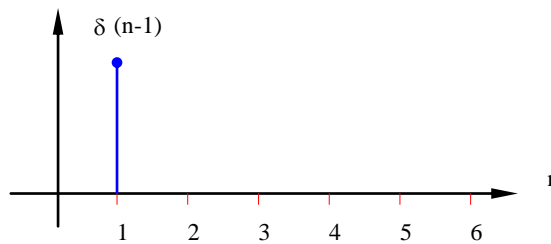


Figura 3.2 Función impulso unitario con retardado de uno

Una propiedad del impulso unitario es que permitirá caracterizar cualquier señal discreta $x(n)$ como

$$x(n) = \sum_{k=-\infty}^{\infty} x(k) \delta(n-k) \quad (3.4)$$

Es decir, que una aplicación de la función delta es que una secuencia arbitraria $x(n)$ puede ser representada como una sumatoria de impulsos unitarios adecuadamente desplazados y multiplicados por los valores respectivos de la secuencia $x(n)$.

Considerando la ecuación (3.4) y observando la figura 3.3 la secuencia $x(n)$ se puede escribir

$$x(n) = 3\delta(n+4) + \delta(n+3) + 0.5\delta(n+2) + 2\delta(n) + 1.5\delta(n-1) - \delta(n-3) - 2\delta(n-4) \quad (3.5)$$

o como una sumatoria

$$x(n) = \sum_{i=-4}^4 x(i)\delta(n-i) \quad (3.6)$$

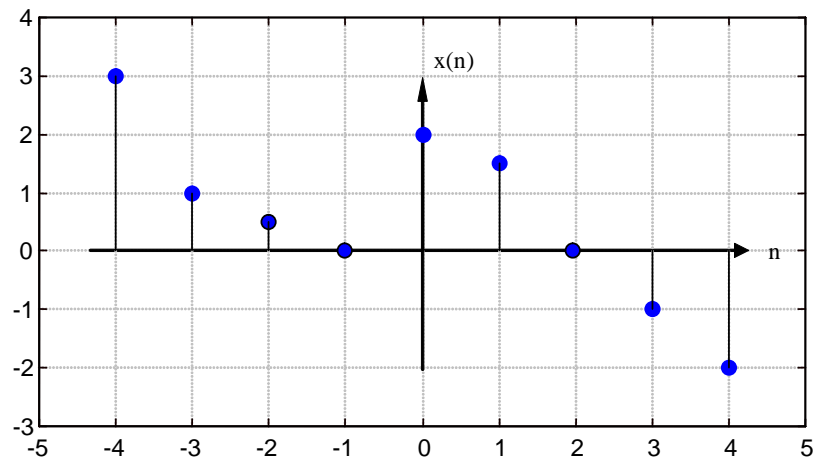


Figura 3.3 Señal discreta x(n)

ESCALÓN UNITARIO

Es una secuencia que toma el valor de la unidad para todos los argumentos de n mayores o iguales que cero y es igual a cero para los demás valores de n , y se define

$$U(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (3.7)$$

$U(n)$ es una función de impulsos desplazados, es decir, que está relacionada con el impulso unitario por una sumatoria de pulsos $\delta(n-k)$ con valor unitario para $n=k$, entonces la función escalón unitario se puede expresar como

$$U(n) = \sum_{k=0}^{\infty} \delta(n-k) \quad (3.8)$$

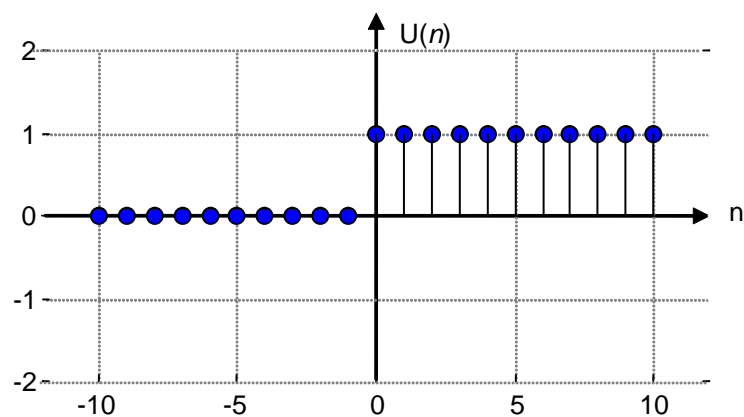


Figura 3.4 Señal o función escalón unitario U(n)

Similar a la función delta en el tiempo continuo, la función delta en el tiempo discreto se puede escribir como una diferencia de funciones escalón

$$\delta(n) = U(n) - U(n-1) \quad (3.9)$$

SECUENCIA RAMPA

Se define por la ecuación (3.10) y corresponde a una función creciente

$$x(n) = \begin{cases} n, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (3.10)$$

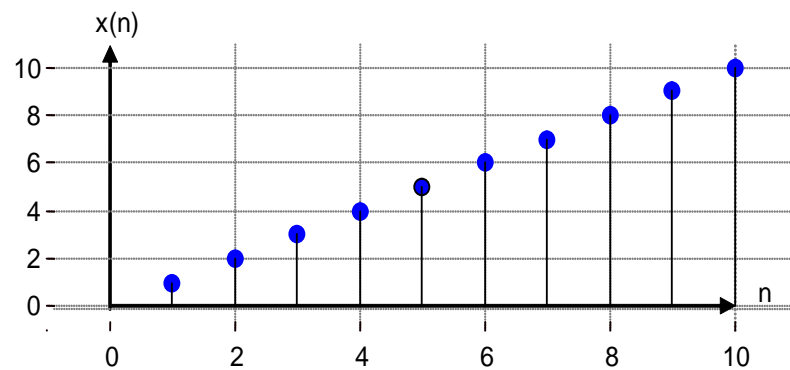


Figura 3.5 Secuencia rampa

FUNCIÓN PERIÓDICA

Una señal periódica de período T_0 , es aquella que cumple con

$$x(n) = x(n + T_0) \quad (3.11)$$

Si el período de muestras es T , entonces

$$x(n) = x(nT) = x(nT + T_0); \forall n \quad (3.12)$$

dado que el dominio de $x(n)$ es el conjunto de los enteros, entonces se requiere que

$$nT + T_0 = mT; m \in \mathbb{Z} \quad (3.13)$$

que se logra con

$$T_0 = NT; mT = (n+N)T \Rightarrow m = n+N \quad (3.14)$$

Sustituyendo en (3.11)

$$x(n) = x(nT + NT) = x(n + N); \quad \forall n \quad \text{con } T=1 \quad (3.15)$$

Entonces la secuencia $x(n)$ es una secuencia periódica con período N , donde N es un número entero.

FUNCIONES SINUSOIDALES EN EL TIEMPO DISCRETO

Una señal o secuencia sinusoidal o cosenoidal se define como:

$$x(n) = A \cos(\Omega n + \phi) = A \cos(2\pi n f + \phi) \quad (3.16)$$

donde:

- Ω = frecuencia en radianes por muestra
- n = número de muestra
- A = amplitud de señal
- ϕ = fase en radianes
- f = ciclos por cada muestra

PROPIEDADES

a) Periodicidad:

La señal sinusoidal $x(n)$ es periódica si y sólo si f es un número racional, es decir, si $x(n)$ es periódica con período $N > 0$, entonces $x(n+N) = x(n)$ para todo n , donde el valor más pequeño es llamado período fundamental, entonces

$$f_0 = \frac{m}{N} \quad \text{ó} \quad N = \frac{m}{f_0} \quad (3.17)$$

Una señal sinusoidal en el tiempo discreto es periódica si y sólo si, su frecuencia f_0 puede ser expresada como una relación de dos números enteros, es decir que f_0 es un número racional. Para determinar el período fundamental N de una señal periódica, se expresa la frecuencia como una relación $f_0 = m/N$, hasta que se obtenga una mínima expresión de tal forma que m y N no tengan factores comunes, así, el período fundamental será igual a N . Se puede apreciar que pequeños cambios en la frecuencia implican grandes cambios en el período.

b) Las señales sinusoidales con frecuencias separadas por un múltiplo entero de 2π , son idénticas

$$x(n) = \cos((\Omega_0 + 2\pi)m n + \phi) = \cos((\Omega_0 n + \phi) + 2\pi m) \quad (3.18)$$

desarrollando la ecuación (3.18)

$$x(n) = \cos(\Omega_0 n + \phi) \cos(2\pi m) - \text{sen}(\Omega_0 n + \phi) \text{sen}(2\pi m) \quad (3.19)$$

donde el término $\cos(2\pi n)$ es uno para todo n y $\text{sen}(2\pi n)$ es cero para todo n .

Como $x(n) = \cos((\Omega_0 + 2\pi)n + \phi) = \cos(\Omega_0 n + \phi)$, se concluye que todas las sinusoidales $x(n) = A \cos(\Omega_n n + \phi)$; $n = 0, 1, 2, \dots$ donde $\Omega_n = \Omega_0 + 2\pi n$ para $-\pi < \Omega_0 < \pi$, son idénticas.

Entonces las secuencias de dos sinusoidales cualesquiera, con frecuencias en el intervalo $-\pi \leq \Omega \leq \pi$ o $-1/2 \leq f \leq 1/2$, son distintas. Por lo tanto, las señales sinusoidales en el tiempo discreto con frecuencias $|\Omega| \leq \pi$ o $|f| \leq 1/2$ son únicas. Cualquier secuencia resultante de una señal sinusoidal con una frecuencia $|\Omega| > \pi$ o $|f| > 1/2$ es idéntica a la secuencia obtenida de una senoide con frecuencia $|\Omega| < \pi$. Debe recordarse que en el tiempo continuo las señales sinusoidales son distintas para ω ó F dentro del intervalo de variación $-\infty < \omega < \infty$ o $-\infty < F < \infty$.

c) Las tasas de oscilación más altas en señales sinusoidales en el tiempo discreto, son alcanzadas cuando $\Omega = \pi$ ó $\Omega = -\pi$, o de manera equivalente, $f = 1/2$ o $f = -1/2$.

Esta propiedad se ilustra analizando las características de la secuencia $x(n) = \cos(\Omega_0 n)$ para valores de Ω_0 que van de 0 a 2π . Si se toma

$$\Omega_0 = 0, \frac{1}{8}\pi, \frac{1}{4}\pi, \frac{1}{2}\pi, \pi, \frac{3}{2}\pi, \frac{7}{4}\pi, \frac{15}{8}\pi, 2\pi \quad (3.20)$$

que corresponde a las frecuencias discretas

$$f_0 = 0, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \frac{15}{16}, 1 \quad (3.21)$$

Para las señales sinusoidales con periodos de $N = \infty, 16, 8, 4, 2, 4, 8, 16, 1$, su representación gráfica se muestra en la figura 3.6. Se verifica que el período de las secuencias sinusoidales se decrementa cuando la frecuencia se incrementa. De hecho, las tasas de oscilación se incrementan cuando las frecuencias se incrementan (esto cuando Ω_0 varía de 0 a π). Cuando Ω_0 varía de π a 2π , la tasa de oscilación de las señales sinusoidales se decrementa [PRO92].

Similar a las señales sinusoidales en el tiempo continuo, en el caso discreto, también se pueden considerar frecuencias negativas, haciendo uso de la siguiente relación

$$x(n) = A \cos(\Omega n + \phi) = \frac{A}{2} e^{j(\Omega n + \phi)} + \frac{A}{2} e^{-j(\Omega n + \phi)} \quad (3.22)$$

Ya que las sinusoidales en el tiempo discreto con frecuencias separadas por un múltiplo entero de 2π son idénticas, se concluye que las frecuencias en cualquier intervalo $\Omega_1 \leq \Omega \leq \Omega_1 + 2\pi$ constituyen todas las sinusoides o exponenciales complejas existentes en el tiempo discreto.

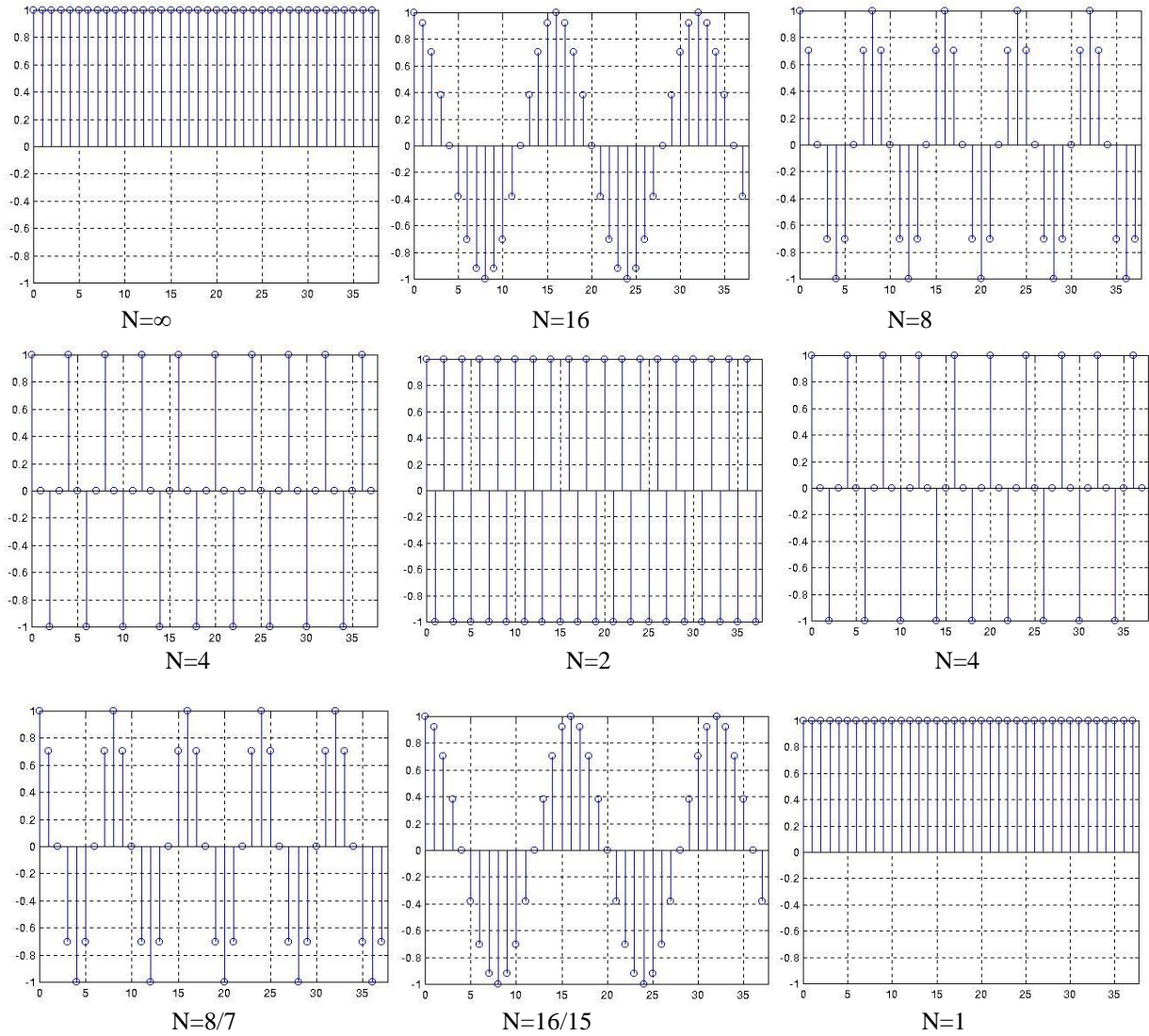


Figura 3.6 Señales senosoidales a diferentes períodos

EXPONENCIALES COMPLEJAS RELACIONADAS ARMÓNICAMENTE

Una secuencia exponencial compleja está dada por $x(n) = A\alpha^n$ donde $A = |A|e^{j\phi}$ y $\alpha = |\alpha|e^{j\omega_0}$, se tiene

$$x(n) = |A||\alpha|^n e^{j(\omega_0 n + \phi)} = |A||\alpha|^n \cos(\omega_0 n + \phi) + j|A||\alpha|^n \text{sen}(\omega_0 n + \phi) \quad (3.23)$$

Si $|A| \neq 1$, tanto la parte real como la parte imaginaria de $x(n)$ serán secuencias sinusoidales amortiguadas si $|\alpha| < 1$, o secuencias que se van amplificando si $|\alpha| > 1$, como se observa en la figura 3.7.

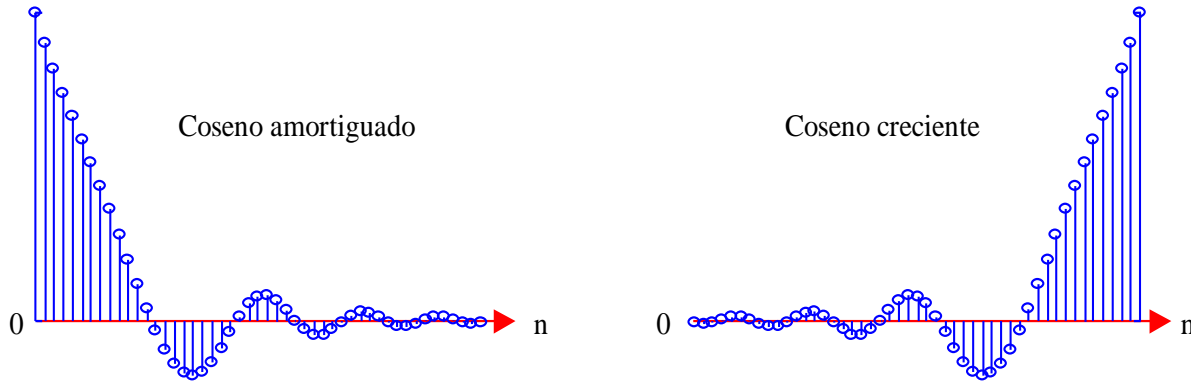


Figura 3.7 Señales coseno amortiguado y creciente

Para el caso $|\alpha| = 1$, la secuencia resultante se denomina sinusoidal compleja, y su expresión es más sencilla

$$x(n) = |A|e^{j(\omega_0 n + \phi)} = |A| \cos(\omega_0 n + \phi) + j|A| \sen(\omega_0 n + \phi) \quad (3.24)$$

Por otra parte, las consideraciones acerca de la periodicidad de las secuencias de tipo sinusoidal formuladas anteriormente se aplican de manera análoga a las sinusoides complejas, debido a que sus partes real e imaginaria, como puede verse en la ecuación (3.24) son de carácter sinusoidal.

SECUENCIAS TRUNCADAS

Cuando se lleva a cabo el procesamiento de señales discretas, es muy útil trabajar con una versión truncada de la secuencia completa, almacenándose en un vector los datos correspondientes a un intervalo de la secuencia. Además, en una computadora sólo se pueden almacenar secuencias finitas a las que normalmente se denominan vectores, por ejemplo, partiendo de la secuencia exponencial $x(n) = A\alpha^n$, mencionada anteriormente, se define el vector

$$\mathbf{x} = x(n) = A\alpha^n, 0 \leq n \leq N-1 \quad (3.25)$$

donde han quedado N muestras de la secuencia original.

Por otra parte, si la secuencia $x(n)$ es periódica de periodo N , se pueden obtener todos los valores de la secuencia a partir de los valores de cualquier intervalo de longitud N . De aquí que una representación de tipo vectorial como la apuntada anteriormente sea muy conveniente en este caso, puesto que nos proporciona toda la información necesaria acerca de la señal.

Las señales exponenciales complejas relacionadas armónicamente forman conjuntos de exponenciales complejas periódicas con frecuencias fundamentales que son múltiplos de una frecuencia positiva. Para una señal sinusoidal con $f_0 = 1/N$, se puede definir a los conjuntos exponenciales complejos relacionados armónicamente de la forma:

$$S_n(n) = e^{j2\pi n f_0 k}, \quad n = 0, \pm 1, \pm 2, \dots \quad (3.26)$$

Pudiéndose comprobar que $S_{n+N}(k) = S_n(k)$, de acuerdo con $x(n+N) = x(n)$ existirán únicamente N exponenciales complejas periódicas diferentes y todos los miembros del conjunto tendrán un periodo común de N muestras. Por lo que se puede escoger cualquier exponencial compleja consecutiva de $n = n_0$ a $n = n_0 + (N-1)$, para formar un conjunto relacionado armónicamente con frecuencia fundamental $f_0 = 1/N$. Si $n_0 = 0$, el conjunto será:

$$S_n(k) = e^{j2\pi kn/N}, \quad n = 0, 1, 2, \dots, N-1 \quad (3.27)$$

3.4 OPERACIONES BÁSICAS CON SECUENCIAS

En esencia se puede decir que existen tres operaciones básicas entre secuencias, la suma, el producto y el retardo, sin embargo, en esta sección se agregan otras operaciones menos usuales.

SUMA Y PRODUCTO DE DOS SECUENCIAS

Sean dos secuencias $x(n)$ e $y(n)$ se define la suma muestra a muestra de las dos secuencias para $-\infty \leq n \leq +\infty$, de la siguiente forma

$$s(n) = x(n) + y(n) = \sum_{k=-\infty}^{\infty} ((x(k) + y(k)) \delta(n-k)) \quad (3.28)$$

El producto muestra a muestra de las dos secuencias está dado por:

$$p(n) = x(n) y(n) = \sum_{k=-\infty}^{\infty} ((x(k) y(k)) \delta(n-k)) \quad (3.29)$$

CAMBIO DE AMPLITUD

Sea la secuencia $x(n)$ y una constante arbitraria A , cada muestra de la secuencia queda multiplicada por la constante A obteniéndose una nueva secuencia $y(n) = A x(n)$. Si $|A| > 1$ entonces $y(n)$ es una versión amplificadora de la secuencia original, si $|A| < 1$ entonces $y(n)$ es una versión atenuada de la original, si $|A| < 0$ se produce un cambio de polaridad.

DESPLAZAMIENTO LINEAL

Dada $x(n]$ y una constante entera n_0 , la secuencia desplazada $y(n) = x(n-n_0)$ será una traslación de n_0 unidades hacia la derecha si $n_0 > 0$ como se observa en la figura 3.8, o hacia la izquierda cuando $n_0 < 0$.

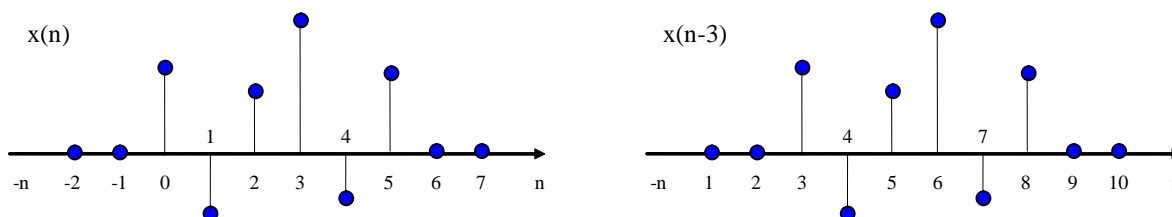


Figura 3.8 Desplazamiento o retardo lineal con $n_0 = 3$

DESPLAZAMIENTO CIRCULAR

Para secuencias truncadas, definidas sobre un intervalo, normalmente el intervalo de valores está comprendido entre 0 y $N-1$, como en la figura 3.9. El desplazamiento circular se define de la siguiente manera, suponiendo que $|n_0| < N$, la secuencia truncada desplazada circularmente consistirá inicialmente en una traslación de n_0 unidades hacia la derecha si $n_0 > 0$, o hacia la izquierda si $n_0 < 0$. A diferencia del desplazamiento lineal, no se origina la pérdida de un conjunto de valores por un extremo del vector, ya que esos valores se añaden por el otro extremo, tal como lo muestra la figura 3.9

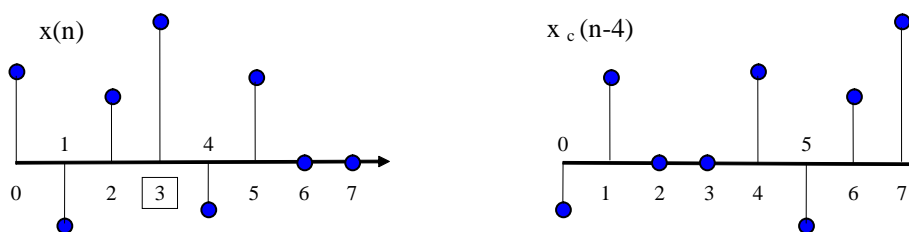


Figura 3.9 Desplazamiento Circular

Por otra parte, si $|n_0| \geq N$, debemos operar con el residuo de n_0 módulo N , es decir, el resto r_0 de la división entre n_0 y N , y efectuar el desplazamiento circular sustituyendo n_0 por r_0 .

REFLEXIÓN (FOLDING)

Para una señal discreta $x(n]$ la secuencia reflejada se denomina $x(-n)$, gráficamente esta transformación consiste en realizar un giro de 180° respecto del eje $n=0$ de la señal $x(n)$ respecto al

eje de ordenadas. Hay secuencias invariantes a la reflexión, es decir, $x(n) = x(-n)$ y se denominan secuencias pares.

SECUENCIAS ASOCIADAS

Dada una secuencia discreta y compleja $x(n)$, podemos definir a partir de ella las siguientes secuencias o extraer una parte de la secuencia:

Parte par	$x_e(n) = \frac{x(n) + x(-n)}{2}$	$x(n) = x_e(n) + x_o(n)$
Parte Impar	$x_o(n) = \frac{x(n) - x(-n)}{2}$	
Parte Real	$x_r(n) = \frac{x(n) + x^*(n)}{2}$	$x(n) = x_r(n) + jx_i(n)$
Parte Imaginaria	$x_i(n) = \frac{x(n) - x^*(n)}{2j}$	
Parte Hermítica	$x_h(n) = \frac{x(n) + x^*(-n)}{2}$	$x(n) = x_h(n) + jx_a(n)$
Parte Antihermítica	$x_a(n) = \frac{x(n) - x^*(-n)}{2j}$	

Donde $x^*(n)$ significa el conjugado complejo de $x(n)$. En relación con las definiciones anteriores, se puede trabajar con una secuencia $x(n)$ si cumple con las propiedades:

Par	$x(n) = x(-n)$	si	$x_o(n) = 0$
Impar	$x(n) = -x(-n)$	si	$x_e(n) = 0$
Real	$x(n) = x^*(n)$	si	$x_i(n) = 0$
Imaginaria	$x(n) = -x^*(n)$	si	$x_r(n) = 0$
Hermítica	$x(n) = x^*(-n)$	si	$x_a(n) = 0$
Antihermítica	$x(n) = -x^*(-n)$	si	$x_h(n) = 0$

3.5 RESUMEN

En este capítulo se han expuesto conceptos básicos sobre señales discretas, algunas señales discretas de interés y operaciones básicas entre señales, estos conceptos y señales serán de utilidad en capítulos posteriores para el análisis y diseño de sistemas discretos.

3.6 PROBLEMAS PROPUESTOS

1. Expresar una señal diente de sierra de longitud N en función de otras señales.
2. Expresar una señal triangular de altura máxima uno y longitud N en función de otras señales.
3. Determinar tres formas de obtener una función impulso.
4. Proponer una señal $x(n)$ y extraer la parte par $x_e(n)$ e impar $x_o(n)$, posteriormente reconstruir $x(n)$.
5. Proponer una señal $x(n)$ compleja y extraer la parte real $x_r(n)$ e imaginaria $x_i(n)$, posteriormente reconstruir $x(n)$.

CAPITULO 4

SISTEMAS DISCRETOS

En la actualidad la mayoría de señales se procesan digitalmente debido al incremento y flexibilidad de los sistemas digitales [KU05], como se describió en la introducción y en el capítulo dos, las señales discretas se representan digitalmente para posteriormente ser procesadas por alguna máquina digital. En este capítulo se describe los sistemas discretos, sus formas de representarlos y caracterizarlos, con base en la caracterización de señales y sistemas se definen las operaciones básicas que realiza un sistema. En general los sistemas discretos realizan tres operaciones que son las suma, la multiplicación y retardo, que en su conjunto definen la operación convolución, ésta última es fundamental en el procesamiento digital de señales.

4.1 SISTEMAS

Los principios fundamentales de la teoría de señales y sistemas se basan en la idea de hacer una abstracción del fenómeno en estudio utilizando ciertos principios matemáticos. La caracterización de cualquier fenómeno puede ser representada o conceptualizada como un conjunto de entradas a un bloque de proceso y un conjunto de salidas, dentro del bloque se efectúan todas las operaciones y transformaciones de las entradas para producir las salidas.

Un sistema está conformado de elementos o dispositivos físicos que realizan transformaciones u operaciones sobre las señales. En general un sistema es caracterizado por el tipo de operaciones que efectúa sobre la señal, por ejemplo si la operación es lineal, el sistema es llamado lineal. Un sistema discreto puede ser descrito en términos de ecuaciones de transformación que definen un algoritmo. En procesamiento digital de señales, las operaciones efectuadas sobre una señal consisten en un conjunto de operaciones matemáticas especificadas por el software del programa.

NOCIÓN DE SISTEMA

Dado un conjunto de señales de entrada “x” se relacionan con señales de salida “y” por medio de una regla que definirá al sistema. El sistema definido de esta forma será una correspondencia o transformación del conjunto de entradas “x” al conjunto de salidas “y”. Como en el dominio discreto las señales son secuencias numéricas, entonces un sistema discreto es realmente un algoritmo que toma las entradas y las transforma en salidas.

$$y(n) = H [x(n)] \quad (4.1)$$

Entrada
Operador
Salida del sistema

La señal $x(n)$ es transformada por el sistema en la señal $y(n)$, entonces el operador H en la ecuación (4.1) denota la transformación. Un ejemplo gráfico lo podemos observar en la figura 4.1.

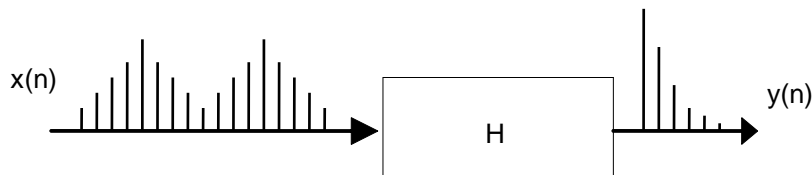


Figura 4.1 Representación de un sistema en forma entrada salida.

4.2 CARACTERIZACION DE UN SISTEMA

En general existen varias formas de caracterizar o describir un sistema discreto, estas representaciones son equivalentes y a través de herramientas matemáticas se puede ir de una representación a otra y dependiendo de cada autor pueden agruparse en forma diferente:

1. Ecuaciones en diferencias o relación entrada salida
2. Bloques básicos
3. Respuesta al impulso $h(n)$
4. Función de transferencia $H(z)$ (se verá en el capítulo 5 y 6)
5. Respuesta en frecuencia $H(e^{j\omega})$
6. Variables de estado

ECUACIÓN EN DIFERENCIAS O RELACIÓN ENTRADA SALIDA

La descripción entrada - salida del sistema es una expresión o regla matemática que define explícitamente la relación entre la entrada y la salida.

$$\begin{array}{ccc}
 & H & \\
 x(n) & \rightarrow & y(n)
 \end{array}$$

Descripción entrada - salida en el tiempo.

Cuyo resultado nos conduce a obtener una ecuación en diferencias, donde la salida $y(n)$ es una función de la entrada y algunas constantes relacionadas por operaciones de sumas, productos y retardos.

Ejemplo 4.1:

Representación de sistemas con ecuación en diferencias:

- a) $y(n) = x(n)$, la salida es igual a la entrada $x(n)$
- b) $y(n) = x(n-1)$, la salida es igual a la entrada $x(n)$ retrasada en una unidad
- c) $y(n) = 5x(n+1) + x(n) + x(n-1)$, la salida es una combinación más elaborada de entrada.

En estos ejemplos la relación entrada salida está dada por una ecuación en diferencias, para la representación gráfica de un sistema discreto se tienen varios bloques que se definen a continuación.

4.3 BLOQUES BÁSICOS DE SISTEMAS DISCRETOS

Con base en las operaciones básicas entre señales discretas, se pueden definir los bloques básicos para representar a un sistema discreto. En general un sistema en el tiempo discreto es un algoritmo que de acuerdo a reglas bien definidas opera sobre una señal discreta $x(n)$ llamada excitación o señal de entrada y produce otra señal de salida o respuesta del sistema $y(n)$, estas reglas constituyen los bloques básicos. Teniendo la ecuación en diferencias que representa al sistema, éste se puede trasladar directamente a un diagrama de bloques y viceversa. La combinación de varios bloques básicos da lugar a sistemas discretos que pueden realizar operaciones complicadas sobre las señales.

SUMADOR

Efectúa suma de secuencia de señales, es decir, $y(n) = x_1(n) + x_2(n)$. Es evidente que no se necesita el almacenamiento de las señales de entrada, por lo que se dice que la operación suma no tiene memoria.

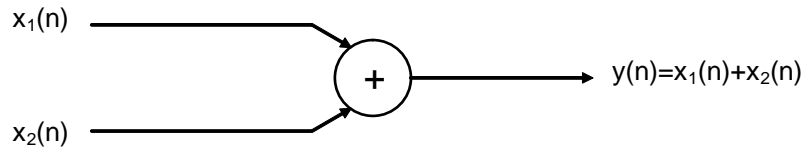


Figura 4.2 Bloque sumador

AMPLIFICADOR O MULTIPLICADOR POR UNA CONSTANTE

Esta operación aplica un factor de escala a una entrada $x(n)$. También es una operación sin memoria.

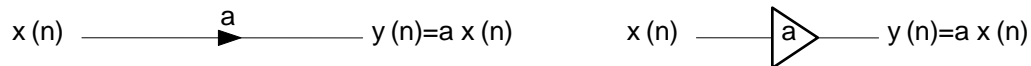


Figura 4.3 Bloque multiplicador por una constante

MULTIPLICACIÓN DE DOS SEÑALES

Las secuencias se multiplican punto a punto y es una operación sin memoria.

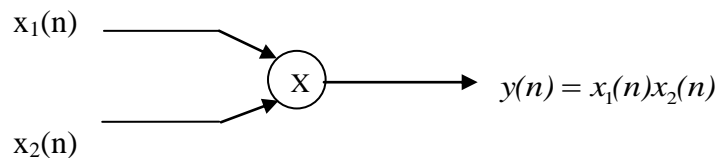


Figura 4.4 Bloque multiplicador entre señales

ADELANTO

Realiza la operación opuesta al bloque de retardo, es decir, que avanza la entrada $x(n)$ en una unidad produciendo una salida, esto significa que a un tiempo “n” se conoce la muestra actual, las anteriores y la siguiente muestra $y(n) = x(n+1)$. Sin embargo, este tipo de bloque es poco usual ya que un adelanto es físicamente irrealizable.

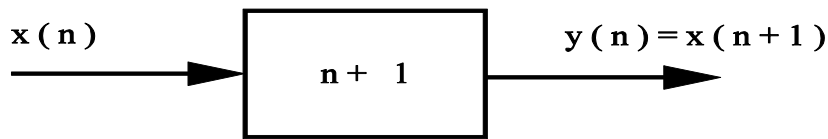


Figura 4.5 Bloque de adelanto

RETARDO

Es una unidad que simplemente retrasa la señal en una muestra. Si la entrada es $x(n)$ entonces la salida es $x(n-1)$, la muestra $x(n-1)$ es almacenada en memoria al tiempo $n-1$, y es requerida o llamada de la memoria al tiempo n , entonces $y(n) = x(n-1)$. Cuando se utiliza la transformada Z, el bloque de retardo se simboliza como z^{-1} .

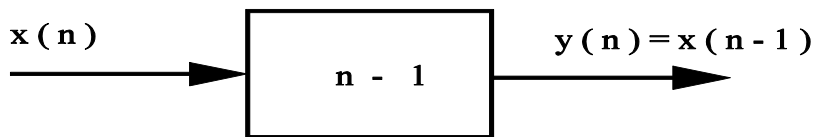


Figura 4.6 Bloque de retardo

A continuación se presentan varios ejemplos de ecuaciones en diferencias de los sistemas y su representación equivalente en diagrama de bloques, este tipo de sistemas se utiliza frecuentemente en el diseño de sistemas.

Ejemplo 4.2

Ecuación en diferencias

Si $y(n) = 2x(n) + x(n-1) - 5x(n-2)$
 $y(0) = 2x(0) + x(-1) - 5x(-2)$
 $y(1) = 2x(1) + x(0) - 5x(-1)$
 $y(2) = 2x(2) + x(1) - 5x(0)$
 $y(3) = 2x(3) + x(2) - 5x(1)$

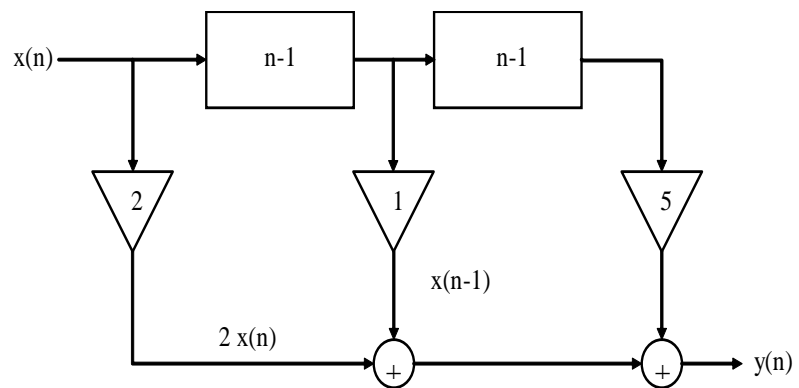


Figura 4.7 Diagrama a bloques del ejemplo 4.2

Ejemplo 4.3

Ecuación en diferencias

$$\begin{aligned}
 y_1(n) &= x(n) + \frac{1}{2}x(n-1) \\
 y_2(n) &= y_1(n) + \frac{1}{4}x(n-2) \\
 &= x(n) + \frac{1}{2}x(n-1) + \frac{1}{4}x(n-2) \\
 y_3(n) &= y_2(n) + \frac{1}{8}x(n-3) \\
 &= x(n) + \frac{1}{2}x(n-1) + \frac{1}{4}x(n-2) + \frac{1}{8}x(n-3) \\
 y(n) &= \sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k x(n-k)
 \end{aligned}$$

Figura 4.8

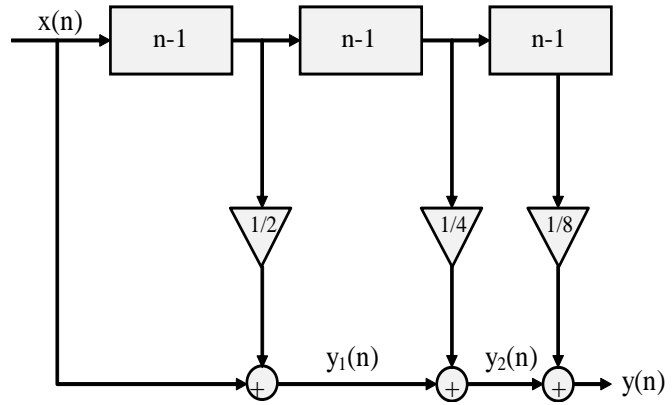


Figura 4.8 Diagrama a bloques del ejemplo 4.3

Ejemplo 4.4

Ecuación en diferencias

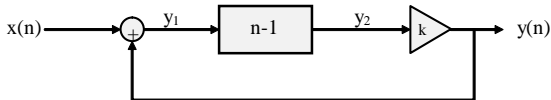


Figura 4.9 Diagrama a bloques del ejemplo 4.4

$$\begin{aligned}
 y_1(n) &= x(n) + y(n) \\
 y_2(n) &= y_1(n-1) = x(n-1) + y(n-1) \\
 y(n) &= ky_2(n) \\
 y(n) &= kx(n-1) + ky(n-1) \\
 y(n-1) &= kx(n-2) + ky(n-2) \\
 y(n) &= kx(n-1) + k^2x(n-2) + k^2y(n-2) \\
 y(n-2) &= kx(n-3) + ky(n-3) \\
 y(n) &= kx(n-1) + k^2x(n-2) + k^3x(n-3) + k^3y(n-3) \\
 y(n) &= \sum_{m=1}^{\infty} k^m x(n-m)
 \end{aligned}$$

Ejemplo 4.5

Ecuación en diferencias

$$y(n] = x(n)h(0) + x(n-1)h(1) + x(n-2)h(2) + \dots + x(n-N+1)h(N-1) = \sum_{i=0}^{N-1} h(i)x(n-i)$$

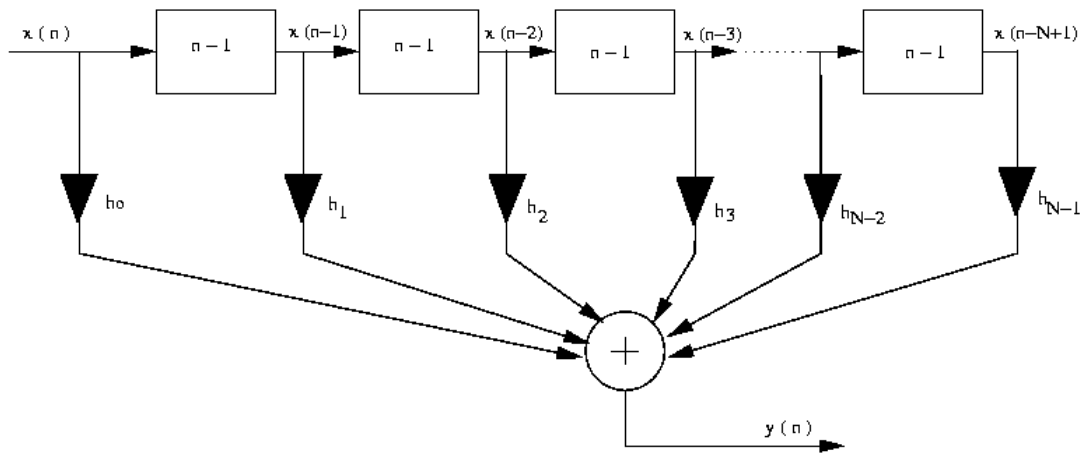


Figura 4.10 Diagrama a bloques del ejemplo 4.5

Ejemplo 4.6

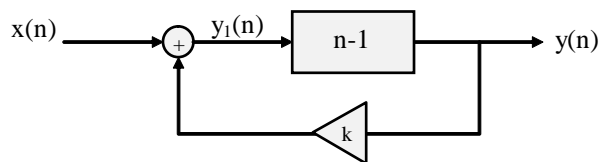


Figura 4.11 Diagrama a bloques del ejemplo 4.6

Ecuación en diferencias

$$y_1(n) = x(n) + ky(n)$$

$$y(n) = y_1(n-1) = x(n-1) + ky(n-1)$$

$$y(n-1) = x(n-2) + ky(n-2) \Rightarrow y(n) = x(n-1) + kx(n-2) + k^2 y(n-2)$$

$$y(n-2) = x(n-3) + ky(n-3) \Rightarrow y(n) = x(n-1) + kx(n-2) + k^2 x(n-3) + k^3 y(n-3)$$

$$y(n) = \sum_{m=1}^{\infty} k^{m-1} x[n-m]$$

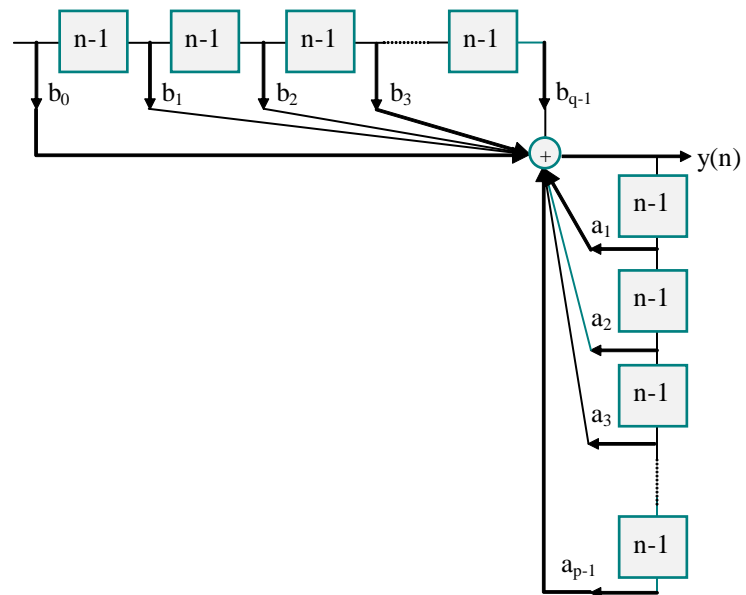
Ejemplo 4.7

Figura 4.12 Diagrama a bloques del ejemplo 4.7

Ecuación en diferencias

$$\sum_{i=0}^{p-1} a_i y(n-i) = \sum_{i=0}^{q-1} b_i y(n-i)$$

ó

$$y(n) = \sum_{i=0}^{q-1} b_i y(n-i) - \sum_{i=1}^{p-1} a_i y(n-i)$$

4.4 CLASIFICACIONES DE LOS SISTEMAS DISCRETOS

Los sistemas discretos se pueden clasificar desde diferentes puntos de vista, entre ellas tenemos: sistemas estáticos o sin memoria y dinámicos o con memoria, lineales y no lineales, invariantes en el tiempo y variantes, causales y no causales. A continuación se describirán estas clasificaciones y sus características.

SISTEMA SIN MEMORIA O ESTÁTICO

La salida de un sistema sin memoria o estático se puede conocer sólo con la muestra n -ésima de la entrada $x(n)$, es decir, que no se requiere de las muestras pasadas ni futuras, ejemplo

$$y(n) = A x(n)$$

SISTEMA DINÁMICO O CON MEMORIA FINITA

Para conocer la muestra presente de salida $y(n)$ se requiere de la muestra presente $x(n)$ y de N muestras anteriores a la entrada, esto es: $y(n) = Ax(n) + Bx(n-N)$. Se dice que este sistema tiene una memoria de duración N .

SISTEMA RECURSIVO

Es un sistema dinámico que para conocer $y(n)$ se requiere de las muestras anteriores de la salida del sistema, ejemplo:

$$y(n) = Cx(n) + Dy(n-N)$$

SISTEMAS LINEALES Y NO LINEALES

Si definimos a $y(n)$ como la señal o secuencia de salida, $x(n)$ la secuencia de entrada y se tiene una función de transformación denotada por T , entonces el sistema puede ser descrito como

$$y(n) = T[x(n)]$$

Un sistema discreto se dice que es lineal si cumple con el principio de *superposición*, el cual se puede enunciar de la siguiente forma: *la salida producida por un sistema que ha sido excitado o alimentado por la combinación lineal de varias entradas en forma simultánea es equivalente a la suma de la combinación lineal de las salidas producidas por el sistema cuando las entradas se aplican en forma individual.*

Para verificar que el modelo matemático de un sistema es lineal, éste debe satisfacer las siguientes dos condiciones: homogeneidad y aditividad

a) Homogeneidad

Sea $f(n)$ una función discreta de entrada a un sistema discreto, entonces la función de transformación T es homogénea si y sólo si

$$T\{Kf(n)\} = KT\{f(n)\} \quad (4.2)$$

b) Aditividad

Considérese la función de transformación T y además las entradas $x_1(n)$ y $x_2(n)$; esta función es aditiva si y sólo si

$$T\{ax_1(n) + bx_2(n)\} = aT\{x_1(n)\} + bT\{x_2(n)\} \quad (4.3)$$

El análisis de los sistemas lineales se basa principalmente en:

- El principio de superposición
- La representación de una señal $x(n)$ por una sumatoria de impulsos unitarios
- Su caracterización está dada por medio de:
 1. Ecuaciones Diferenciales
 2. La Respuesta al impulso
 3. La Función de Transferencia
- Debe cumplir que para una entrada cero produzca una salida cero (sistema no excitado).

Si algunas de las condiciones anteriores no se cumplen, entonces el sistema es no lineal.

SISTEMA INVARIANTE

Un sistema es invariante en el tiempo, si para una excitación de entrada genera la misma respuesta independientemente del tiempo en que se aplique la entrada.

SISTEMAS CAUSALES Y NO CAUSALES

Un sistema físico es causal si éste no reacciona hasta que le es aplicado un estímulo, es decir, en un sistema no se puede esperar el efecto de la transformación sin la existencia previa de una causa. Para un sistema dinámico no recursivo la salida dependerá solamente de muestras pasadas y presentes de la entrada, esto implica que el sistema es causal si $h[n] = 0$ para $n < 0$. Matemáticamente

$$y(n) = f\{x(n), x(n-1), x(n-2), x(n-3), \dots\} \quad (4.4)$$

De no satisfacerse las condiciones anteriores, se dice el sistema es no causal.

Para un sistema recursivo, la salida también dependerá de las muestras pasadas de la salida, y el sistema seguirá siendo causal, es decir

$$y(n) = f\{x(n), x(n-1), x(n-2), x(n-3), \dots, y(n-1), y(n-2), \dots\} \quad (4.5)$$

4.5 SISTEMAS LINEALES E INVARIANTES EN EL TIEMPO DISCRETO (SLITD)

De la clasificación anterior los sistemas SLITD cumplen con la linealidad y la invarianza en el tiempo, las bases principales de su análisis son:

- a) El principio de superposición.

- b) La representación de la señal de entrada $x(n)$ está dada por una sumatoria de impulsos, es decir

$$x(n) = \sum_{k=0}^{\infty} x(k) \delta(n-k) \quad (4.6)$$

Si se define a $h(n)$ como la respuesta al impulso unitario $\delta(n)$, entonces $T\{\delta(n)\} = h(n)$, si el sistema es invariable se tiene que $h(n-k) = T\{\delta(n-k)\}$, esto quiere decir que los sistemas son caracterizados completamente por su respuesta al impulso unitario $h(n)$.

Este último resultado puede observarse tomando en cuenta que la entrada $x(n)$ se puede escribir como una sumatoria de muestras multiplicadas por la función impulso unitario desplazada para cada muestra, esto es, para una entrada $x(n)$ y aplicando el principio de superposición

$$y(n) = T\{x(n) \delta(n)\} = x(n)T\{\delta(n)\} ; T\{\delta(n)\} \triangleq h(n) \quad (4.7)$$

Analizando la respuesta de un SLID a una entrada impulso, se observa en la figura 4.13 que la entrada $\delta(n)$ debe producir una salida $h(n)$

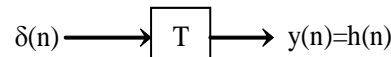


Figura 4.13 Definición de respuesta al impulso

Si se conoce esta respuesta $y(n)$ y se define a $h(n)$ como la respuesta al impulso unitario $\delta(n)$, entonces

$$T\{\delta(n)\} \triangleq h(n) \quad \text{si } h(n) = 0 \text{ para } n < 0 \quad (4.8)$$

La ecuación (4.8) implica que el sistema es causal.

También, por su invariancia en el tiempo: $T\{\delta(n-k)\} \triangleq h(n-k)$

Bajo estas consideraciones se puede encontrar la respuesta de un SLITD a cualquier señal discreta $x(n)$ como se muestra en la figura 4.14

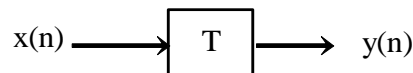


Figura 4.14 Respuesta de un sistema SLITD

Una propiedad de la función $\delta(n)$ es que cualquier secuencia discreta $x(n)$ se puede escribir como una sumatoria de impulsos discretos desplazados y ponderados por los valores de $x(n)$, si $x(n)$ es causal, entonces

$$x(n) = \sum_{k=0}^{\infty} x(k) \delta(n-k) = x(0) \delta(n) + x(1) \delta(n-1) + x(2) \delta(n-2) + \dots \quad (4.9)$$

aplicando superposición y como $y(n)=T\{x(n)\}$, entonces

$$\begin{aligned} y(n) &= T\{x(n)\} = T\{x(0) \delta(n)\} + T\{x(1) \delta(n-1)\} + T\{x(2) \delta(n-2)\} + \dots \\ &= x(0) T\{\delta(n)\} + x(1) T\{\delta(n-1)\} + x(2) T\{\delta(n-2)\} + \dots \\ &= x(0) h(n) + x(1) h(n-1) + x(2) h(n-2) + \dots \\ y(n) &= \sum_{k=0}^{\infty} x(k) T\{\delta(n-k)\} \\ y(n) &= \sum_{k=0}^{\infty} x(k) h(n-k) = x(n) * h(n) = h(n) * x(n) \end{aligned} \quad (4.10)$$

La ecuación (4.10) se define como la suma de convolución, donde el símbolo * representa al operador de convolución.

4.6 CONVOLUCIÓN

La operación convolución entre señales discretas está compuesta de suma de productos y se le puede considerar *la operación básica o fundamental de PDS*, esta operación se encuentra en muchos algoritmos de PDS, además los procesadores de señales digitales (DSP) se diseñan con el propósito de realizar convoluciones eficientemente en tiempo real [ESC00]. Si tenemos una secuencia de entrada a un sistema lineal $x(n) = \sum x(n) \delta(n-k)$, entonces la salida se obtiene de

$$y(n) = T[x(n)] = \sum_{k=0}^{\infty} x(k) T[\delta(n-k)] = \sum_{k=0}^{\infty} x(k) h(n-k) = h(n) * x(n) \quad (4.11)$$

La transformación queda determinada si conocemos la respuesta al impulso del sistema. Esta sumatoria puede transformarse en la realización de un algoritmo para un sistema digital.

DESARROLLO DE LA CONVOLUCIÓN

Se puede realizar a través de los siguientes pasos:

Primero se varía el índice "i" de la sumatoria y después el tiempo "n", es decir, desarrollando la sumatoria

$$y(n) = \sum_{i=0}^{\infty} x(i)h(n-i)$$

$$y(n) = x(0)h(n) + x(1)h(n-1) + x(2)h(n-2) + x(3)h(n-3) + \dots]x(i)h(n-i) \quad (4.12)$$

Para diferentes valores de $n \geq 0$

$$y(n) = x(0)h(n) + x(1)h(n-1) + x(2)h(n-2) + x(3)h(n-3) + \dots + x(i)h(n-i)$$

$$y(0) = x(0)h(0) + x(1)h(-1) + x(2)h(-2)$$

$$y(1) = x(0)h(1) + x(1)h(0) + x(2)h(-1)$$

$$y(2) = x(0)h(2) + x(1)h(1) + x(2)h(0) + x(3)h(-1)$$

$$y(3) = x(0)h(3) + x(1)h(2) + x(2)h(1) + x(3)h(0)$$

Se ha calculado la salida $y(n)$ para cualquier tiempo $n \geq 0$ con base a la entrada $x(n)$ y la respuesta al impulso $h(n)$, como se observa el desarrollo de la operación convolución nos lleva a un conjunto de sumas de productos para cualquier valor de $y(n)$, a la vez existe un desplazamiento de la secuencia $h(n)$. Para efectos de hacer entendible este proceso en la figura 4.15 se muestra un procedimiento gráfico y otro matricial utilizando dos secuencias finitas y causales.

PASOS PARA EFECTUAR LA CONVOLUCIÓN

La realización de la convolución entre dos secuencias discretas $x(n)$ y $h(n)$ se puede resumir en los siguientes pasos:

1. Se hace girar $h(n)$ sobre el eje $n = 0$ lo que permite obtener $h(-n)$, es decir, efectuar la operación folding.
2. Se corre $h(-n)$ k unidades a la derecha, entonces se obtiene $h(n-k)$.
3. Se multiplican las muestras $x(k)$ por $h(n-k)$.
4. Se hace la suma de los productos del inciso anterior.

Ejemplo 4.8

Dadas dos secuencias discretas $x(n)$ y $h(n)$:

$$x(n) = \{ \begin{array}{cccc} 0.25 & 0.5 & 1 & 0.25 \end{array} \}$$

$$\begin{array}{cccc} x_0 & x_1 & x_2 & x_3 \end{array}$$

$$h(n) = \{ \begin{array}{cccc} 1 & 1 & 0.5 & 0.5 \end{array} \}$$

$$\begin{array}{cccc} h_0 & h_1 & h_2 & h_3 \end{array}$$

Se calcula $y(n) = x(n) * h(n)$ multiplicando los términos solapados de $x(n)$ y $h(-n)$ efectuando la suma de productos.

$h(-n)$ secuencia móvil $x(n)$ secuencia fija

h_3	h_2	h_1	h_0	x_0	x_1	x_2	x_3	
					0.25	0.5	1	0.25
$h(-k)$	0.5	0.5	1	1				$y(0) = 1(0.25) = 0.25$
		0.5	0.5	1	1			$y(1) = 0.5 + 0.25 = 0.75$
			0.5	0.5	0.5	1	1	$y(2) = 1 + 0.5 + (0.5)(0.25) = 1.625$
				0.5	0.5	1	1	$y(3) = 1 + 0.25 + (0.5)^2 + (0.25)(0.5) = 1.625$
					0.5	0.5	1	$y(4) = 1$
						0.5	0.5	$y(5) = 0.5 + (0.5)(0.25) = 0.625$
							0.5	$y(6) = 1/8$

Observando el proceso anterior, se puede construir un arreglo matricial

$$y(n) = \begin{bmatrix} 0.25 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.25 & 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0.25 & 0 & 0 & 0 \\ 0.25 & 1 & 0.5 & 0.25 & 1 & 1 \\ 0 & 0.25 & 1 & 0.5 & 0.25 & 0 \\ 0 & 0 & 0.25 & 1 & 0.5 & 0.25 \\ 0 & 0 & 0 & 0.25 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 0.25 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.75 \\ 1.625 \\ 1.625 \\ 1 \\ 0.625 \\ 0.125 \\ 0 \end{bmatrix}$$

INTERPRETACIÓN GRÁFICA

En la figura 4.15 se observa que la secuencia $x(n)$ está fija, y a $h(n)$ se le aplica la operación folding, a $h(n)$ se le aplican desplazamientos a la derecha y por cada desplazamiento l se efectúa la suma de productos entre $x(n)$ y $h(n-l)$ para obtener un valor de $y(n)$

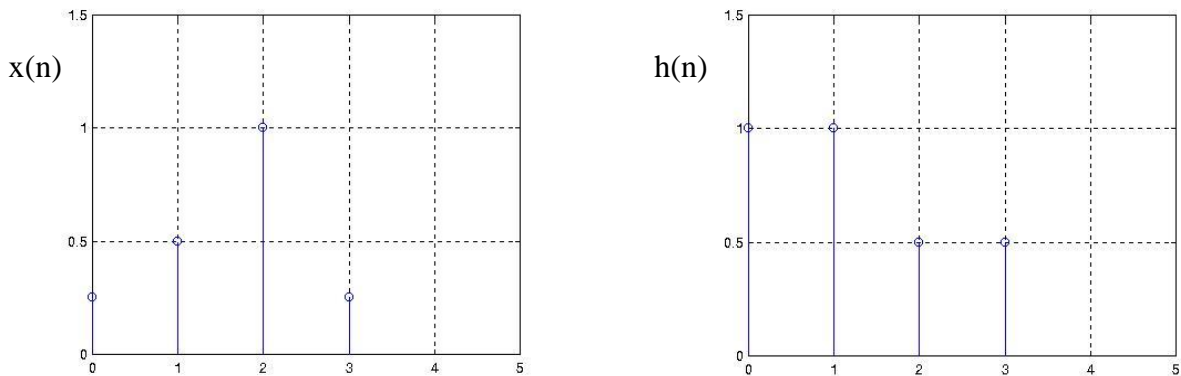
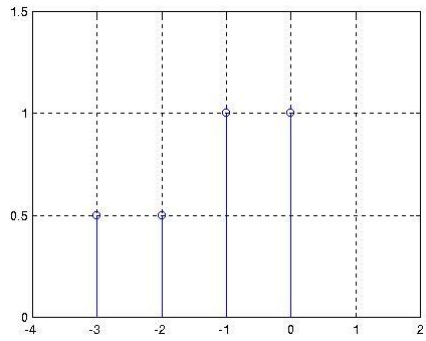
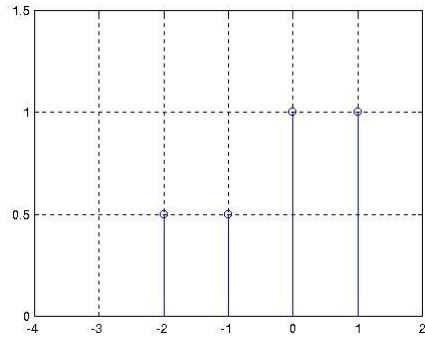


Figura 4.15 Proceso de convolución entre $x(n)$ y $h(n)$

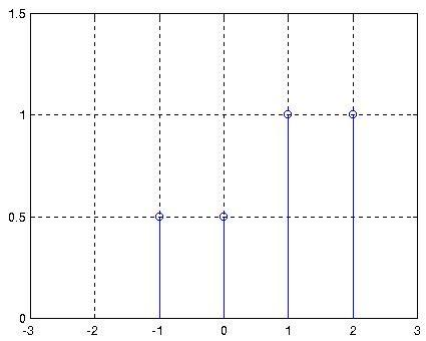
Figura 4.15 Proceso de convolución (continuación)



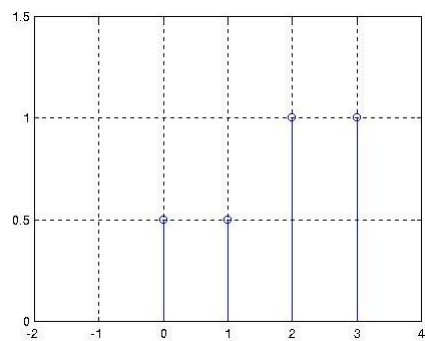
$h(-n)$



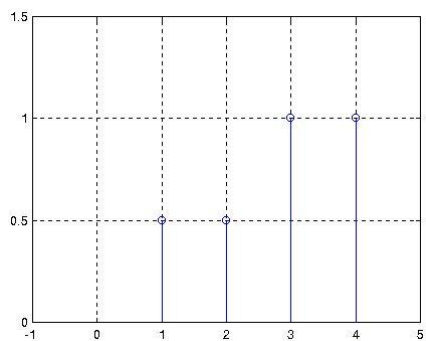
$h(-n-1)$



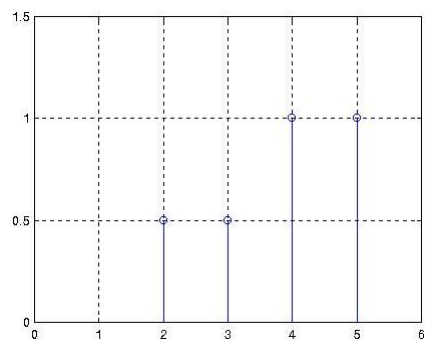
$h(-n-2)$



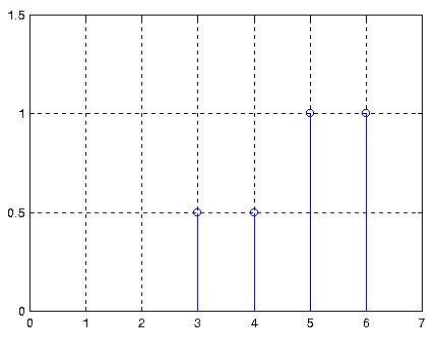
$h(-n-3)$



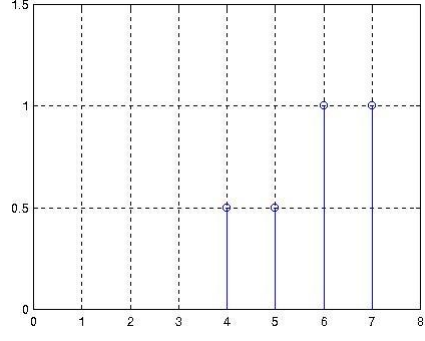
$h(-n-4)$



$h(-n-5)$



$h(-n-6)$



$h(-n-7)$

LA CONVOLUCIÓN UTILIZANDO DOS REGLETAS

El proceso de la suma de convolución también puede ser visualizado como dos reglas con valores de las secuencias, una regla fija y una regla móvil. Específicamente, sobre la escala fija se escriben los números $h(n)$ vs n , mientras que en la escala deslizable se muestran los valores $x(n)$ vs n . Para determinar el valor $y(n)$, simplemente se desliza la escala móvil en “ n ” unidades a la derecha hasta que $x(0)$ quede sobre $h(0)$. Después se multiplican los elementos solapados entre las dos escalas se realiza la suma de dichos productos (véase la figura 4.16). El procedimiento continúa hasta que no existen valores solapados entre la secuencia fija y móvil.

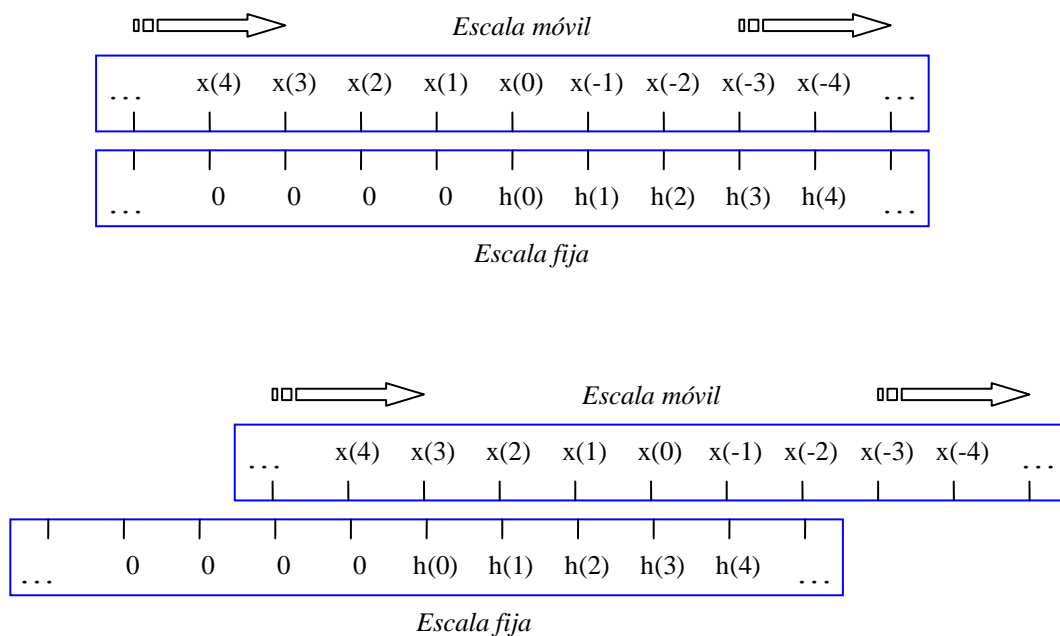


Figura 4.16 Interpretación de la convolución con dos reglas

4.7 INTERCONEXIÓN DE SISTEMAS

Con la operación convolución se pueden obtener sistemas que pueden realizar aplicaciones de interés, además los sistemas se pueden interconectar en dos formas básicas, en cascada y en paralelo, cada una de estas dos configuraciones tiene una respuesta al impulso total.

SISTEMAS ACOPLADOS EN CASCADA

En la figura 4.17 se muestran dos sistemas $h_1(n)$ y $h_2(n)$ acoplados en cascada.

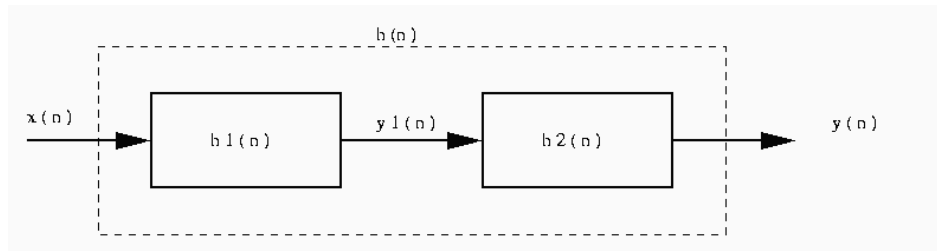


Figura 4.17 Sistemas acoplados en cascada

Resolviendo por partes

$$\begin{aligned} y_1(n) &= T\{x(n)\} = h_1(n) * x(n) \\ y(n) &= T\{y_1(n)\} = h_2(n) * y_1(n) \end{aligned}$$

entonces

$$\begin{aligned} y(n) &= h_2(n) * (h_1(n) * x(n)) \\ &= (h_2(n) * h_1(n)) * x(n) \end{aligned} \quad (4.13)$$

De la ecuación (4.13) se comprueba que la respuesta al impulso del sistema completo es la convolución de las respuestas al impulso individuales $h_i(n)$, es decir

$$h(n) = h_1(n) * h_2(n) = h_2(n) * h_1(n) \quad (4.14)$$

La longitud de $h(n)$ es igual a la suma de las longitudes de $h_1(n)$ y $h_2(n)$ menos 1, es decir

$$L_h = L_{h_1} + L_{h_2} - 1 \quad (4.15)$$

SISTEMAS ACOPLADOS EN PARALELO

En la figura 4.18 se tiene los sistemas $h_1(n)$ y $h_2(n)$ conectados en paralelo, con entrada común $x(n)$

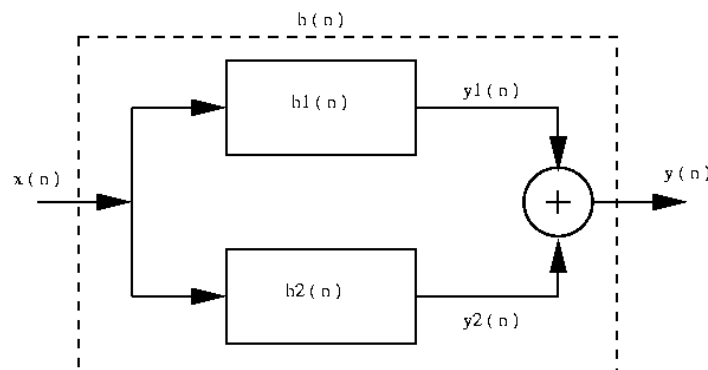


Figura 4.18 Sistemas acoplados en paralelo

entonces

$$y(n) = y_1(n) + y_2(n)$$

donde

$$\begin{aligned} y_1(n) &= h_1(n) * x(n) \\ y_2(n) &= h_2(n) * x(n) \end{aligned}$$

entonces

$$\begin{aligned} y(n) &= h_1(n) * x(n) + h_2(n) * x(n) \\ y(n) &= (h_1(n) + h_2(n)) * x(n) \end{aligned} \quad (4.16)$$

por tanto la respuesta al impulso total del sistema es

$$h(n) = h_1(n) + h_2(n) \quad (4.17)$$

por la propiedad de distributividad de la convolución, la respuesta al impulso del sistema es la suma de las respuestas al impulso individuales como lo comprueba la ecuación (4.17) y la longitud de $h(n)$ es la máxima longitud de las respuestas al impulso $h_i(n)$.

Una tercera posibilidad de acoplamiento de sistemas sería la combinación de sistemas en cascada y paralelo en cualquier orden y cantidad de respuestas al impulso $h_i(n)$, sin embargo sabiendo analizar los sistemas en cascada y en paralelo, es fácil calcular la respuesta al impulso, la salida y la longitud de la respuesta al impulso.

4.8 CORRELACIÓN

La correlación es una operación matemática que se parece a la convolución, el objetivo de la correlación entre dos señales es medir que tanto se parecen entre sí para extraer alguna información, calcular retardos entre señales, verificar y calcular la periodicidad de una señal, calcular la energía de una señal. La correlación entre señales se encuentra aplicada frecuentemente en sistemas como radar, sonar, comunicaciones digitales, síntesis de voz, reconocimiento de voz, procesamiento de imágenes, geología y otras áreas de la ingeniería.

Si $x(n)$ y $y(n)$ son dos señales de energía finita entonces la correlación entre $x(n)$ y $y(n)$ se define como

$$r_{xy}(\ell) = \sum_{n=-\infty}^{\infty} x(n) y(n-\ell) \quad ; \quad \ell = 0, \pm 1, \pm 2, \dots \quad (4.18)$$

$$r_{xy}(\ell) = \sum_{n=-\infty}^{\infty} x(n+\ell) y(n) \quad ; \quad \ell = 0, \pm 1, \pm 2, \dots \quad (4.19)$$

donde ℓ es el índice de corrimiento en el tiempo, el subíndice xy indica que “ x ” está fijo, y “ y ” se desplaza en ℓ unidades. De forma similar se puede escribir $r_{yx}(\ell)$ como en la ecuación (4.20)

$$r_{yx}(\ell) = \sum_{n=-\infty}^{\infty} y(n) x(n-\ell) \quad ; \quad \ell = 0, \pm 1, \pm 2, \dots \quad (4.20)$$

$$r_{yx}(\ell) = \sum_{n=-\infty}^{\infty} y(n+\ell) x(n) \quad ; \quad \ell = 0, \pm 1, \pm 2, \dots \quad (4.21)$$

Comparando las ecuaciones (4.18) con (4.22) y (4.19) con (4.20) se observa que $r_{xy}(\ell) = r_{yx}(-\ell)$, esto significa que $r_{yx}(\ell)$ es un doblamiento de $r_{xy}(\ell) = r_{yx}(-\ell)$.

Entonces la correlación de $r_{xy}(\ell)$ se puede escribir como la convolución:

$$r_{xy}(\ell) = x(\ell) * y(-\ell) \quad (4.22)$$

DESARROLLO DE LA CORRELACIÓN

Las operaciones necesarias para el cálculo de correlación se desarrollan de la ecuación (4.18)

$$\begin{aligned} r_{xy}(\ell) &= \sum_{n=0}^{\infty} x(n)y(n-\ell) \\ r_{xy}(\ell) &= x(0)y(-\ell) + x(1)y(1-\ell) + x(2)y(2-\ell) + x(3)y(3-\ell) + \dots \\ r_{xy}(0) &= x(0)y(0) + x(1)y(1) + x(2)y(2) + x(3)y(3) + \dots \\ r_{xy}(1) &= x(0)y(-1) + x(1)y(0) + x(2)y(1) + x(3)y(2) + \dots \\ r_{xy}(2) &= x(0)y(-2) + x(1)y(-1) + x(2)y(0) + x(3)y(1) + \dots \\ r_{xy}(-1) &= x(0)y(1) + x(1)y(2) + x(2)y(3) + x(3)y(4) + \dots \\ r_{xy}(-2) &= x(0)y(2) + x(1)y(3) + x(2)y(4) + x(3)y(5) + \dots \\ r_{xy}(-3) &= x(0)y(3) + x(1)y(4) + x(2)y(5) + x(3)y(6) + \dots \end{aligned}$$

De forma similar si se desarrolla la ecuación (4.19)

$$\begin{aligned} r_{xy}(\ell) &= \sum_{n=-\infty}^{\infty} x(n+\ell) y(n) \quad ; \quad \ell = 0, \pm 1, \pm 2, \dots \\ r_{xy}(\ell) &= x(\ell)y(0) + x(1+\ell)y(1) + x(2+\ell)y(2) + x(3+\ell)y(3) \\ r_{xy}(0) &= x(0)y(0) + x(1)y(1) + x(2)y(2) + x(3)y(3) \\ r_{xy}(1) &= x(1)y(0) + x(2)y(1) + x(3)y(2) + x(4)y(3) \\ r_{xy}(2) &= x(2)y(0) + x(3)y(1) + x(4)y(2) + x(5)y(3) \\ r_{xy}(-1) &= x(-1)y(0) + x(0)y(1) + x(1)y(2) + x(2)y(3) \\ r_{xy}(-2) &= x(-2)y(0) + x(-1)y(1) + x(0)y(2) + x(1)y(3) \\ r_{xy}(-3) &= x(-3)y(0) + x(-2)y(1) + x(-1)y(2) + x(0)y(3) \end{aligned}$$

Como se observa a nivel de operaciones matemáticas es muy similar a la convolución, y que el resultado de la ecuación (4.18) y (4.19) son exactamente iguales si se desarrollan para todo n y l desde $-\infty$ a ∞ .

PROPIEDADES DE LA CORRELACIÓN

1. La convolución de $x(n)$ con $y(-n)$ se puede calcular como una correlación (ecuación 4.22)

$$r_{xy}(\ell) = x(\ell) * y(-\ell)$$

2. Para secuencias causales y de duración finita “N” la correlación y la autocorrelación se calculan

$$r_{xy}(\ell) = \sum_{n=i}^{N-|k|-1} x(n) y(n-\ell) \quad (4.23)$$

$$i = \ell, \quad k = 0 \quad \text{para } \ell \geq 0$$

$$i = 0, \quad k = \ell \quad \text{para } \ell < 0$$

3. Para secuencias periódicas con período N se tiene:

$$r_{xy}(\ell) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) y(n-\ell) \quad (4.24)$$

4. Reflexión $r_{xy}(\ell) = r_{yx}(-\ell)$ (4.25)

5. Correlación normalizada

$$r_{nxy}(\ell) = \frac{r_{xy}(\ell)}{\sqrt{r_{xx}(0)r_{yy}(0)}} \quad (4.26)$$

En la práctica se utiliza para escalamiento. Donde $r_{xx}(0)$ y $r_{yy}(0)$ son las autocorrelaciones de $x(n)$ y $y(n)$ evaluadas en cero respectivamente.

6. Si $y(n) = x(n) + w(n)$, donde $w(n)$ es una señal de ruido y si el cálculo se realiza en M muestras de $y(n)$, $0 \leq n \leq M-1$, y $x(n)$ es una señal periódica de período N , y $M \gg N$

$$r_{yy}(\ell) = \frac{1}{M} \sum_{n=0}^{M-1} [x(n) + w(n)][x(n-\ell) + w(n-\ell)]$$

$$r_{yy}(\ell) = r_{xx}(\ell) + r_{xw}(\ell) + r_{wx}(\ell) + r_{ww}(\ell) \quad (4.27)$$

donde

$$r_{xx}(\ell) \gg r_{ww}(\ell), \quad r_{xx}(\ell) \gg r_{wx}(\ell) \quad \text{y} \quad r_{xx}(\ell) \gg r_{xw}(\ell)$$

Este resultado permite detectar la presencia de la señal periódica $x(n)$ entre el ruido $w(n)$ e identificar el período.

7. Para un sistema con entrada $x(n)$ y respuesta al impulso $h(n)$ y salida $y(n)$, donde

$$y(n) = \sum_{k=-\infty}^{\infty} h(n)x(n-k)$$

La correlación entre la salida y la entrada es

$$\begin{aligned} r_{xy}(\ell) &= y(\ell) * x(-\ell) = [h(\ell) * x(\ell)] * x(-\ell) \\ &= h(\ell) * [x(\ell) * x(-\ell)] = h(\ell) * r_{xx}(\ell) \end{aligned}$$

Se obtiene

$$r_{xy}(\ell) = h(-\ell) * r_{xx}(\ell) \quad (4.28)$$

Entonces la autocorrelación de la salida se puede escribir como

$$\begin{aligned} r_{yy}(\ell) &= y(\ell) * y(-\ell) \\ &= [h(\ell) * x(\ell)] * [h(-\ell) * x(-\ell)] \\ &= [h(\ell) * h(-\ell)] * [x(\ell) * x(-\ell)] \\ r_{yy}(\ell) &= r_{hh}(\ell) * r_{xx}(\ell) = r_{xy}(\ell) * h(\ell) \end{aligned} \quad (4.29)$$

La autocorrelación $r_{hh}(l)$ de la respuesta al impulso existe si el sistema es estable [PRO92].

8. La energía de la salida de la señal en términos de autocorrelaciones se calcula para $l=0$

$$r_{yy}(0) = \sum_{k=-\infty}^{\infty} r_{hh}(k)r_{xx}(k) \quad (4.30)$$

AUTO CORRELACIÓN

Si $y(n) = x(n)$, entonces se tiene la autocorrelación de $x(n)$ definida como

$$r_{xx}(\ell) = \sum_{n=-\infty}^{\infty} x(n) x(n-\ell) = \sum_{n=-\infty}^{\infty} x(n+\ell) x(n) \quad (4.31)$$

PROPIEDADES DE LA AUTOCORRELACIÓN

1) Simetría $r_{xx}(\ell) = r_{xx}(-\ell)$ (4.32)

2) Energía de la señal

$$r_{xx}(0) = Ex = \sum_{n=-\infty}^{\infty} x(n)x^*(n) = \sum_{n=-\infty}^{\infty} |x(n)|^2 \quad (4.33)$$

3) Máximos de $r_{xx}(\ell)$ $r_{xx}(\ell) \leq r_{xx}(0)$ (4.34)

4) Para secuencias aleatorias de longitud N se tienen las autocorrelaciones

Estimador sin sesgo $r_{xx}[\ell] = \frac{1}{N} \sum_{n=0}^{N-|\ell|-1} x^*(n) x(n-\ell)$ (4.35)

Estimador con sesgo $r_{xx}[\ell] = \frac{1}{N-|\ell|} \sum_{n=0}^{N-|\ell|-1} x^*(n) x(n-\ell)$ (4.36)

5) Autocorrelación normalizada

$$r_{n,xx}(\ell) = \frac{r_{xx}(\ell)}{r_{xx}(0)} \leq 1 \quad (4.37)$$

6) El espectro de $X(\omega)$ es la Transformada de Fourier de $r_{xx}(\ell)$

$$r_{xx}(\ell) \xrightarrow{TF} \mathcal{R}_{xx}(\omega) \quad (4.38)$$

7) Autocorrelación calculada por convolución

$$r_{xx}(\ell) = x(\ell) * x(-\ell) \quad (4.39)$$

4.9 RESUMEN

En este capítulo se han definido los sistemas discretos, sus formas de representarlos y se han tratado algunas características y propiedades más importantes; con todas estas herramientas hemos llegado a definir las operaciones convolución y correlación que son de mucha utilidad y aplicación en el PDS. Con los conceptos vistos en este capítulo se está en la posibilidad de analizar pequeños sistemas discretos que realicen aplicaciones básicas.

4.10 PROBLEMAS PROPUESTOS

1. Dadas dos secuencias discretas $h(n) = \{1.235, -2.456, 7.999, 3.55, -0.9996, 7.523\}$ y $x(n) = \{5.256, 7.345, -9.975, -2.345, 12.687, 8.68, -3.44, 6.88\}$ calcular la convolución.
2. Dado un sistema SLITD con respuesta al impulso $h(n) = a^n U(n)$ y entrada $x(n) = U(n)$, calcular la salida $y(n)$.
3. Se tienen dos subsistemas acoplados en cascada $h_1(n) = (1/a)^n U(n)$ y $h_2(n) = (1/b)^n U(n)$ calcular la respuesta al impulso $h(n)$ del sistema total.
4. Para un sistema con dos subsistemas en cascada $h_1(n)$ y $h_2(n)$ calcular $r_{xy}(l)$ y $r_{yy}(l)$ en términos de correlaciones y convoluciones.
5. Para un sistema con tres subsistemas en paralelo $h_1(n)$, $h_2(n)$ y $h_3(n)$ calcular $r_{xy}(l)$ y $r_{yy}(l)$ en términos de correlaciones y convoluciones.

CAPÍTULO 5

TRANSFORMADA Z

Durante muchos años los matemáticos han utilizado métodos de transformaciones para simplificar el análisis y síntesis de sistemas lineales cuya dinámica es gobernada por ecuaciones diferenciales o en diferencias. Una transformada es el traslado de una función de un dominio a otro dominio, en el caso concreto de la transformada Z (TZ) realiza una transformación de señales y sistemas discretos en el dominio n al dominio z , donde z pertenece a los complejos, posteriormente las funciones resultantes en z se analizan el plano complejo z , en este dominio se pueden observar algunas características del sistema o la señal que son difíciles de analizar en el dominio del tiempo discreto n . Además como se mencionó en el capítulo cuatro, otra forma de caracterizar un sistema es a través de su función de transferencia, que para sistemas discretos la definiremos como $H(z)$.

En este capítulo se presenta la definición de TZ, su relación con la transformada de Laplace, la relación de la variable z con la frecuencia angular ω , algunas de las propiedades de la TZ, la estabilidad de los sistemas y algunos ejemplos de aplicación.

5.1 RESPUESTA DE UN SLITD A UNA EXPONENCIAL COMPLEJA

Suponiendo un sistema lineal e invariante en el tiempo discreto (SLITD) con la entrada $x(n) = z^n$ y si definimos $z=e^{j\omega}$, entonces la salida $y(n)$ puede calcularse como la convolución entre la entrada y la respuesta al impulso $h(n)$ del sistema dado por la ecuación (5.1)

$$y(n) = x(n) * h(n) = h(n) * x(n) = \sum_{k=-\infty}^{\infty} h(k) x(n-k) \quad (5.1)$$

sustituyendo $x(n) = z^n$ en ecuación (5.1), se obtiene

$$y(n) = \sum_{k=-\infty}^{\infty} h(k) z^{n-k} = z^n \sum_{k=-\infty}^{\infty} h(k) z^{-k} \quad (5.2)$$

de la ecuación (5.2) se define

$$H(z) = \sum_{k=-\infty}^{\infty} h(k) z^{-k} \quad (5.3)$$

entonces $y(n)$ también se puede escribir como $y(n) = H(z) z^n$.

Se observa que cuando la entrada a un sistema SLITD es una exponencial compleja, la salida es la misma exponencial multiplicada por $H(z)$, donde $H(z)$ se define como la *función de transferencia del sistema* y “ z ” es un *valor propio* asociado a la función propia z^n del sistema. Es decir, que entre $h(n)$ y $H(z)$ existe una correspondencia designada por la notación:

TZ

$$h(n) \leftrightarrow H(z)$$

esta correspondencia es válida para cualquier secuencia discreta $x(n)$, y a $X(z)$ se le conoce como la TZ de la secuencia $x(n)$.

RESPUESTA DE UN SLITD A UN CONJUNTO DE EXPONENCIALES COMPLEJAS

De manera similar se puede calcular la salida $y(n)$ de un SLITD a una entrada $x(n)$ representada como una combinación lineal de exponenciales complejas dada por la ecuación (5.4)

$$x(n) = \sum_k C_k z_k^n \quad (5.4)$$

la salida $y(n)$ del SLITD estará dada por

$$y(n) = \sum_k C_k H(z) z_k^n \quad (5.5)$$

Es decir, que la salida puede representarse por una combinación lineal de estas mismas exponenciales complejas, donde cada coeficiente se obtiene multiplicando los coeficientes de la entrada C_k por los valores propios del sistema $H(z)$, asociados a las funciones propias z_k^n , por lo que se puede concluir que:

- a) $H(z)$ es la TZ de $h(n)$
- b) Si $x(n) = z^n$, entonces $H(z)$ contiene el conjunto de coeficientes de la respuesta resultante $y(n) = H(z)z^n$
- c) Por definición y por el teorema de convolución $H(z) = Y(z)/X(z)$

Donde la respuesta al impulso $h(n)$ y la función de transferencia $H(z)$ son las formas más utilizadas para caracterizar sistemas SLITD.

5.2 DEFINICIÓN DE TRANSFORMADA Z

La TZ es una poderosa herramienta para el estudio y análisis de señales y sistemas discretos. En sistemas discretos, se puede decir que la TZ se comporta de manera análoga a la transformada de Laplace (TL) para las señales continuas. La TZ permite trabajar ecuaciones en diferencias como ecuaciones algebraicas y analizar a las funciones de transferencia de sistemas discretos como diagramas de polos y ceros en el plano complejo Z, de esta manera se puede interpretar a un sistema discreto en términos de su estabilidad y respuesta en frecuencia de acuerdo a un diagrama de polos y ceros.

La TZ para una secuencia $x(n)$ está definida por la ecuación (5.6)

$$\text{TZ} \{x(n)\} = X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \quad (5.6)$$

En la ecuación (5.6), $X(z)$ se denomina Transformada Z directa y es una función de la variable compleja z que transforma una secuencia $x(n)$ en dominio del tiempo discreto a una función $X(z)$ en el dominio de z , donde $z = re^{j\omega}$. Para sistemas causales se define

$$\text{TZ}\{x(n)\} = X(z) = \sum_{n=0}^{\infty} x(n) z^{-n} \quad (5.7)$$

De la definición de la TZ, ecuaciones (5.6) y (5.7), se observa que es en esencia la sumatoria de los términos de una serie infinita, donde los coeficientes de la serie están en correspondencia uno a uno con las muestras de la secuencia. Por tanto una secuencia $x(n)$ y su TZ constituyen un par único, sustituyendo $z = re^{j\omega}$ en la ecuación (5.6) se obtiene

$$X(re^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)(re^{j\omega})^{-n} = \sum_{n=-\infty}^{\infty} x(n)(r^{-n})(e^{-j\omega n}) \quad (5.8)$$

la ecuación (5.8) puede interpretarse como la transformada de Fourier en el tiempo discreto (DTFT) de la secuencia original $x(n)$ multiplicada por la secuencia r^{-n} , cuando $r=1$ la ecuación (5.8) se reduce a la transformada en el tiempo discreto de Fourier. La convergencia de la serie, depende de los valores adoptados por z , una condición de convergencia de la serie está dada por el criterio de la ecuación (5.9)

$$\sum_{n=-\infty}^{\infty} |x(n)r^{-n}| < \infty \quad (5.9)$$

5.3 DEDUCCIÓN DE LA TZ POR TRANSFORMADA DE LAPLACE

Otra forma de obtener la TZ es muestreando una señal $x(t)$, es decir, tomar valores de $x(nT)$ que ponderan a la delta de Dirac desplazada, es decir

$$x(n) = \sum_{n=-\infty}^{\infty} x(nT) \delta(t - nT) \quad (5.10)$$

La TL de una función continua $x(t)$ se define

$$X(s) = \int_0^{\infty} x(t) e^{-st} dt \quad (5.11)$$

aplicando la TL a $x(n)$ y propiedades de la función delta

$$X(s) = \int_0^{\infty} x(t) e^{-st} dt = \int_0^{\infty} \sum_{n=0}^{\infty} x(nT) \delta(t - nT) e^{-st} dt \quad (5.12)$$

y utilizando propiedades de la función $\delta(t)$ obtenemos

$$X(s) = \sum_{n=0}^{\infty} x(nT) \int_0^{\infty} \delta(t - nT) e^{-st} dt \quad (5.13)$$

como $s = j\omega$ podemos definir la relación entre las variables complejas s y z

$$z = e^{sT} \quad \text{ó} \quad s = \frac{\ln(z)}{T} \quad (5.14)$$

Si $T=1$ y $z = e^{j\omega} = e^s$, entonces se obtiene la transformada Z para una secuencia causal $x(n)$

$$X(z) = \sum_{n=0}^{\infty} x(nT) z^{-n} \quad (5.15)$$

lo que comprueba la relación entre TZ a partir de la TL.

Por lo tanto se concluye que la Transformada de Laplace de una señal muestreada es igual a la TZ de la señal.

Se observa que $X(s)$ es una función periódica con período $2\pi/T = \omega$:

$$X(s + (j2\pi/T)) = \sum_{n=-\infty}^{\infty} x(nT) e^{-nT(s + j2\pi/T)} = \sum_{n=-\infty}^{\infty} x(nT) e^{-nTs} e^{-jn2\pi} = X(s) \quad (5.16)$$

de la ecuación (5.13) se verifica que la TZ es periódica.

De la definición de la TZ se observa que en la secuencia los coeficientes de la serie están en correspondencia uno a uno con las muestras de la secuencia, por lo que una secuencia y su TZ constituyen un par único, esto asegura la unicidad de la TZ, es decir

$$x(n) = \sum_{n=-\infty}^{\infty} x(nT) \delta(t - nT) \quad \text{y} \quad X(s) = \sum_{s=-\infty}^{\infty} x(nT) e^{-nTs} \quad (5.17)$$

5.4 REGIÓN DE CONVERGENCIA DE LA TZ (ROC)

La definición de la TZ como una sumatoria infinita la hace impráctica en su manejo, por lo que es necesario obtener expresiones simplificadas (cerrada), esto se logra debido a que converge para algunos valores de z . La magnitud de $X(z)$ para un valor complejo de z , puede ser finita o infinita. Para todos aquellos valores de z en los que $X(z)$ es finita es llamada región de convergencia (ROC) de $X(z)$, mientras que el conjunto de valores de z para los que $X(z)$ es infinita se le llama región de divergencia de $X(z)$. El problema de hallar la región de convergencia ROC de $x(z)$ es equivalente a determinar el intervalo de valores de " r " para el cual la secuencia $x(n)r^{-n}$ es absolutamente sumable. La forma cerrada de la TZ no especifica una señal única en el dominio del tiempo, esta ambigüedad se resuelve si además de la TZ de la señal o sistema se especifica la ROC. Es decir, que una señal discreta $x(n)$ queda unívocamente determinada por su TZ $X(z)$ y la región de convergencia de $X(z)$ [PRO92].

Para conocer la región de convergencia de una TZ es importante observar la naturaleza de la secuencia $x(n)$, es decir, reducir la serie a su forma simplificada para obtener la región de transformación para valores de z .

CRITERIOS DE CONVERGENCIA

1. Si algún punto $z = z_1$ en el plano está en la región de convergencia, entonces todos los valores de z que se encuentran en la circunferencia definida por $|z| = |z_1|$, también se encuentran en la región de convergencia.
2. La Transformada de Fourier en el tiempo discreto de $x(n)$ converge absolutamente si y sólo si la región de convergencia de su TZ incluye al círculo unitario. La transformada de Fourier para el tiempo discreto es igual a la TZ cuando $z = e^{j\omega}$ y $|z| = 1$.
3. Si $x(n)$ es una secuencia por la derecha, es decir que tiene valores diferentes de cero en $N_1 < n < \infty$, entonces su región de convergencia se extiende hacia el exterior de un círculo y puede incluir $|z| = \infty$. La región de convergencia de una secuencia por la derecha es $|z| > |z_1|$, y z_1 es el valor más pequeño donde la transformada converge. Si la región de convergencia incluye al ∞ entonces la secuencia es causal.
4. Si $x(n)$ es una secuencia por la izquierda, es decir, que tiene valores diferentes de cero en $-\infty \leq n \leq N_2$, entonces su región de convergencia se extiende hacia el interior de un círculo y puede incluir al origen. Si $N_2 > 0$, la región de convergencia no incluye al origen.
5. Una secuencia por ambos lados es aquella que se extiende de $-\infty$ hasta ∞ y genera una región de convergencia en forma de anillo o simplemente no converge. Esto se concluye al pensar que la secuencia por ambos lados puede descomponerse en una secuencia por

la derecha y una por la izquierda, cuyas regiones pueden o no intersectarse. La región posible de convergencia será el único sitio donde toda la secuencia converja, de modo que sólo en la intersección quedará definida la TZ.

6. La ROC resultante cuando combinamos varias transformadas Z, es al menos la intersección de las ROC de las transformadas individuales.
7. Si la combinación lineal de varias señales discretas es de duración finita (ejemplo $x(n) = U(n) - U(n-k)$), entonces la ROC de $X(z)$ está determinada por la naturaleza finita de la señal, no por las regiones ROCs de las transformadas individuales.
8. Si $x(n)$ es una secuencia de duración finita, es decir que sólo existe en un intervalo finito $[N_1, N_2]$ tal que $-\infty < N_1 \leq n \leq N_2 < \infty$, entonces su región de convergencia es todo el plano z, excepto posiblemente en $z = 0$ ó $z = \infty$. Es decir,

$$X(z) = \sum_{n=N_1}^{N_2} x(n) z^{-n}$$

Una condición necesaria para que la serie $x(n)$ converja es que $|x(n)| < \infty$, si esto se cumple entonces z adopta todos los valores del plano excepto $z = \infty$ cuando $N_1 < 0$ y $z = 0$ cuando $N_2 > 0$.

Una condición necesaria y suficiente para que un sistema converja es que la magnitud de sus polos $|p_i| < 1$, esto implica que los polos de la función de transferencia del sistema estable están dentro del círculo unitario.

5.5 ESTABILIDAD DE UN SISTEMA DISCRETO

Como se mencionó a inicio del capítulo, la TZ es una herramienta para el análisis de sistemas, en esta sección se verifica la estabilidad de un sistema desde el dominio del tiempo discreto y el dominio de z utilizando su función de transferencia $H(z)$.

Dada la función de transferencia $H(z)$ que es la TZ de la respuesta al impulso $h(n)$ de un sistema discreto

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - az^{-1}} \Rightarrow h(n) = a^n U(n) \quad (5.18)$$

Para diferentes valores de la constante “a” podemos verificar la convergencia de $h(n)$, estos resultados también se muestran gráficamente en la figura 5.1

- a) Si $a = \frac{1}{2} \Rightarrow h(n) = \left\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots\right\} \Rightarrow$ el sistema es estable
- b) Si $a = 2 \Rightarrow h(n) = \{1, 2, 4, 8, \dots\} \Rightarrow$ caso inestable
- c) Si $a = 1 \Rightarrow y(n) = \{1 \ 1 \ 1 \ 1 \ 1 \ 1 \dots\} \Rightarrow$ caso cuasiestable
- d) Si $a = -1 \Rightarrow y(n) = \{1 \ -1 \ 1 \ -1 \ 1 \ -1 \dots\} \Rightarrow$ caso cuasiestable
- e) Si $a = -\frac{1}{2} \Rightarrow h(n) = \left\{1, -\frac{1}{2}, +\frac{1}{4}, -\frac{1}{8}, \dots\right\} \Rightarrow$ el sistema es estable
- f) Si $a = -2 \Rightarrow h(n) = \{1, -2, 4, -8, \dots\} \Rightarrow$ el sistema es inestable

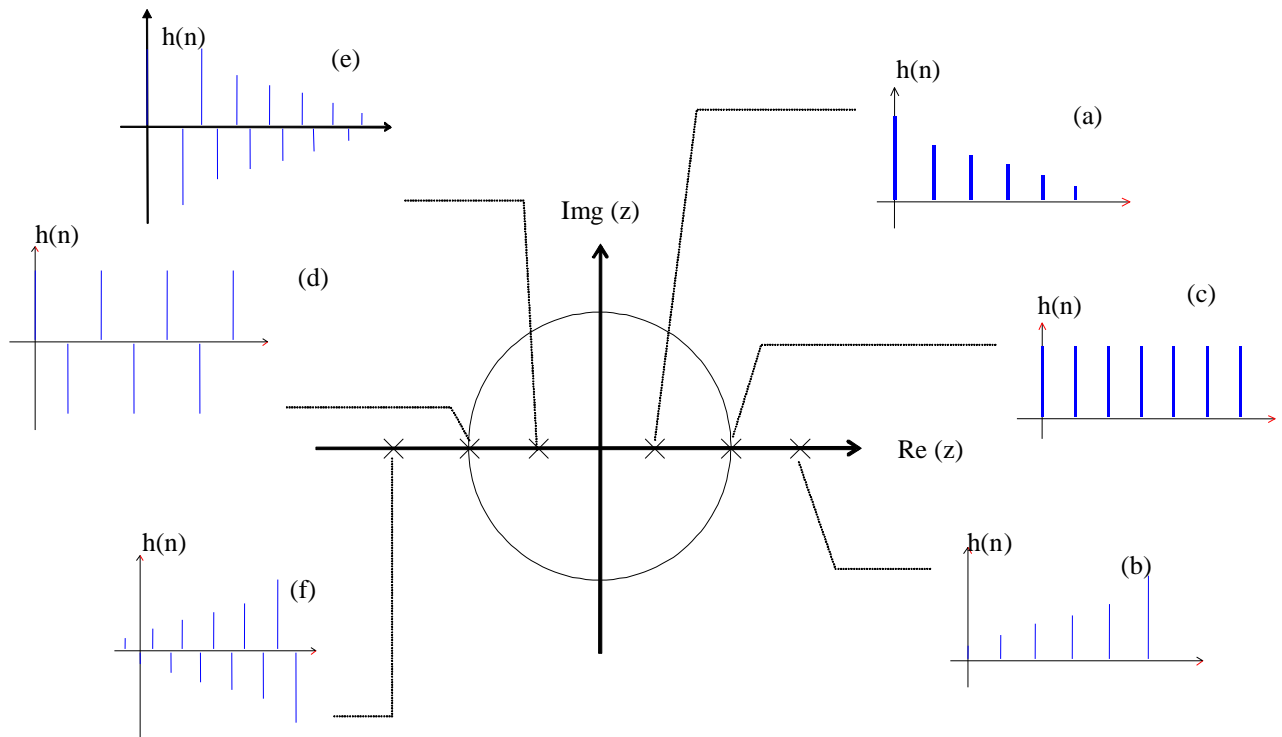


Figura 5.1: Localización de polos de H(z) en el plano z y estabilidad de h(n).

De la figura 5.1 se concluye que para este caso el sistema H(z) es estable si los polos están dentro del círculo unitario.

En aplicaciones prácticas, es de interés diseñar sistemas estables, es decir, aquellos sistemas cuyas salidas no tiendan al infinito. Se dice que una secuencia es acotada si cada elemento es menor que algún número finito M , es decir $x(n)$ es limitada si $|x(n)| < M$ para todo n . Entonces, se dice que un sistema es estable si para cada secuencia de entrada acotada produce una secuencia de salida acotada.

Considerando un sistema con una relación entre la entrada y salida dada por

$$y(n) = \sum_{m=-\infty}^{n-1} x(m) \quad (5.19)$$

Si $x(n) = \delta(n)$, entonces $y(n) = 1$ para $n \geq 0$ la salida es una secuencia acotada. Si $x(n) = U(n)$, entonces $y(n) = nU(n)$, la salida es una secuencia no acotada. En estos dos ejemplos, utilizando la ecuación (5.19) se muestra que para diferentes entradas un sistema puede o no producir salidas acotadas, independientemente de que las entradas sean acotadas. En conclusión, un sistema es estable si para cualquier entrada $x(n)$ acotada, el mismo sistema produce una salida $y(n)$ acotada.

Entonces se puede decir que un sistema SLITD es estable, si y sólo sí, el factor denotado por S es definido por

$$S = \sum_{k=-\infty}^{\infty} |h(k)| < \infty \quad (5.20)$$

Siendo este criterio necesario y suficiente. Para una entrada acotada $x(n)$ y tomando el valor absoluto de la salida expresada como una suma de convolución

$$|y(n)| = \left| \sum_{k=-\infty}^{\infty} h(k)x(n-k) \right| \quad (5.21)$$

La suma debe ser menor o igual que la suma del producto de las magnitudes.

$$\longrightarrow |y(n)| \leq \sum_{k=-\infty}^{\infty} |h(k)x(n-k)| \quad (5.22)$$

Dado que todos los valores de la entrada son acotados, para toda n se tiene:

$$\longrightarrow |y(n)| \leq M \sum_{k=-\infty}^{\infty} |h(k)| \text{ entonces } |y(n)| \leq MS \quad (5.23)$$

Si M y S son constantes finitas entonces la salida es acotada.

Para un sistema lineal su respuesta en el dominio de Z está dada por

$$Y(z) = H(z) X(z) \quad (5.24)$$

donde $H(z)$ es la función de transferencia del sistema

Para obtener la respuesta $y(n)$ del sistema, es necesario hacer un desarrollo en fracciones parciales de $Y(z)$ y luego aplicar la transformada z inversa (TZI), entonces

$$y(n) = c_1(p_1)^n + c_2(p_2)^n + c_3(p_3)^n + \dots + c_n(p_n)^n \quad (5.25)$$

donde los valores p_i son los polos de la función de transferencia y las constantes c_i se determinan al calcular la TZ inversa. Para que la respuesta permanezca acotada, se requiere que cada uno de los polos tenga una magnitud menor a uno, esto garantiza que los términos de la respuesta generada por el sistema permanezca acotada para cualquier señal de entrada acotada. Si se dibujan los polos en el plano complejo de z , un sistema estable debe caracterizarse por tener sus polos dentro de un círculo unitario centrado en el origen. Es decir, un sistema es estable cuando su respuesta al impulso es absolutamente sumable (la sumatoria es acotada).

Para un sistema causal y estable se tendrá necesariamente una $H(z)$ cuya región de convergencia incluya al círculo unitario, como se observa en la figura 5.2.

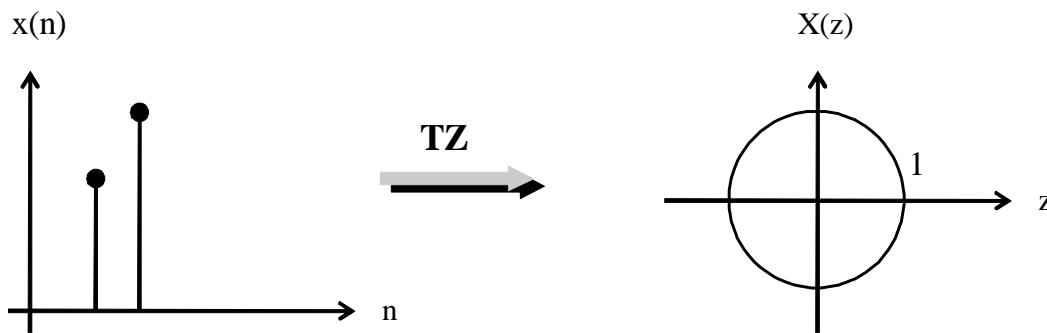


Figura 5.2 Región de convergencia.

5.6 SECUENCIA GEOMÉTRICA

En la mayoría de las aplicaciones de la transformada Z se involucran secuencias geométricas y su comprensión es esencial en el diseño de sistemas SLITD. La secuencia geométrica es una de las secuencias más utilizadas en sistemas lineales discretos y cuya definición está dada por

$$f(n) = \begin{cases} 0 & \text{para } n < 0 \\ a^n & \text{para } n \geq 0 \end{cases} \quad (5.26)$$

aplicando la TZ a la ecuación (5.26) se obtiene

$$F(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n \quad (5.27)$$

truncando la sumatoria (5.27) a N términos para poder evaluarla y aplicando el límite cuando $N \rightarrow \infty$

$$F_n(z) = \sum_{n=0}^{N-1} (az^{-1})^n = 1 + az^{-1} + a^2 z^{-2} + \dots + a^{N-1} z^{-(N-1)} \quad ; \quad N \rightarrow \infty \quad (5.28)$$

$$F_n(z) = 1 + az^{-1}(1 + az^{-1} + a^2 z^{-2} + \dots + (az^{-1})^{n-2} + (az^{-1})^{n-1}) - a^n z^{-n} \quad (5.29)$$

una de las ventajas de la versión truncada de $F(z)$ es que se pueden obtener expresiones cerradas, es decir

$$F_n(z) = \frac{1 - (az^{-1})^N}{1 - az^{-1}} \quad (5.30)$$

en la ecuación (5.30) el interés se centra en este término $(az^{-1})^N$, ya que es el único que depende de N , si $|a| < 1$ y cuando N crece arbitrariamente el término $(az^{-1})^N$ tiende a cero, por lo que $F(z)$ queda expresado de la siguiente forma

$$F(z) = \lim_{N \rightarrow \infty} F_n(z) = \frac{1}{1 - az^{-1}} = \frac{z}{z - a} \quad \text{para } |az^{-1}| > 1 \text{ y } a < 1 \quad (5.31)$$

la región de convergencia está dada para todo $|z| > a$, para $|z| < a$ se tiene la región de divergencia y $F(z)$ se hace infinita. De este último desarrollo se observa que la TZ de cualquier secuencia $x(n)$ se calcula utilizando la definición (5.3), sin embargo no siempre es fácil obtener expresiones cerradas, por lo tanto se utilizan las propiedades que se enuncian en la siguiente sección.

5.7 PROPIEDADES DE LA TZ

1. Linealidad, la TZ cumple con el teorema de superposición, es decir

$$\begin{aligned}\text{Si } X_1(z) &= TZ\{x_1(n)\} \\ X_2(z) &= TZ\{x_2(n)\} \\ Y(z) &= TZ\{y(n)\}\end{aligned}$$

si a y b son constantes y si $y(n) = ax_1(n) + bx_2(n)$

$$\Rightarrow Y(z) = TZ\{ax_1(n) + bx_2(n)\} = aX_1(z) + bX_2(z) \quad (5.32)$$

2. Retardo en el tiempo (atraso)

$$TZ\{x(n-1)\} = z^{-1} X(z) \quad (5.33)$$

$$TZ\{x(n-m)\} = z^{-m} X(z) \quad (5.34)$$

Las propiedades de linealidad y desplazamiento en el tiempo son las características fundamentales que hacen de la TZ una herramienta poderosa en el análisis de sistemas SLITD.

3. Adelanto

$$\begin{aligned}TZ\{x(n+m)\} &= z^m X(z) - \sum_{n=0}^{m-1} x(n)z^{m-n} = z^m X(z) - z^m x(0) - z^{m-1} X(1) + \dots \\ &+ zx(m-1)\end{aligned} \quad (5.35)$$

4. Multiplicación por a^n

$$TZ\{a^n x(n)\} = X\left(\frac{z}{a}\right) = X(a^{-1}z) \quad (5.36)$$

5. Diferenciación

$$TZ\{nx(n)\} = -z \frac{dX(z)}{dz} \Rightarrow TZ\{n^m x(n)\} = \left(-z \frac{d}{dz}\right)^m X(z) \quad (5.37)$$

6. Convolución

Para un sistema SLITD con entrada $x(n)$, con respuesta al impulso $h(n)$ y salida $y(n)$, donde $y(n) = h(n)*x(n)$

$$\Rightarrow TZ\{y(n) = h(n) * x(n)\} \quad (5.38)$$

$$Y(z) = H(z) X(z) \quad (5.39)$$

es decir, que la convolución de las secuencias $x(n)$ y $h(n)$ en el tiempo discreto implica la multiplicación de sus transformadas Z en el dominio de Z. Despejando de la ecuación (5.39) la función de transferencia del sistema se define

$$H(z) = \frac{Y(z)}{X(z)} \quad (5.40)$$

Demostración del Teorema de la Convolución Discreta

$$Y(z) = X(z) \cdot H(z) \quad (5.41)$$

para un sistema SLITD con entrada $x(n)$, respuesta al impulso $h(n)$ y salida $y(n)$ dada por la ecuación (5.42)

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) \quad (5.42)$$

aplicando TZ a la ecuación (5.42) y si $i = n-m$, entonces $n = i+m$

$$Y(z) = \sum_{n=-\infty}^{\infty} \left\{ \sum_{m=-\infty}^{\infty} x(m)h(n-m) \right\} z^{-n}$$

$$Y(z) = \sum_{i+m=-\infty}^{\infty} \left\{ \sum_{m=-\infty}^{\infty} x(m)h(i) \right\} z^{-i} z^{-m}$$

separando las sumatorias con índices m e i

$$Y(z) = \sum_{i=-\infty}^{\infty} h(i)z^{-i} \sum_{m=-\infty}^{\infty} x(m)z^{-m} \quad (5.43)$$

por definición de TZ se obtiene la ecuación (5.41)

$$Y(z) = X(z) \cdot H(z)$$

7. Reversión en el tiempo

$$TZ\{x(-n)\} = X(z^{-1}) = X\left(\frac{1}{z}\right) \quad (5.44)$$

8. Para una secuencia periódica con período N

$$TZ\{x(n) = x(n+N)\} = \frac{z^N}{z^N - 1} X_1(z) \quad (5.45)$$

donde

$$X_1(z) = \sum_{n=0}^{N-1} x(n)z^{-n} \quad (5.46)$$

9. Teorema del Valor Inicial

Si $x(n)$ es causal con TZ $X(z)$ y si el $\lim_{z \rightarrow \infty} X(z)$ existe, entonces el valor inicial $x(0)$ de $x(n)$ está dado por

$$x(0) = \lim_{z \rightarrow \infty} X(z) \quad (5.47)$$

10. Teorema del Valor Final

Sea $x(n)$, donde $x(n) = 0$ para $n < 0$, es decir, que $x(n)$ es causal y tiene transformada Z $X(z)$ y todos los polos están dentro del círculo unitario, con la posible excepción de un polo simple en $z = 1$, entonces el teorema del valor final determina el valor de $x(n)$ cuando n se aproxima al infinito dado por

$$\lim_{n \rightarrow \infty} x(n) = \lim_{z \rightarrow 1} [(1 - z^{-1})X(z)] \quad (5.48)$$

Si $(z-1)X(z)$ es analítica para $|z| \geq 1$

11. Complejo conjugado

$$TZ\{x^*(n)\} = X^*(z^*) \quad (5.49)$$

$$TZ\{x^*(-n)\} = X^*\left(\frac{1}{z^*}\right) \quad (5.50)$$

12. Correlación de secuencias: dada dos señales discretas $x(n)$ y $y(n)$ con correlación

$$r_{xy}(\ell) = \sum_{n=-\infty}^{\infty} x(n)y(n-\ell) \quad (5.51)$$

$$\Rightarrow R_{xy}(z) = TZ\{r_{xy}(\ell)\} = X(z)Y(z^{-1}) \quad (5.52)$$

13. Teorema de Parseval

$$\sum_{n=-\infty}^{\infty} x(n)y^*(n) = \frac{1}{2\pi j} \int_{-\pi}^{\pi} X(z)Y\left(\frac{1}{z}\right) \frac{dz}{z} \quad (5.53)$$

14. Multiplicación de dos secuencias en el tiempo

$$TZ\{x_1(n)x_2^*(n)\} = \frac{1}{2\pi j} \oint_c X_1(v)X_2^*\left(\frac{z}{v}\right)v^{-1}dv \quad (5.54)$$

5.8 EJEMPLOS DE TZ DE FUNCIONES ELEMENTALES

A continuación se resuelven algunas TZ utilizando la definición, propiedades y la TZ de una secuencia geométrica.

Función escalón unitario

$$x(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases} \quad X(z) = TZ\{x(n)\} = \sum_{n=0}^{\infty} 1z^{-n} = \sum_{n=0}^{\infty} z^{-n}$$

$$= 1 + z^{-1} + z^{-2} + z^{-3} + \dots$$

$$= \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}$$

nótese que la serie converge si $|z| > 1$

La transformada Z de $x(n)$ obtenida de esta forma es válida a través de todo el plano Z, excepto en los polos de $X(z)$.

Función rampa unitaria

$$x(n) = \begin{cases} n & 0 \leq n \\ 0 & n < 0 \end{cases} \quad \text{Observe que} \quad x(nT) = nT \quad n = 0, 1, 2, 3, \dots$$

$$X(z) = TZ[n] = \sum_{n=0}^{\infty} x(n)z^{-n} = \sum_{n=0}^{\infty} nz^{-n} = \sum_{n=0}^{\infty} nz^{-n}$$

$$= (z^{-1} + 2z^{-2} + 3z^{-3} + \dots)$$

aplicando la propiedad de diferenciación se obtiene

$$X(z) = \frac{z^{-1}}{(1 - z^{-1})^2} = \frac{z}{(z - 1)^2}$$

Función polinomial a^n o Serie Geométrica (siendo “a” una constante)

$$x(n) = \begin{cases} a^n & n = 0, 1, 2, \dots \\ 0 & n < 0 \end{cases} \quad X(z) = TZ[a^n] = \sum_{n=0}^{\infty} x(n)z^{-n} = \sum_{n=0}^{\infty} a^n z^{-n}$$

$$X(z) = TZ[a^n] = 1 + az^{-1} + a^2z^{-2} + a^3z^{-3} + \dots = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}$$

nótese que esta serie geométrica converge si $|z| > |a|$

Función exponencial

$$x(n) = \begin{cases} e^{-an} & n \geq 0 \\ 0 & n < 0 \end{cases}$$

$$\begin{aligned} X(z) &= TZ[e^{-an}] = \sum_{n=0}^{\infty} x(n)z^{-n} = \sum_{n=0}^{\infty} e^{-an} z^{-n} \\ &= 1 + e^{-a} z^{-1} + e^{-2a} z^{-2} + e^{-3a} z^{-3} + \dots \\ &= \frac{1}{1 - e^{-a} z^{-1}} = \frac{z}{z - e^{-a}} \end{aligned}$$

Función seno

$$x(n) = \begin{cases} \text{sen } \omega_0 n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

donde: $\text{sen } \omega_0 n = \frac{1}{2j} (e^{j\omega_0 n} - e^{-j\omega_0 n})$ y $TZ \{e^{-an}\} = \frac{1}{1 - e^{-a} z^{-1}}$

$$\begin{aligned} X(z) &= TZ[\text{sen } \omega_0 n] = \frac{1}{2j} \left(\frac{1}{1 - e^{j\omega_0} z^{-1}} - \frac{1}{1 - e^{-j\omega_0} z^{-1}} \right) \\ &= \frac{1}{2j} \frac{(e^{j\omega_0} - e^{-j\omega_0}) z^{-1}}{1 - (e^{j\omega_0} - e^{-j\omega_0}) z^{-1} + z^{-2}} = \frac{z^{-1} \text{sen } \omega_0}{1 - 2z^{-1} \cos \omega_0 + z^{-2}} \\ &= \frac{z \text{sen } \omega_0}{z^2 - 2z \cos \omega_0 + 1} \end{aligned}$$

Función coseno

$$x(n) = \begin{cases} \cos \omega_o n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

procediendo en forma similar a la función seno: $TZ \{ \cos \omega_o n \} = \frac{1}{2} TZ [e^{j\omega_o n} + e^{-j\omega_o n}]$

$$\begin{aligned} X(z) &= TZ \{ \cos \omega_o n \} = \frac{1}{2} \left(\frac{1}{1 - e^{j\omega_o} z^{-1}} + \frac{1}{1 - e^{-j\omega_o} z^{-1}} \right) \\ &= \frac{1}{2} \frac{2 - (e^{-j\omega_o} + e^{j\omega_o}) z^{-1}}{1 - (e^{j\omega_o} + e^{-j\omega_o}) z^{-1} + z^{-2}} = \frac{1 - z^{-1} \cos \omega_o}{1 - 2z^{-1} \cos \omega_o + z^{-2}} \\ &= \frac{z^2 - z \cos \omega_o}{z^2 - 2z \cos \omega_o + 1} \end{aligned}$$

Función seno amortiguada

$$x(n) = \begin{cases} e^{-an} \sin \omega_o n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

$$\begin{aligned} X(z) &= TZ [e^{-an} \sin \omega_o n] \\ &= \frac{1}{2j} TZ [e^{-an} e^{j\omega_o n} - e^{-an} e^{-j\omega_o n}] \\ &= \frac{1}{2j} \left[\frac{1}{1 - e^{-(a-j\omega_o)} z^{-1}} - \frac{1}{1 - e^{-(a+j\omega_o)} z^{-1}} \right] \\ &= \frac{1}{2j} \left[\frac{(e^{j\omega_o} - e^{-j\omega_o}) e^{-a} z^{-1}}{1 - (e^{j\omega_o} - e^{-j\omega_o}) e^{-a} z^{-1} + e^{-2a} z^{-2}} \right] \\ &= \frac{e^{-a} z^{-1} \sin \omega_o}{1 - 2e^{-a} z \cos \omega_o + e^{-2a} z^{-2}} \\ &= \frac{e^{-a} z \sin \omega_o}{z^2 - 2e^{-a} z \cos \omega_o + e^{-2a}} \end{aligned}$$

Función coseno amortiguada

$$x(n) = \begin{cases} e^{-an} \cos \omega_o n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

$$\begin{aligned} X(z) &= \text{TZ}[e^{-an} \cos \omega_o n] \\ &= \frac{1}{2} \text{TZ} [e^{-an} e^{j\omega_o n} + e^{-an} e^{-j\omega_o n}] \\ &= \frac{1}{2} \left[\frac{1}{1 - e^{-(a-j\omega_o)} z^{-1}} + \frac{1}{1 - e^{-(a+j\omega_o)} z^{-1}} \right] \\ &= \frac{1}{2} \left[\frac{2 - (e^{-j\omega_o} + e^{j\omega_o}) e^{-a} z^{-1}}{1 - (e^{j\omega_o} + e^{-j\omega_o}) e^{-a} z^{-1} + e^{-2a} z^{-2}} \right] \\ &= \frac{1 - e^{-a} z^{-1} \cos \omega_o}{1 - 2e^{-a} z^{-1} \cos \omega_o + e^{-2a} z^{-2}} \\ &= \frac{z^2 - e^{-a} z \cos \omega_o}{z^2 - 2e^{-a} z \cos \omega_o + e^{-2a}} \end{aligned}$$

En la tabla 5.1 se resumen las TZ de algunas funciones discretas de interés y que en conjunto con las propiedades es posible obtener la TZ de cualquier función discreta $x(n)$.

Tabla 5.1 TZ de funciones más importantes

$x(n)$	$X(z)$	ROC
$\delta(n)$	1 (5.55)	$\forall z$
$U(n)$	$\frac{1}{1-z^{-1}}$ (5.56)	$ z > 1$
$a^n U(n)$	$\frac{1}{1-az^{-1}}$ (5.57)	$ z > a $
$na^n U(n)$	$\frac{az^{-1}}{(1-az^{-1})^2}$ (5.58)	$ z > a $
$-a^n U(-n-1)$	$\frac{1}{1-az^{-1}}$ (5.59)	$ z < a $
$-na^n U(-n-1)$	$\frac{az^{-1}}{(1-az^{-1})^2}$ (5.60)	$ z < a $
$\cos(\omega_0 n) U(n)$	$\frac{1-z^{-1} \cos \omega_0}{1-2 \cos \omega_0 z^{-1} + z^{-2}}$ (5.61)	$ z > 1$
$\text{sen}(\omega_0 n) U(n)$	$\frac{-z^{-1} \sin \omega_0}{1-2 \cos \omega_0 z^{-1} + z^{-2}}$ (5.62)	$ z > 1$
$a^n \cos(\omega_0 n) U(n)$	$\frac{1-az^{-1} \cos \omega_0}{1-2a \cos \omega_0 z^{-1} + a^2 z^{-2}}$ (5.63)	$ z > a $
$a^n \text{sen}(\omega_0 n) U(n)$	$\frac{az^{-1} \sin \omega_0}{1-2az^{-1} \cos \omega_0 + a^2 z^{-2}}$ (5.64)	$ z > a $
$n U(n)$	$\frac{z}{(z-1)^2}$ (5.65)	$ z > 1$
$n^2 U(n)$	$\frac{z(z+1)}{(z-1)^3}$ (5.66)	$ z > 1$
$n^3 U(n)$	$\frac{z(z+4z+1)}{(z-1)^4}$ (5.67)	$ z > 1$
$n^2 a^n U(n)$	$\frac{az(z+a)}{(z-a)^3}$ (5.68)	$ z > a$
$n^3 a^n U(n)$	$\frac{az(z^2+4az+a^2)}{(z-a)^4}$ (5.69)	$ z > a$

5.9 DISEÑO DE OSCILADORES DISCRETOS

Los osciladores son sistemas con características de sistemas inestables que son la base de otros sistemas más complicados como generadores senoidales, osciladores controlados por voltaje (VCO), moduladores, mallas de fase amarrada (PLL), etc. En general existen tres métodos para generar una función o señal senoidal, por medio de una tabla de valores senoidales, por el desarrollo de la función seno por una serie de Taylor y un sistema SLITD inestable.

Si utilizamos el último método, un oscilador senoidal puede diseñarse como un filtro paso banda de alta calidad que solo deja pasar una frecuencia. Si se diseña en el plano z , un oscilador consta de dos polos conjugados complejos ubicados sobre el círculo unitario a una frecuencia de oscilación ω_o . Para un sistema discreto SLITD de segundo orden con la función de transferencia de un oscilador se puede escribir como

$$H(z) = \frac{1}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})} \quad (5.70)$$

donde $p_1 = p_2^*$ son polos conjugados y $p_1 = e^{j\omega_o}$, ω_o es una la frecuencia de oscilación normalizada de cero a π , y se puede calculara como $\omega_o = 2\pi f_{osc}/f_s$, f_{osc} es la frecuencia de oscilación analógica en Hertz y f_s la frecuencia de muestreo que cumple con el teorema de Nyquist. Efectuando operaciones

$$H(z) = \frac{b_o}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (5.71)$$

con los valores de las constantes: $a_2 = 1$ y $a_1 = -2\cos(\omega_o)$.

Si se aplican las fórmulas de la tabla 5.1 y propiedades de TZ se puede obtener la respuesta al impulso del sistema (5.70) como

$$h(n) = \frac{b_o}{\text{sen } \omega_o} \text{sen}((n+1)\omega_o)U(n) \quad (5.72)$$

haciendo la constante $b_o = \text{sen } \omega_o$ en la ecuación (5.72) se obtiene la respuesta

$$h(n) = \text{sen}((n+1)\omega_o)U(n) \quad (5.73)$$

para que el sistema oscile y se tenga una salida $y(n)$ igual a la respuesta al impulso dado por la ecuación (5.73), la entrada debe ser un impulso $\delta(n)$. Con ecuación en diferencias

$$y(n) = b_o x(n) + a_1 y(n-1) - a_2 y(n-2) \quad (5.74)$$

Con condiciones iniciales $y(-2)=0$, $y(-1) = -A \text{sen } \omega_o$ y A la amplitud de la señal senoidal generada.

Por otro lado un oscilador discreto también puede diseñarse considerando que su respuesta al impulso es una señal cosenoidal o senoidal con TZ las ecuaciones (5.61) y (5.62)

respectivamente de la tabla 5.1 con entrada un función impulso $\delta(n)$. Para obtener la salida como una ecuación en diferencias se escribe $H(z)$ como el cociente de $Y(z)$ con $X(z)$, se calcula la transformada Z inversa y se despeja $y(n)$.

5.10 RESUMEN

La transformada Z es una herramienta de gran importancia en el diseño y análisis de sistemas discretos. En este capítulo se ha abordado la TZ, su relación con la transformada de Laplace, la relación entre la variable z y la variable s , propiedades, el teorema de la convolución, una tabla de transformadas de funciones de interés y el diseño de un oscilador discreto utilizando conceptos relacionados con la TZ. Se hace notar que teoría de TZ es mucho más extensa, sin embargo, en este capítulo solo se han considerado los temas necesarios para los propósitos de los conceptos básicos del procesamiento digital de señales.

5.11 PROBLEMAS PROPUESTOS

Calcular la TZ de las secuencias:

1. $x(n) = n^2 a^n U(n)$
2. $x(n) = n^4 a^n U(n)$
3. $x(n) = (n+1) a^n U(n)$
4.
$$x(n) = \begin{cases} 1 & 0 \leq n \leq 15 \\ 0 & 16 \leq n < \infty \end{cases}$$
5. $x(n) = \{1, 3, 5, -4, 3, -8, 10\}$

CAPITULO 6

TRANSFORMADA Z INVERSA

La obtención de la transformada Z inversa (TZI) es el proceso contrario a la TZ y sirve para calcular una secuencia $x(n)$ a partir de la función $X(z)$, $x(n)$ se puede obtener en forma abierta o cerrada, es decir, como una función de n o como una secuencia infinita de valores respectivamente. Ambas soluciones son equivalentes, sin embargo, las soluciones abiertas pueden ser imprácticas debido a su longitud o cuando se requiere saber un valor particular a un tiempo n determinado. En este capítulo se presentan cuatro métodos de solución para obtener la TZI, para facilitar su comprensión se le da seguimiento a un ejemplo.

6.1 MÉTODOS PARA OBTENER LA TZI

En esta sección se presentan cuatro métodos para el cálculo de la TZI, por definición de TZI y uso de la teoría de Cauchy, por desarrollo en fracciones parciales, por división de polinomios y el método computacional, los dos primeros generan una solución cerrada mientras que los dos últimos una solución abierta.

6.1.1 POR DEFINICIÓN DE TZI

La transformada Z inversa de $X(z)$ se define como

$$x(n) = \frac{1}{2\pi j} \oint_c X(z)z^{n-1} dz \quad (6.1)$$

donde c es una curva cerrada que encierra al origen y los polos de $X(z)$.

La definición de TZI dada por la ecuación (6.1) se puede obtener utilizando la teoría de Cauchy, es decir, partiendo de la definición de TZ y utilizando la variable m para el tiempo discreto

$$X(z) = \sum_{m=-\infty}^{\infty} x(m)z^{-m} \quad (6.2)$$

multiplicando ambos lados de la ecuación (6.2) por z^{n-1} e integrado sobre un contorno cerrado c que encierra al origen en el plano complejo z

$$\oint_c X(z)z^{n-1} dz = \oint_c \sum_{m=-\infty}^{\infty} x(m)z^{n-1-m} dz \quad (6.3)$$

por linealidad de operadores sumatoria e integral la ecuación (6.3) se reescribe como

$$\oint_c X(z)z^{n-1} dz = \sum_{m=-\infty}^{\infty} x(m) \oint_c z^{n-1-m} dz \quad (6.4)$$

por la Teoría de Cauchy

$$\frac{1}{2\pi j} \oint_c z^{n-1-m} dz = \begin{cases} 1 & n = m \\ 0 & n \neq m \end{cases} \quad (6.5)$$

entonces para $m=n$

$$\oint_c X(z)z^{n-1} dz = 2\pi j x(n) \quad (6.6)$$

despejando $x(n)$ de (6.6) se llega a la definición de TZI

$$x(n) = \frac{1}{2\pi j} \oint_c x(z)z^{n-1} dz \quad (6.7)$$

TEOREMA DE RESIDUOS DE CAUCHY

Si $f(z)$ es una función de variable compleja z y c es una trayectoria cerrada en el plano z , y si existe la derivada $df(z)/dz$ sobre y dentro del contorno de c y $f(z)$ no tiene polos en $z = z_0$, entonces

$$\frac{1}{2\pi j} \oint_c \frac{f(z)}{z - z_0} dz = \begin{cases} f(z_0) & \text{si } z \text{ está dentro de } c \\ 0 & \text{si } z \text{ está fuera de } c \end{cases} \quad (6.8)$$

En forma mas general para polos múltiples de orden k , si existe la derivada $(k-1)$ de $f(z)$ y no existen polos en $z = z_0$, entonces

$$\frac{1}{2\pi j} \oint_c \frac{f(z)}{(z-z_0)^k} dz = \begin{cases} \frac{1}{(k-1)!} \left. \frac{d^{k-1} f(z)}{dz^{k-1}} \right|_{z_0} & \text{si } z_0 \text{ está dentro de } c \\ 0 & \text{si } z_0 \text{ está fuera de } c \end{cases} \quad (6.9)$$

Suponiendo la integral de contorno, si $p(z) = \frac{f(z)}{g(z)}$ y $g(z)$ tiene raíces distintas z_1, z_2, \dots, z_n , dentro de c , entonces se tiene

$$\frac{1}{2\pi j} \oint_c \frac{f(z)}{g(z)} dz = \frac{1}{2\pi j} \oint_c \left[\sum_{i=1}^n \frac{A_i(z)}{z-z_i} \right] dz = \sum_{i=1}^n \frac{1}{2\pi j} \oint_c \frac{A_i(z)}{z-z_i} dz = \sum_{i=1}^n A_i(z_i) \quad (6.10)$$

donde
$$A_i(z) = (z-z_i) \frac{f(z)}{g(z)} \quad (6.11)$$

Los valores $A_i(z)$ corresponde a los residuos en $z=z_i$, aquí el valor de la integral de contorno es igual a la suma de residuos de todos los polos dentro de contorno c .

En el caso de la TZI

$$x(n) = \frac{1}{2\pi j} \oint_c X(z)z^{n-1} dz = \sum_{\forall \text{ polo}} \text{residuo } X(z)z^{n-1} \Big|_{z=z_i} = \sum (z-z_i)X(z)z^{n-1} \Big|_{z=z_i} \quad (6.12)$$

este método genera una solución cerrada de $x(n)$.

Utilizando la Teoría de Fourier, como $z=e^{j\omega T}$ y sustituyendo en la definición de TZ obtenemos

$$F(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} f(n)e^{-j\omega T} \quad (6.13)$$

como $F(z)$ es periódica con período $2\pi/T$, sus coeficientes se calculan

$$f(n) = \frac{1}{2\sigma} \int_{-\sigma}^{\sigma} F(e^{j\omega T}) e^{jn\omega T} d\omega \quad (6.14)$$

$$\sigma = \frac{\pi}{T} \quad (6.15)$$

si $T=1 \Rightarrow \sigma=\pi$

$$dz = je^{j\omega} d\omega = jz d\omega \Rightarrow d\omega = \frac{dz}{jz} \quad (6.16)$$

$$\Rightarrow f(n) = \frac{1}{2\pi j} \oint_c F(z)z^{n-1} dz \quad (6.17)$$

que corresponde a la definición de TZI.

6.1.2 POR INSPECCIÓN Y RECONOCIMIENTO DE SERIES DE POTENCIAS DE z^{-i} O MÉTODO DE DIVISIÓN DIRECTA

Significa observar que $X(z)$ es una serie de potencias z^{-i} , a partir de la cual es posible reconocer los coeficientes de la secuencia $x(n)$, es decir que la TZ de una secuencia se puede escribir como

$$X(z) = \sum_{n=0}^{\infty} x(nT)z^{-n} = x(0) + x(T)z^{-1} + x(2T)z^{-2} + \dots + x(nT)z^{-n} + \dots \quad (6.18)$$

esto implica que se expande $X(z)$ de tal manera que aparezcan los coeficientes de $x(n)$ como producto de z^{-i} teniendo en cuenta la región de convergencia y llevarla a la forma original de la definición de la TZ

$$F(z) = \frac{Q(z)}{P(z)} \quad (6.19)$$

A las raíces del polinomio $Q(z)$ se le llaman los ceros de la función en z , y a las raíces del polinomio $P(z)$, polos de la función en z . Los polos y ceros de la función $F(z)$ contienen toda la información de interés de la función y juegan un papel importante en el estudio de la transformada z . La localización de los polos de un sistema y la convergencia de la respuesta al impulso es una clave característica a investigar en la respuesta en frecuencia de un sistema.

Algunas veces este procedimiento es factible efectuando la división de los polinomios que constituyen $X(z)$, donde $X(z)$ es una fracción propia.

Ejemplo 6.1

Determinar $x(n)$ para $n=0, 1, 2, 3, 4$; donde $X(z)$ está dada por

$$X(z) = \frac{10z + 5}{(z-1)(z-0.2)}$$

convirtiendo $X(z)$ en

$$X(z) = \frac{10z^{-1} + 5z^{-2}}{1 - 1.2z^{-1} + 0.2z^{-2}}$$

efectuando la división de polinomios

$$\begin{array}{r}
 10z^{-1} + 17z^{-2} + 18.4z^{-3} + 18.68z^{-4} + \dots \\
 1 - 1.2z^{-1} + 0.2z^{-2} \overline{) 10z^{-1} + 5z^{-2}} \\
 \underline{10z^{-1} - 12z^{-2} + 2z^{-3}} \\
 17z^{-2} - 2z^{-3} \\
 \underline{17z^{-2} - 20.4z^{-3} + 3.4z^{-4}} \\
 18.4z^{-3} - 3.4z^{-4} \\
 \underline{18.4z^{-3} - 22.08z^{-4} + 3.68z^{-5}} \\
 18.68z^{-4} - 3.68z^{-5} \\
 \underline{18.68z^{-4} - 22.416z^{-5} + 3.736z^{-6}}
 \end{array}$$

entonces $X(z)$ es igual al cociente

$$X(z) = 10z^{-1} + 17z^{-2} + 18.4z^{-3} + 18.68z^{-4} + \dots$$

Por inspección los coeficientes de $x(n)$ son

$$x(0) = 0, \quad x(1) = 10, \quad x(2) = 17, \quad x(3) = 18.4, \quad x(4) = 18.68$$

Este método no produce un resultado en forma cerrada de $x(n)$, excepto en ciertos casos particulares, además implica muchos cálculos para valores grandes de n .

6.1.3 MÉTODO COMPUTACIONAL

Este es un método que hace uso de la computadora, donde todos los coeficientes de $X(z)$ deben ser numéricos para que la solución sea posible.

Ejemplo 6.2

Considérese la misma expresión de la sección anterior $X(z) = \frac{10z+5}{(z-1)(z-0.2)}$

La cual se puede escribir como $X(z) = \frac{10z+5}{z^2 - 1.2z + 0.2} U(z)$

Si $U(z)$ es la TZ de la entrada, y si se hace $U(z) = \delta(z) = 1$

donde

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n = 1, 2, 3, \dots \end{cases}$$

entonces $X(z)$ se puede transformar en la siguiente ecuación en diferencias

$$x(n+2) - 1.2x(n+1) + 0.2x(n) = 10U(n+1) + 5U(n)$$

donde los datos iniciales $x(0)$ y $x(1)$ se determinan al sustituir $n = -2$ en la expresión anterior, obteniéndose

$$x(0) - 1.2x(-1) + 0.2x(-2) = 10U(-1) + 5U(-2)$$

ya que por causalidad

$$x(-1) = x(-2) = 0 \text{ y } U(-1) = U(-2) = 0, \text{ obtenemos } x(0) = 0$$

si ahora se hace la sustitución $n = -1$, entonces se obtiene

$$x(1) - 1.2x(0) + 0.2x(-1) = 10U(0) + 5U(-1)$$

es decir, $x(1) = 10$

calculando la transformada Z de $X(z)$ se llega a resolver la siguiente ecuación en diferencias para $x(n)$:

$$x(n+2) - 1.2x(n+1) + 0.2x(n) = 10U(n+1) + 5U(n)$$

con valores iniciales $x(0) = 0$, $x(1) = 10$ y $U(0) = 1$, $U(n) = 0$ para $n = 1, 2, 3, \dots$

Con base en el análisis anterior, se presentan los resultados programados para encontrar $x(n)$, la transformada Z inversa de $X(z)$, como se muestra en la tabla 6.1

De los valores obtenidos en la computadora se determinó un valor final de $x(n) = 18.75$, este valor también puede verificarse utilizando el teorema del valor final

$$\lim_{n \rightarrow \infty} x(n) = \lim_{z \rightarrow 1} \left[(1 - z^{-1})(Xz) \right] = \lim_{z \rightarrow 1} \left[(1 - z^{-1}) \frac{10z^{-1} + 5z^{-2}}{1 - 1.2z^{-1} + 0.2z^{-2}} \right] = \lim_{z \rightarrow 1} \frac{10z^{-1} + 5z^{-2}}{1 - 0.2z^{-1}} = 18.75$$

Además en la tabla 6.1 se observa que $x(n)$ converge a este valor final para $n > 3$.

n	x(n)
0	0
1	10
2	17
3	18.4
4	18.68
5	18.736
6	18.7472
7	18.7495
8	18.7499
9	18.75
10	18.75
11	18.75
12	18.75
13	18.75
14	18.75
15	18.75

Tabla 6.1 Resultados del ejemplo 6.2

6.1.4 POR EXPANSIÓN EN FRACCIONES PARCIALES Y UTILIZANDO TABLAS DE TRANSFORMACIÓN

Para un SLITD con función de transferencia $H(z)$ expresada en la forma general de cocientes de polinomios

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \quad (6.20)$$

la cual se puede expandir en fracciones parciales simples cuyas transformadas inversas son conocidas mediante tablas de transformación. Suponiendo que se tienen polos simples (no repetidos) y que $N > M$, entonces $H(z)$ se puede expandir en fracciones parciales

$$H(z) = \sum_{k=1}^N \frac{A_k}{1 - P_k z^{-k}} \quad (6.21)$$

donde los coeficientes A_k son los residuos k de $H(z)$ que se calculan

$$A_k = H(z)(1 - P_k z^{-k}), \quad z = P_k \quad (6.22)$$

Calculando la Transformada Z inversa, se puede calcular la respuesta al impulso $h(n)$ como

$$h(n) = \sum_{k=1}^N A_k (P_k)^n; n > 0 \quad (6.23)$$

Cuando $M \geq N$ hay que efectuar la división de polinomios hasta que el orden del numerador sea menor que el orden del denominador.

Ejemplo 6.3

Dada la función $X(z)$, donde a es una constante y T representa el periodo de muestreo, determine la TZI de $x(nT)$.

$$X(z) = \frac{(1 - e^{-aT})z}{(z-1)(z - e^{-aT})} \quad (6.24)$$

La expansión en fracciones parciales de $X(z)/z$ se determina haciendo:

$$\frac{X(z)}{z} = \frac{1}{z-1} - \frac{1}{z - e^{-aT}} = \frac{1}{1 - z^{-1}} - \frac{1}{1 - e^{-aT} z^{-1}} \quad (6.25)$$

nótese que

$$\text{TZI} \left\{ \frac{z}{z-1} \right\} = U(nT) \quad (6.26)$$

y

$$\text{TZI} \left\{ \frac{z}{z - e^{-aT}} \right\} = e^{-anT} U(nT) \quad (6.27)$$

La TZI de $X(z)$ está dada por

$$x(nT) = [1 - e^{-anT}] U(nT) \quad (6.28)$$

6.2 POLOS DE ORDEN SUPERIOR

Para una función $F(z)$ con polos de orden superior, es decir, polos que se repiten l veces, la función $F(z)$ se puede expandir

$$F_0(z) = \frac{F(z)}{z} = \frac{A_1}{z-p} + \frac{A_2}{(z-p)^2} + \dots + \frac{A_l}{(z-p)^l} \quad (6.29)$$

los residuos A_m se calculan

$$A_m = \text{Res}[F_0(z)] = \frac{1}{(l-m)!} \lim_{z \rightarrow p} \frac{d^{l-m}}{dz^{l-m}} \{(z-p)^l F_0(z)\} \quad (6.30)$$

donde m es índice y el orden de los polos, $m=1,2,3,\dots,l$

Aplicando TZI

$$f(n) = \{A_1 n p^n + A_2 n^2 p^n + \dots + A_l n^l p^n\} U(n) \quad (6.31)$$

6.3 POLOS CONJUGADOS COMPLEJOS

Para una función $X(z)$ con polos conjugados complejos,

$$\text{Si } X(z) = \frac{1}{(1-p_1 z^{-1})(1-p_2 z^{-1})} \quad (6.32)$$

$$\text{donde } \begin{aligned} p_1 &= r_a e^{j\theta_a} \\ p_2 &= p_1^* = r_a e^{-j\theta_a} \end{aligned}$$

Desarrollando a $X(z)$ en fracciones parciales

$$X(z) = \frac{A}{1-p_1 z^{-1}} + \frac{A^*}{1-p_1^* z^{-1}} \quad (6.33)$$

$$\begin{aligned} A &= |r_A| e^{j\theta_A} \\ A^* &= |r_A| e^{-j\theta_A} \end{aligned}$$

Aplicando TZI

$$\begin{aligned} x(n) &= (A p_1^n + A^* (p_1^*)^n) U(n) \\ x(n) &= (r_A e^{j\theta_A} r_a^n e^{nj\theta_a} + r_A r_a^n e^{-j\theta_A} e^{-jn\theta_a}) U(n) \\ x(n) &= r_A r_a^n \{e^{j(\theta_A+n\theta_a)} + e^{-j(\theta_A+n\theta_a)}\} U(n) \\ x(n) &= 2r_A r_a^n \cos(\theta_A + n\theta_a) U(n) \end{aligned} \quad (6.34)$$

Ejemplo 6.4

Utilizando los cuatro métodos para obtener la transformada Z inversa, calcular la respuesta al impulso $h(n)$ del sistema dado por la ecuación en diferencias

$$y(n) = x(n) - 0.3 y(n-1) + 0.1 y(n-2)$$

Solución:

Aplicando la TZ a la ecuación anterior

$$Y(z) = X(z) - 0.3 z^{-1} Y(z) - 0.1 z^{-2} Y(z)$$

despejando $H(z) = Y(z) / X(z)$

a) *Por definición de TZI y teorema de Cauchy*

$$H(z) = \frac{1}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

La función de transferencia $H(z)$ debe de escribirse en forma no causal con el denominador factorizado para detectar sus polos

$$H(z) = \frac{z^2}{z^2 + 0.3z - 0.1} = \frac{z^2}{(z + 0.5)(z - 0.2)}$$

entonces

$$h(n) = \frac{1}{2\pi j} \oint_C \frac{z^2 z^{n-1}}{(z + 0.5)(z - 0.2)} dz = \left. \frac{zz^n}{z - 0.2} \right|_{z = -0.5} + \left. \frac{zz^n}{z + 0.5} \right|_{z = 0.2}$$

$$h(n) = [0.714(-0.5)^n + 0.2857(0.2)^n] U(n)$$

b) *Por desarrollo en fracciones parciales*

La función de transferencia $H(z)$ debe de escribirse en forma no causal con el denominador factorizado para detectar sus polos

Si $H(z) = z H_1(z)$

$$H_1(z) = \frac{H(z)}{z} = \frac{z}{(z + 0.5)(z - 0.2)} = \frac{A}{z + 0.5} + \frac{B}{z - 0.2}$$

entonces

$$A = H_1(z)(z + 0.5) \Big|_{z = -0.5} = \frac{z}{z - 0.2} \Big|_{z = -0.5} = 0.7142$$

$$B = H_2(z)(z - 0.2) \Big|_{z = 0.2} = \frac{z}{z + 0.5} \Big|_{z = 0.2} = 0.2857$$

obteniendo

$$H(z) = \frac{0.7142z}{z + 0.5} + \frac{0.2857z}{z - 0.2}$$

De tablas de TZ, se obtiene la TZI de H(z)

$$h(n) = [0.714(-0.5)^n + 0.2857(0.2)^n]U(n)$$

c) *Por división de polinomios y definición de TZ*

Los polinomios que componen a H(z) en forma causal se dividen directamente y se comparan los coeficientes del cociente con los de la definición de TZ

$$\begin{array}{r} 1 - 0.3z^{-1} + 0.19z^{-2} + \dots \\ 1 + 0.3z^{-1} - 0.1z^{-2} \overline{)1} \\ \underline{-(1 + 0.3z^{-1} - 0.1z^{-2})} \\ -0.3z^{-1} + 0.1z^{-2} \\ \underline{-(0.3z^{-1} - 0.9z^{-2} + 0.3z^{-3})} \\ 0.19z^{-2} - 0.3z^{-3} \end{array}$$

entonces

$$h(n) = \{1, -0.3, 0.19, \dots\}$$

d) *Por el método computacional*

Se utiliza directamente la ecuación en diferencias que define al sistema y se asume que la entrada x(n) es un impulso, es decir

$$y(n) = x(n) - 0.3 y(n-1) + 0.1 y(n-2)$$

es decir que h(n) = y(n) y x(n) = δ(n)

$$h(n) = \delta(n) - 0.3h(n-1) + 0.1h(n-2)$$

desarrollando la ecuación en diferencias para todo $n \geq 0$

$$h(0) = \delta(0) - 0.3h(-1) + 0.1h(-2) = 1$$

$$h(1) = \delta(-1) - 0.3h(0) + 0.1h(-1) = -0.3(1) = -0.3$$

$$h(2) = \delta(-2) - 0.3h(1) + 0.1h(0) = -0.3(-0.3) + 0.1(1) = 0.19$$

Como se observa del ejemplo 6.4, las soluciones c y d son abiertas, y se pueden obtener de las soluciones a y b que son cerradas al evaluar $h(n)$ para $n \geq 0$.

6.4 RESUMEN

En este capítulo se ha analizado la transformada Z inversa que está muy ligada a la transformada Z directa en el análisis de sistemas discretos, principalmente se ha visto cuatro métodos para el cálculo de la transformada Z inversa, sobre todo estos métodos se han explicado con ejemplos para hacerlos más ilustrativos.

6.5 PROBLEMAS PROPUESTOS

1. Dado un sistema SLITD representado por la ecuación en diferencias

$$y(n) = x(n) + 0.4y(n-1) + 0.05y(n-2)$$

calcular la respuesta al impulso $h(n)$ del sistema utilizando los cuatro métodos de TZI descritos en este capítulo.

2. Diseñar un sistema discreto que genere una señal de doble tono con frecuencias de 770 Hz y 1336 Hz a una frecuencia de muestreo de 4000 Hz.

3. Dado un sistema SLITD representado por la ecuación en diferencias

$$y(n) = x(n) + x(n-1) + y(n-1) + 0.5y(n-2)$$

calcular la respuesta al impulso $h(n)$ del sistema utilizando los cuatro métodos de TZI descritos en este capítulo.

4. Para un sistema con función de transferencia $H(z) = \log(1+az^{-1})$ calcular su respuesta al impulso con cualquier método.

CAPITULO 7

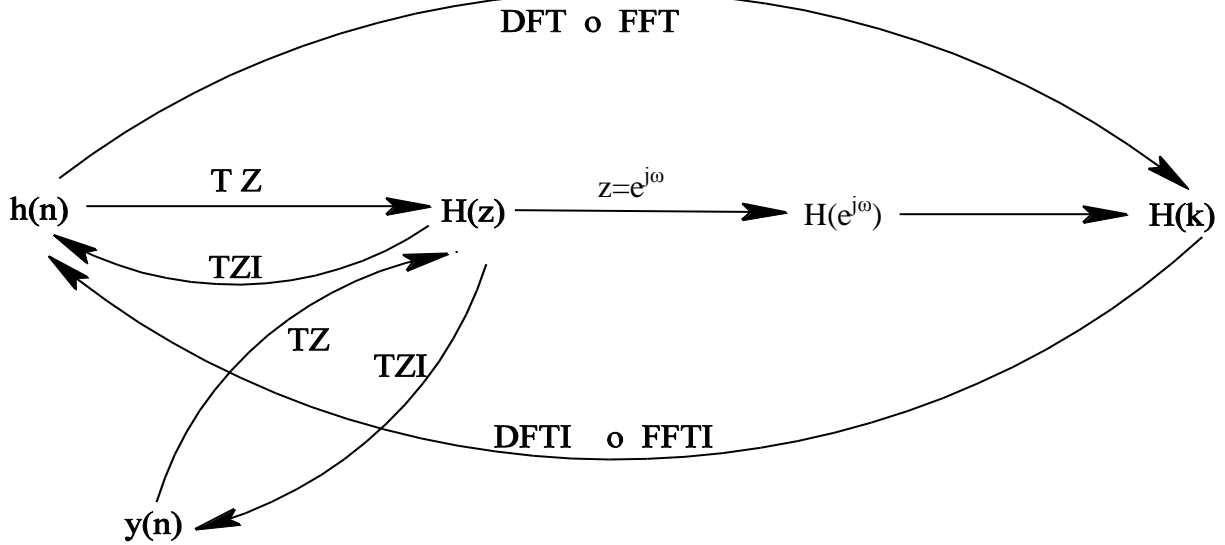
TRANSFORMADA DISCRETA DE FOURIER

La serie y la transformada de Fourier han desempeñado un papel importante en la representación y análisis de sistemas lineales en el tiempo continuo, en el mundo de las señales y sistemas discretos existe la transformada discreta de Fourier (DFT). Recordando que el espectro de una señal discreta es continuo, la DFT es una transformada que involucra un proceso de discretización.

En este capítulo se presenta una breve introducción al análisis espectral, el tratamiento de espectros discretos, se abordan una interpretación gráfica de la DFT, su desarrollo analítico, sus propiedades y características.

7.1 ESPECTRO DE SEÑALES Y SISTEMA DISCRETOS

Como se mencionó en el capítulo cuatro un sistema lineal e invariante en el tiempo discreto (SLITD) puede representarse de varias formas, considerando los conceptos vistos en este material se ha llegado hasta la obtención de la función de transferencia $H(z)$ de un sistema. Como se observa en el diagrama de la figura 7.1, de la función $H(z)$ podemos obtener el espectro de un sistema discreto al sustituir $z=e^{j\omega}$, es decir obtenemos la respuesta en frecuencia $H(e^{j\omega})$ del sistema, que se conoce como la transformada de Fourier en el tiempo discreto (TDFT).



7.1 Representación de un sistema SLITD

En general un sistema SLITD con función de transferencia $H(z)$ con q ceros y p polos, su TDFT se puede escribir

$$H(e^{j\omega}) = \frac{\sum_{k=0}^q b_k e^{-j\omega k}}{1 + \sum_{k=1}^p a_k e^{-j\omega k}} = |H(e^{j\omega})| \angle \phi(e^{j\omega}) \tag{7.1}$$

donde:

$|H(e^{j\omega})|$ es la magnitud

$\phi(e^{j\omega})$ la fase

El intervalo de análisis de la frecuencia ω es de $-\pi < \omega < \pi$ o $0 < \omega < 2\pi$, además ω es una variable continua, por tanto el cálculo de espectro $H(e^{j\omega})$ necesita de infinidad de cálculos, que serían imposibles de realizar en una computadora, por tanto es necesario utilizar otras herramientas que hagan posible el manejo discreto del espectro de una señal discreta. Para efectos de reducir la cantidad de operaciones matemáticas se discretiza la frecuencia ω de $H(e^{j\omega})$, se obtiene el espectro discreto $H(k)$ de un sistema SLITD. Como se observa en la figura 7.1, la función discreta $H(k)$ se puede obtener directamente de la respuesta al impulso

del sistema a través de la transformada discreta de Fourier (DFT) o la transformada rápida de Fourier (FFT). La DFT se tratará en este capítulo y la FFT en el capítulo ocho.

7.2 LA TRANSFORMADA DISCRETA DE FOURIER

Al expresar una secuencia cualquiera como una combinación lineal de exponenciales complejas z^n (representación de Fourier), donde dichas funciones son funciones propias o eigenfunciones de un sistema lineal discreto e invariante en el tiempo, la salida del sistema es la misma secuencia compleja z^n multiplicada por una función $H(z)$ que depende del sistema. Cuando se hizo uso de la transformada Z se comprobó que $z=e^{j\omega}$.

La transformada discreta de Fourier (DFT), al igual que la forma continua, es reversible. Sin embargo, tiene la característica de ser periódica, a pesar de que en algunos casos esta periodicidad no sea evidente, la transformada tiene un período igual a la longitud N de la señal usada para la transformación. Para una señal $x(n)$ acotada la transformada discreta de Fourier se calcula para una cantidad de N puntos.

La DFT está definida por

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi kn/N)} \quad (7.2)$$

y la transformada discreta de Fourier inversa (DFTI)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j(2\pi kn/N)} \quad (7.3)$$

$k = 0, 1, 2, \dots, N-1$, corresponde al índice del espectro discreto
 $n = 0, 1, 2, \dots, N-1$, el índice del tiempo discreto

Ambas secuencias son periódicas y se conocen como ecuaciones de síntesis y análisis. También se puede definir la DFT y la DFTI como

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad (7.4)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \quad (7.5)$$

donde

$k = 0, 1, 2, \dots, N-1$

$n = 0, 1, 2, \dots, N-1$

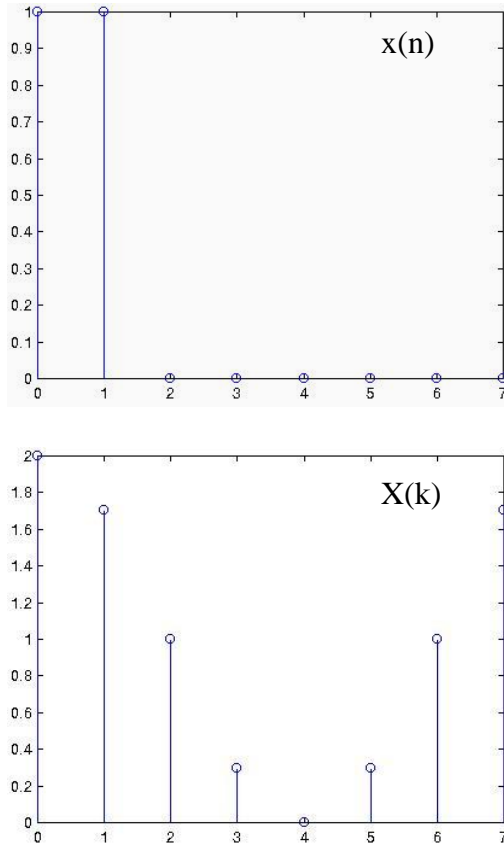
Los factores exponenciales $W_N^k = e^{-j2\pi k/N}$ se les conoce como factores “twiddle”.

Ejemplo 7.1

Dada la secuencia discreta $x(n) = \{1, 1, 0, 0, 0, 0, 0, 0\}$ hallar su DFT. Obsérvese primeramente, que $N = 8$, es decir, se debe calcular la DFT para 8 puntos.

Entonces

$$X(k) = \sum_{n=0}^{8-1} x(n) e^{-j\frac{2\pi kn}{8}} = (1) e^{-j\frac{2\pi k(0)}{8}} + (1) e^{-j\frac{2\pi k(1)}{8}} = 1 + e^{-j\frac{\pi k}{4}}$$



$$X(0) = 1 + e^{-j\frac{\pi(0)}{4}} = 1 + 1 = 2 \angle 0^\circ$$

$$X(1) = 1 + e^{-j\frac{\pi(1)}{4}} = 1.8478 \angle -22.5^\circ$$

$$X(2) = 1 + e^{-j\frac{\pi(2)}{4}} = 1.4142 \angle -45.0^\circ$$

$$X(3) = 1 + e^{-j\frac{\pi(3)}{4}} = 0.7654 \angle -67.5^\circ$$

$$X(4) = 1 + e^{-j\frac{\pi(4)}{4}} = 0 \angle 0^\circ$$

$$X(5) = 1 + e^{-j\frac{\pi(5)}{4}} = 0.7654 \angle 67.5^\circ$$

$$X(6) = 1 + e^{-j\frac{\pi(6)}{4}} = 1.4142 \angle 45^\circ$$

$$X(7) = 1 + e^{-j\frac{\pi(7)}{4}} = 1.8478 \angle 22.5^\circ$$

Figura 7.2 $x(n)$ y su DFT

Obsérvese en el ejemplo anterior que la DFT es simétrica respecto a $\frac{N}{2} = 4$.

Factores Twiddle para N=8

$$W_N^k = e^{j\frac{2\pi k}{N}}$$

$$W_8^0 = e^{j\frac{2\pi(0)}{8}} = 1$$

$$W_8^1 = e^{j\frac{2\pi(1)}{8}} = \cos\left(\frac{\pi}{4}\right) - j \operatorname{sen}\left(\frac{\pi}{4}\right) = 0.707 - 0.707j$$

$$W_8^2 = e^{j\frac{2\pi(2)}{8}} = \cos\left(\frac{\pi}{2}\right) - j \operatorname{sen}\left(\frac{\pi}{2}\right) = -j$$

$$W_8^3 = e^{j\frac{2\pi(3)}{8}} = \cos\left(\frac{3}{4}\pi\right) - j \operatorname{sen}\left(\frac{3}{4}\pi\right) = -0.707 - 0.707j$$

$$W_8^4 = e^{j\frac{2\pi(4)}{8}} = \cos(\pi) - j \operatorname{sen}(\pi) = -1$$

$$W_8^5 = e^{j\frac{2\pi(5)}{8}} = \cos\left(\frac{5}{4}\pi\right) - j \operatorname{sen}\left(\frac{5}{4}\pi\right) = -0.707 + 0.707j$$

$$W_8^6 = e^{j\frac{2\pi(6)}{8}} = \cos\left(\frac{3}{2}\pi\right) - j \operatorname{sen}\left(\frac{3}{2}\pi\right) = +j$$

$$W_8^7 = e^{j\frac{2\pi(7)}{8}} = \cos\left(\frac{7}{4}\pi\right) - j \operatorname{sen}\left(\frac{7}{4}\pi\right) = 0.707 + 0.707j$$

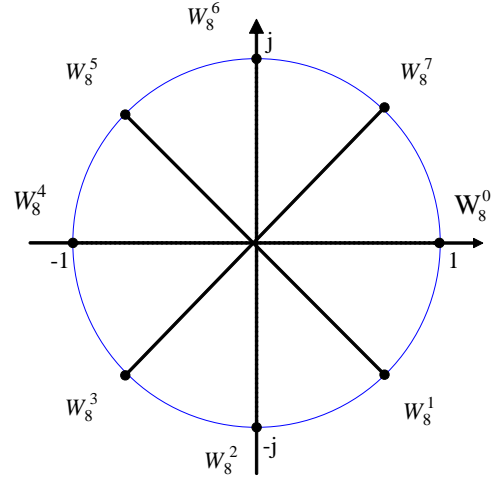


Figura 7.3 Factores Twiddle en el plano complejo.

7.3 INTERPRETACIÓN GRÁFICA DE LA DFT

Para tener una explicación gráfica de la DFT, se considera la función $h(t)$ continua y su transformada de Fourier $H(f)$ (figura 7.4a), para determinar la transformada de Fourier por medio de un análisis discreto, es necesario muestrear a la frecuencia f_s mayor o igual al doble que el ancho de banda de la señal $h(t)$ por el Teorema de Nyquist. El muestreo se produce al multiplicar la función $h(t)$ por un tren de impulsos (figura 7.4b), la función muestreada y su transformada se ilustran en la figura 7.4c.

El par de gráficas de la figura 7.4c representa la primera modificación al par original de la figura 7.4a. En este primer punto el par transformado difiere del original por el efecto de “aliasing” que resulta del muestreo. Si la señal $h(t)$ es limitada en banda y es muestreada con un intervalo menor o igual a $\frac{1}{2f_c}$, donde f_c es la componente mayor de frecuencia de $h(t)$ (su ancho de banda), por lo tanto no existe pérdida de información debido al muestreo. Si la función de $h(t)$ no está limitada en banda, entonces $H(f) \neq 0$ para $|f| > f_c$, entonces el muestreo introducirá el solapamiento espectral ilustrado en la figura 7.4c. Para reducir este efecto, es necesario tomar al intervalo de muestreo lo más pequeño posible, es decir, aumentar la frecuencia de muestreo. En la figura 7.4b y 7.4e, es necesario recordar que el símbolo *, significa convolución.

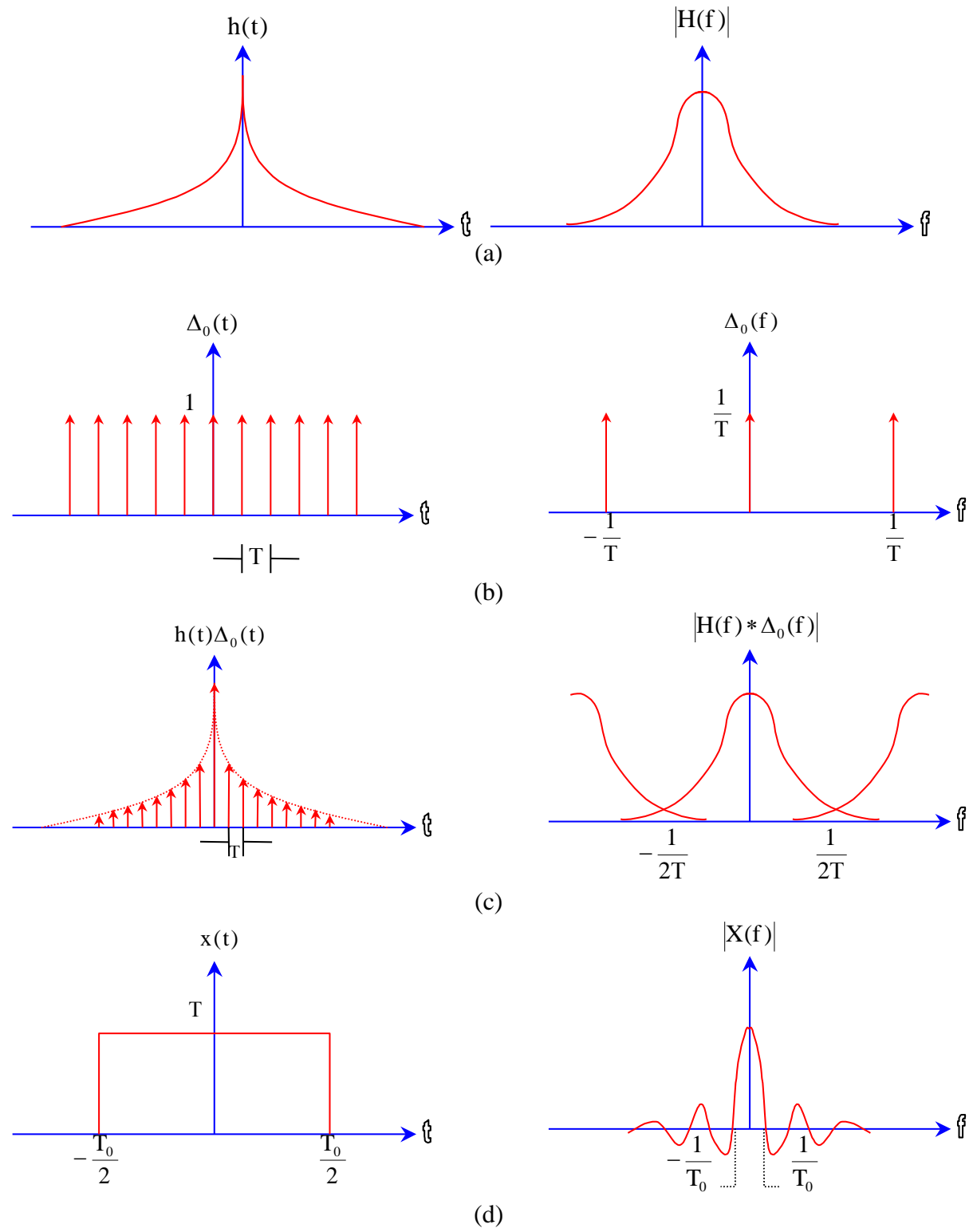


Figura 7.4 Interpretación gráfica de la DFT

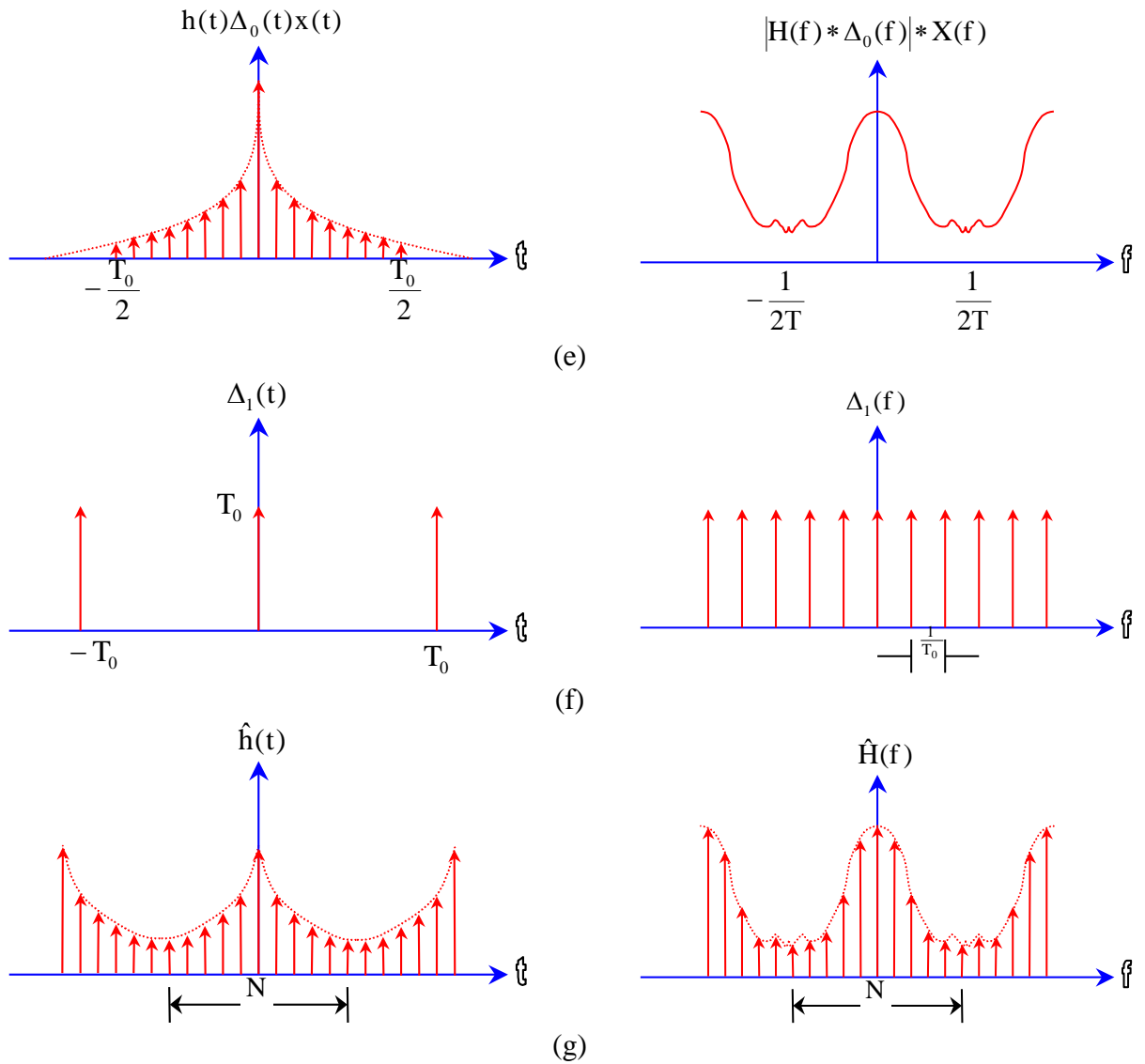


Figura 7.4 Interpretación gráfica de la DFT (continuación)

Debido a que $H(\omega)$ es periódica con período 2π , es suficiente analizar sólo este intervalo de frecuencia, por conveniencia se seleccionan N puntos igualmente espaciados en el intervalo $0 \leq \omega \leq 2\pi$, es decir que

$$\omega_k = \frac{2\pi k}{N} \tag{7.6}$$

donde k es el índice de frecuencia discreta que varía de 0 a $N - 1$

y $\frac{k}{N}$ es la frecuencia discreta

Como la función $\hat{h}(t)$ está definida para un intervalo muy grande o infinito es necesario truncarla, considerando un número finito de N puntos. La función de truncamiento y su transformada se representan en la figura 7.4d. El resultado del producto entre la secuencia infinita $\hat{h}(t)$ por la función de truncamiento $x(t)$ se muestra en la figura 7.4e. El efecto del truncamiento introduce la segunda modificación al original, esto se debe a la convolución de la transformada de la señal con la transformada de la función de truncamiento, agregando una deformación o un rizo (figura 7.4e).

Para reducir el error espectral debido al rizo, se considera que cuando la función de truncamiento crece en el dominio del tiempo, la transformada de esta función es de la forma $\text{sen}(f)/f$ en el dominio de la frecuencia y se aproxima a un impulso, esto es por la propiedad de expansión en el tiempo de la transformada de Fourier que ocasiona una contracción en frecuencia. De esta manera se reduce el error introducido por la convolución que resulta del truncamiento. Hasta aquí el espectro todavía es continuo y aún no se puede tratar digitalmente. Muestreando en la frecuencia por la función ilustrada en la figura 7.4f, donde en el intervalo de muestreo en la frecuencia está dado por $1/T_0$, este último por el teorema de convolución provoca que la función en el tiempo se vuelva periódica.

El par de funciones discretas de la figura 7.4g son el resultado para procesarse digitalmente, ya que tanto en el tiempo como en la frecuencia se tiene a las funciones representadas por valores discretos. Además en el dominio del tiempo da como resultado una función periódica. Por lo tanto, un período en el dominio del tiempo y otro en el dominio de la frecuencia están compuestos de N muestras y constituye la propiedad de convolución circular de la DFT que se explica más adelante.

7.4 DESARROLLO ANALÍTICO EN EL TIEMPO

Para obtener la transformada discreta de Fourier es necesario observar los cambios matemáticos que resultan de cada una de las operaciones sucesivas desde la señal $h(t)$ en el tiempo continuo hasta obtener el espectro discreto $H(k)$. Una primera modificación es el muestreo de la función en el dominio del tiempo, donde $\Delta_0(t)$ (figura 7.5b) es la función muestreadora en el dominio del tiempo, quedando así desarrollada la primera modificación (figura 7.4a)

$$\Delta_0(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (7.7)$$

$$\hat{h}(t) = h(t) \cdot \Delta_0(t) \quad (7.8)$$

$$\hat{h}(t) = \sum_{n=-\infty}^{\infty} h(nT) \cdot \delta(t - nT) \quad (7.9)$$

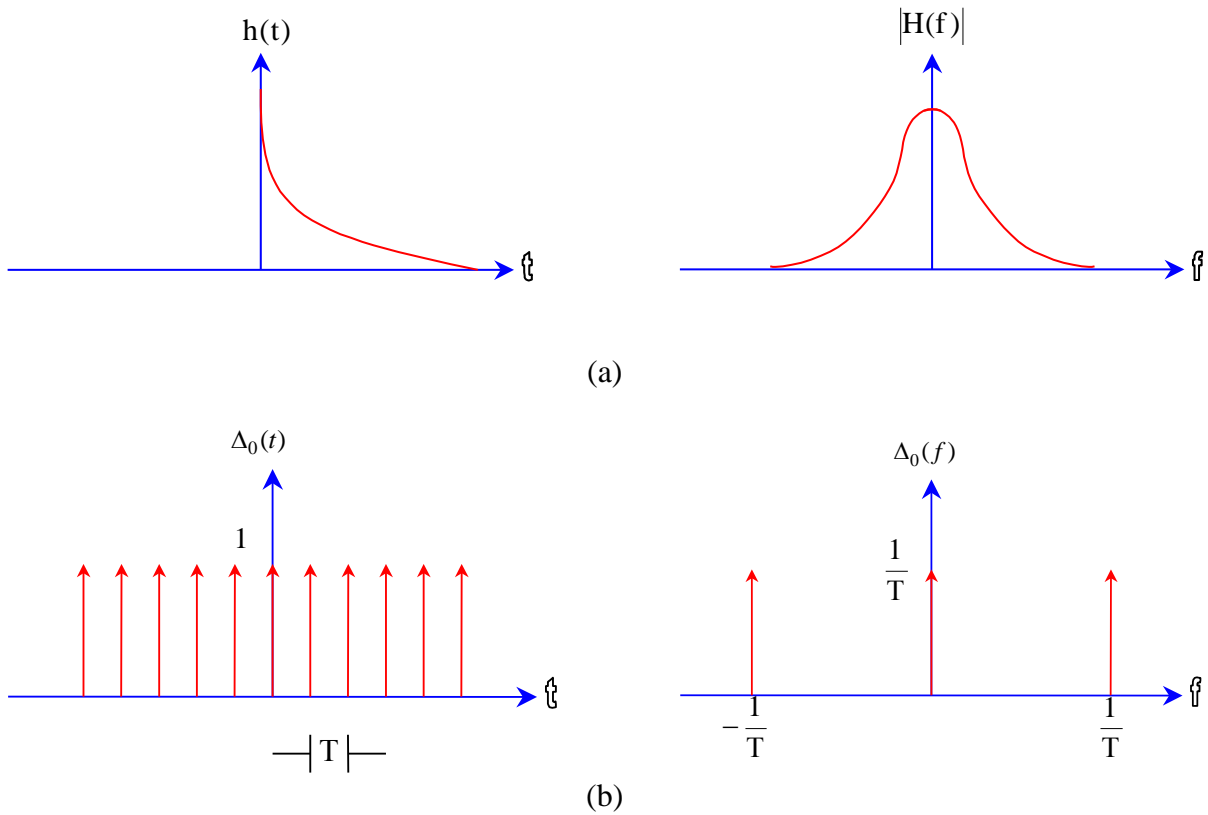


Figura 7.5 Muestreo de una señal continua

Otra modificación que debe aplicarse es el truncamiento de la función muestreada en el dominio del tiempo. Esto se logra multiplicando la función muestreada $h(t)$ (figura 7.6a) por la función de truncamiento $w(t)$, figura 7.6b

$$w(t) = \begin{cases} 1 & ; \quad -\frac{T}{2} \leq t \leq T_0 - \frac{T}{2} \\ 0 & ; \quad t < -\frac{T}{2} \text{ y } t > T_0 - \frac{T}{2} \end{cases} \quad (7.10)$$

donde T es el período de muestreo y $T_0 = NT$, de esta manera se efectúa la segunda modificación (figura 7.6c).

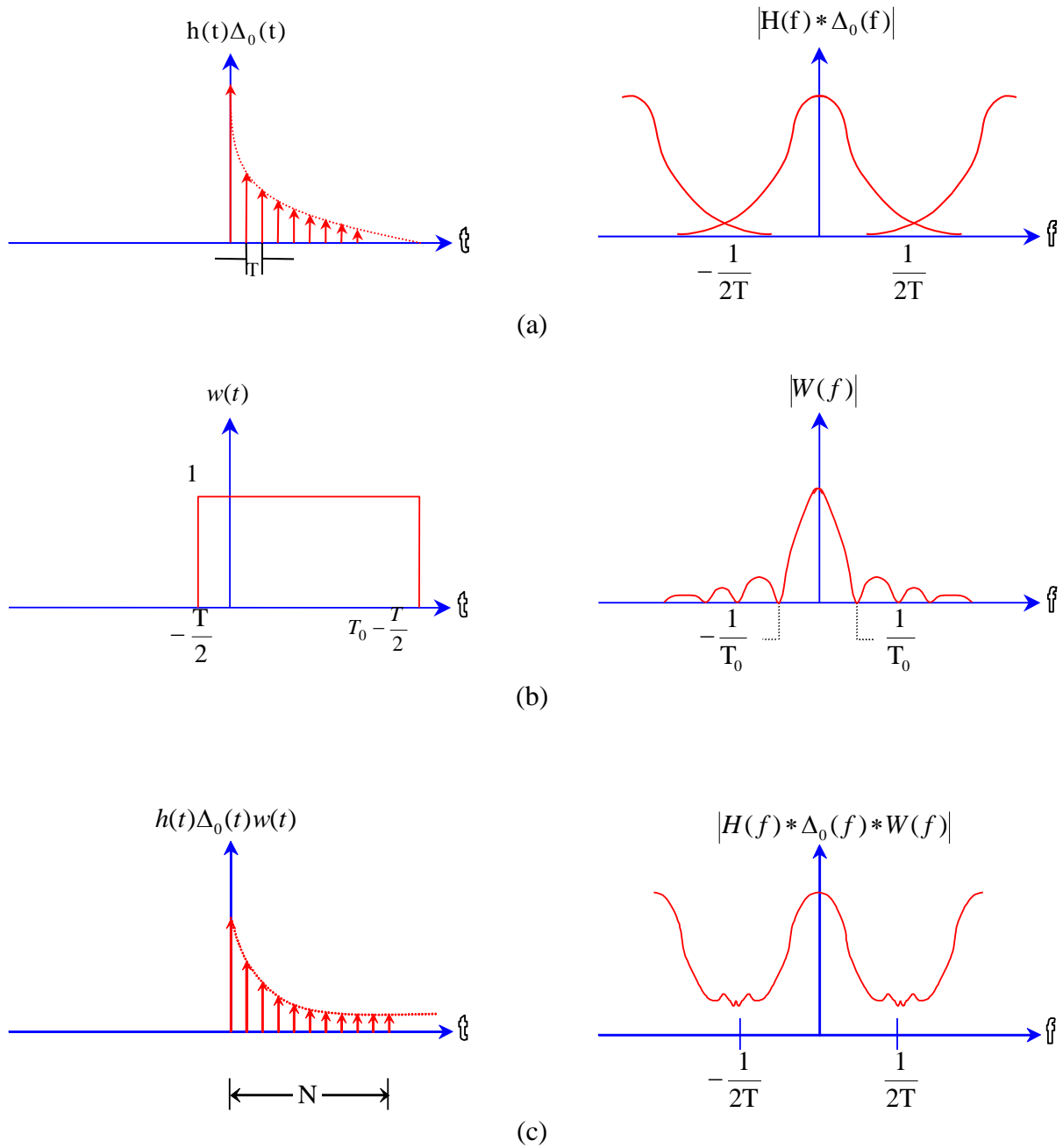


Figura 7.6 Truncamiento de una señal muestreada

la función muestreada está formada por N muestras de la siguiente manera

$$\hat{h}(t) = h(t) \cdot \Delta_0(t) \cdot w(t) \tag{7.11}$$

$$\hat{h}(t) = \left[\sum_{n=-\infty}^{\infty} h(nt) \cdot \delta(t - nT) \right] w(t) \tag{7.12}$$

aplicando el truncamiento y restringiendo la sumatoria, se obtiene

$$\hat{h}(t) = \sum_{n=0}^{N-1} h(nt) \cdot \delta(t - nT) \tag{7.13}$$

La última modificación que se debe aplicar es el muestreo en el dominio de las frecuencias de la función $|H(f) * \Delta_0(f) * W(f)|$ dada en la figura 7.7.

La función de muestreo en el dominio de la frecuencia es el tren de impulsos mostrado en la figura 7.7b y cuya expresión es

$$\Delta_1(f) = \sum_{r=-\infty}^{\infty} \delta(f - \frac{r}{T_0}) = \sum_{r=-\infty}^{\infty} \delta(f - rf_0) \ ; \ \frac{1}{T_0} = f_0 \tag{7.14}$$

La función correspondiente en el dominio del tiempo estará dada por

$$\Delta_1(t) = T_0 \sum_{r=-\infty}^{\infty} \delta(t - rT_0) \tag{7.15}$$

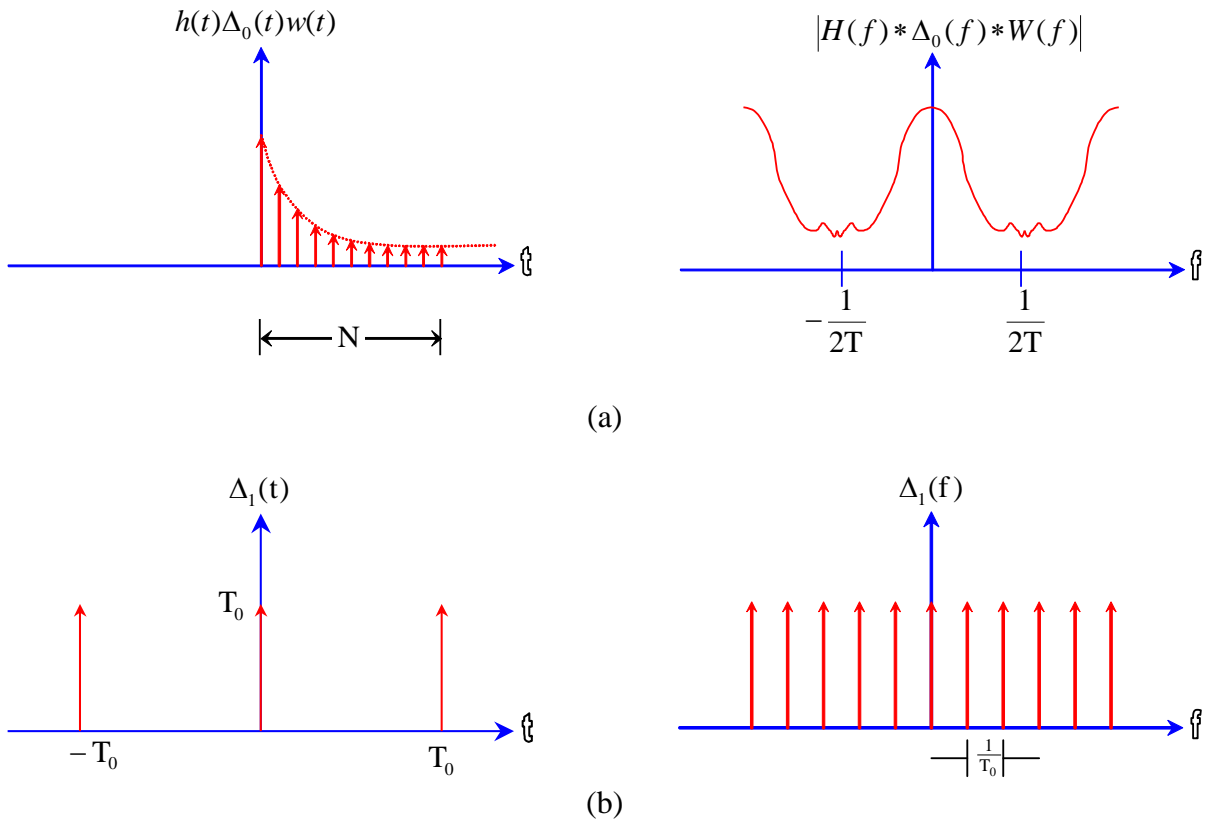


Figura 7.7 Muestreo en frecuencia

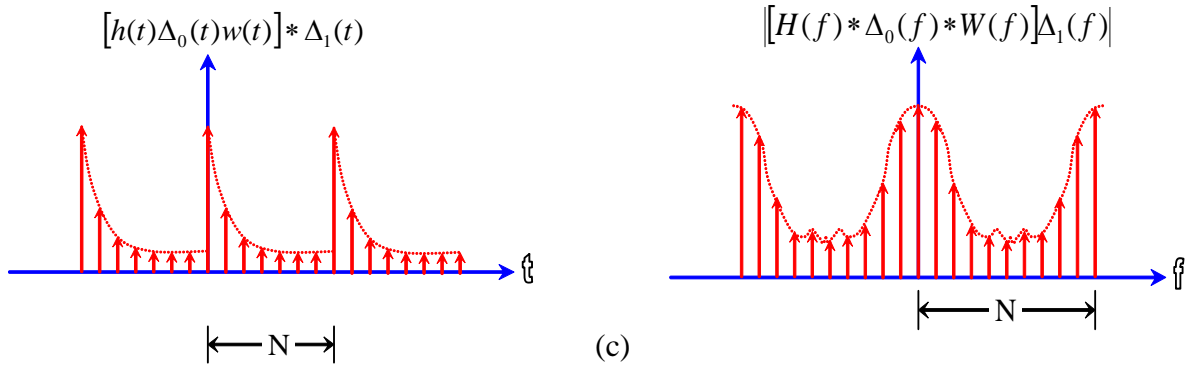


Figura 7.7 Muestreo en frecuencia (continuación)

Por el teorema de convolución, la función en el tiempo que resulta del muestreo en el dominio de la frecuencia está dada por

$$\hat{h}(t) = [h(t) \cdot \Delta_0(t) \cdot w(t)] * \Delta_1(t) \quad (7.16)$$

considerando la expresión

$$\hat{h}(t) = \sum_{n=0}^{N-1} h(nt) \cdot \delta(t - nT) \quad (7.17)$$

se obtiene la ecuación (7.18)

$$\hat{h}(t) = \left[\sum_{n=0}^{N-1} h(nT) \cdot \delta(t - nT) \right] * \left[T_0 \sum_{r=-\infty}^{\infty} \delta(t - rT_0) \right] \quad (7.18)$$

reordenando términos y desarrollando la convolución con el tren de impulsos

$$\hat{h}(t) = T_0 \sum_{r=-\infty}^{\infty} \sum_{n=0}^{N-1} h(nT) \cdot \delta(t - nT - rT_0) \quad (7.19)$$

el subíndice n indica las muestras y el subíndice r da la periodicidad de la función (figura 7.7c).

La transformada de la ecuación (7.19), es una secuencia de impulsos equidistantes multiplicados por una función D_k , de modo que por la serie de Fourier

$$\hat{H}\left(\frac{k}{T_0}\right) = \sum_{k=-\infty}^{\infty} D_k \delta(f - kf_0) \quad (7.20)$$

si se considera que

$$D_k = \frac{1}{T_0} \int_{-\frac{T}{2}}^{T_0 - \frac{T}{2}} \hat{h}(t) e^{-\frac{j2\pi kt}{T_0}} dt \quad (7.21)$$

se obtiene la ecuación (7.22)

$$D_k = \frac{1}{T_0} \int_{-\frac{T}{2}}^{T_0 - \frac{T}{2}} \left[T_0 \sum_{r=-\infty}^{\infty} \sum_{n=0}^{N-1} h(nT) \cdot \delta(t - nT - rT_0) \right] e^{-\frac{j2\pi kt}{T_0}} dt \quad (7.22)$$

por la periodicidad del espectro la integración debe ser sólo sobre un período, entonces

$$D_k = \int_{-\frac{T}{2}}^{T_0 - \frac{T}{2}} \left[\sum_{n=0}^{N-1} h(nT) \cdot \delta(t - nT) \right] e^{-\frac{j2\pi kt}{T_0}} dt \quad (7.23)$$

cambiando el orden de integración en (7.23)

$$D_k = \sum_{n=0}^{N-1} h(nT) \int_{-\frac{T}{2}}^{T_0 - \frac{T}{2}} e^{-\frac{j2\pi kt}{T_0}} \delta(t - nT) dt \quad (7.24)$$

y aplicando propiedades de la función impulso

$$D_k = \sum_{n=0}^{N-1} h(nT) e^{-\frac{j2\pi knT}{T_0}} \quad (7.25)$$

como $T_0 = NT$, entonces $T/T_0 = 1/N$; por lo tanto

$$D_k = \sum_{n=0}^{N-1} h(nT) e^{-\frac{j2\pi kn}{N}} \quad (7.26)$$

sustituyendo la ecuación (7.26) en la ecuación (7.20) se obtiene la transformada de Fourier

$$\hat{H}\left(\frac{k}{NT}\right) = \sum_{k=-\infty}^{\infty} \sum_{n=0}^{N-1} h(nT) e^{-\frac{j2\pi kn}{N}} \delta(f - kf_0) \quad (7.27)$$

Para obtener una expresión más adecuada de la transformada de Fourier discreta, es necesario demostrar que la ecuación (7.27) es una función periódica con periodicidad igual a N muestras; para $k=r$, donde r es un número entero arbitrario, entonces

$$\hat{H}\left(\frac{k}{NT}\right) = \sum_{n=0}^{N-1} h(nT) e^{-\frac{j2\pi nk}{N}} \quad (7.28)$$

sustituyendo $k = r + N$ en (7.28)

$$\hat{H}\left(\frac{r+N}{NT}\right) = \sum_{n=0}^{N-1} h(nT) e^{-\frac{j2\pi n(r+N)}{N}} \quad (7.29)$$

$$\hat{H}\left(\frac{r+N}{NT}\right) = \sum_{n=0}^{N-1} h(nT) e^{-\frac{j2\pi nr}{N}} e^{-j2\pi n} \quad (7.30)$$

Donde $e^{-j2\pi n} = 1$ para todo valor de n

por lo tanto

$$\hat{H}\left(\frac{r+N}{NT}\right) = \sum_{n=0}^{N-1} h(nT) e^{-\frac{j2\pi nr}{N}} = \hat{H}\left(\frac{r}{NT}\right) \quad (7.31)$$

De la ecuación (7.31) se concluye que la DFT sólo se calcula en N muestras distintas a través de una sola sumatoria que va de cero a $N-1$, si se desea representar a la transformada en una forma periódica, entonces se agrega la segunda sumatoria que irá de $-\infty$ a ∞ .

$$\hat{H}\left(\frac{k}{NT}\right) = \sum_{n=0}^{N-1} h(nT) e^{-\frac{j2\pi kn}{N}} \quad (7.32)$$

La expresión (7.32) define a la transformada discreta de Fourier, donde $k = 0, 1, 2, \dots, N-1$.

7.4.1 MUESTREO EN EL DOMINIO DE LA FRECUENCIA.

De la teoría del PDS sabemos que el espectro $X(e^{j\omega})$ de una señal discreta $x(n)$ es continuo y periódico, si $x(n)$ es cualquier señal discreta aperiódica, con energía finita y transformada de Fourier

$$X(\omega) = X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \quad (7.33)$$

donde $X(\omega)$ es continua y periódica con período 2π . Muestreando a $X(\omega)$ desde $-\infty$ a ∞ en N puntos equidistantes en intervalos de 2π , entonces $d\omega = 2\pi/N$ implica que evaluamos $\omega = 2\pi k/N$ (figura 7.4 y 7.7c)

$$X(\omega) = \left| \omega = \frac{2\pi k}{N} = X\left(\frac{2\pi}{N}k\right) = \dots + \sum_{n=-N}^{-1} x(n)e^{\frac{-j2\pi kn}{N}} + \sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi kn}{N}} + \sum_{n=N}^{2N-1} x(n)e^{\frac{-j2\pi kn}{N}} + \dots \right. \quad (7.34)$$

$$X\left(\frac{2\pi}{N}k\right) = \sum_{\ell=-\infty}^{\infty} \sum_{n=\ell N}^{\ell N+N-1} x(n)e^{\frac{-j2\pi kn}{N}} \quad \text{para } k = 0, 1, 2, \dots, N-1 \quad (7.35)$$

intercambiando el índice de la sumatoria interna de la ecuación (7.35) de $n=0$ a $N-1$

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{N-1} \left\{ \sum_{\ell=-\infty}^{\infty} x(n-\ell N) \right\} e^{\frac{-j2\pi kn}{N}} \quad (7.36)$$

la suma interna de (7.36) es periódica y se puede definir como

$$x_p(n) = \sum_{\ell=-\infty}^{\infty} x(n-\ell N) \quad \text{con período } N \quad (7.37)$$

expandiendo $x_p(n)$ en la serie exponencial de Fourier

$$x_p(n) = \sum_k^{N-1} c_k e^{\frac{j2\pi kn}{N}} \quad n = 0, 1, \dots, N-1 \quad (7.38)$$

los coeficientes c_k se calculan por

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{\frac{-j2\pi kn}{N}} \quad k = 0, 1, 2, \dots, N-1 \quad (7.39)$$

comparando los coeficientes c_k y $X\left(\frac{2\pi}{N}k\right)$ se concluye que

$$c_k = \frac{1}{N} X\left(\frac{2\pi}{N}k\right) \quad k = 0, 1, \dots, N-1 \quad (7.40)$$

por lo tanto

$$x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{\frac{j2\pi kn}{N}} \quad n = 0, 1, 2, \dots, N-1 \quad (7.41)$$

De la expresión (7.41) se concluye que $x_p(n)$ es la reconstrucción de $x(n)$ a partir del muestreo del espectro $X(\omega)$, $x_p(n)$ es una representación de $x(n)$ en un período N , entonces es factible reconstruir $x(n)$ de $x_p(n)$ si no existe "aliasing" en el dominio del tiempo n , es decir que $x(n)$ está limitada en el dominio del tiempo o es de longitud menor que el período N de $x_p(n)$.

Si $x(n)$ es de duración finita en el intervalo:

$$0 \leq n \leq L-1 \quad \text{donde } N \geq L$$

entonces

$$x(n) = x_p(n) \quad 0 \leq n \leq N-1$$

esto implica que $x(n)$ puede recuperarse de $x_p(n)$. De lo contrario si $N < L$ no se puede reconstruir $x(n)$.

Para reconstruir $x(n)$ se considera que $N \geq L$:

$$x(n) = \begin{cases} x_p(n) & 0 \leq n \leq N-1 \\ 0 & \text{otro } n \end{cases}$$

y dado que

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} x\left(\frac{2\pi}{N}k\right) e^{j\frac{2\pi kn}{N}} \quad 0 \leq n \leq N-1 \quad (7.42)$$

sustituyendo $x(n)$ en la respuesta espectral $X(\omega)$

$$X(\omega) = \sum_{n=0}^{N-1} \left\{ \frac{1}{N} \sum_{k=0}^{N-1} x\left(\frac{2\pi}{N}k\right) e^{j\frac{2\pi kn}{N}} \right\} e^{-j\omega n} \quad (7.43)$$

$$X(\omega) = \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) \left\{ \frac{1}{N} \sum_{n=0}^{N-1} e^{-j(\omega - 2\pi k/N)n} \right\} \quad (7.44)$$

De la ecuación (7.44), la expresión entre llaves se puede escribir como $P(\omega)$ que es conocida como función de interpolación o reconstructora. La sumatoria se puede desarrollar para obtener (7.45)

$$P(\omega) = \frac{1}{N} \sum_{n=0}^{N-1} e^{-j\omega n} = \frac{1}{N} \frac{1 - e^{j\omega N}}{1 - e^{-j\omega}} = \frac{1}{N} \frac{\sin(\omega N/2)}{\sin(\omega/2)} e^{-j\omega(N-1)/2} \quad (7.45)$$

Con la función reconstructora se puede concluir que el espectro de una señal periódica discreta de duración finita L puede ser recuperada exactamente a partir de sus muestras en frecuencia $\omega_k = 2\pi k/N$ si $N \geq L$, es decir

$$X(\omega) = \sum_{k=0}^{N-1} x\left(\frac{2\pi}{N}k\right) P\left(\omega - \frac{2\pi}{N}k\right) \quad \text{si } N \geq L \quad (7.46)$$

la interpolación de $X(\omega)$ representa el muestreo exacto de los valores de

$$X\left(\frac{2\pi k}{N}\right) \quad \text{en } \omega = \frac{2\pi k}{N}$$

(7.47)

donde $P(\omega)$ es de la forma $\sin\theta/\theta$ y periódica debido a la naturaleza de $X(\omega)$.

Además $P(\omega)$ tiene la propiedad

$$P\left(\frac{2\pi}{n}K\right) = \begin{cases} 1 & k = 0 \\ 0 & k = 1, 2, \dots, N-1 \end{cases} \quad (7.48)$$

7.5 TRANSFORMADA DISCRETA DE FOURIER INVERSA (DFTI)

En general la función equi-espaciada $X(2\pi k/N)$ donde $k = 0, 1, \dots, N-1$, no representa la secuencia original $x(n)$ cuando $x(n)$ tiene duración finita, sino que corresponde a una secuencia periódica $x_p(n)$ que es una versión alias de $x(n)$

$$X\left(\frac{2\pi k}{N}\right) \leftrightarrow x_p(n) = x_p(n - N) \Rightarrow x_p(n) = \sum_{\ell=-\infty}^{\infty} x(n - \ell N) \quad (7.49)$$

el rellenado de ceros de $x(n)$ no provee información adicional de $X(\omega)$, entonces la transformada discreta de Fourier de una secuencia $x(n)$ de longitud L se calcula el espectro

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{j\omega n} \quad 0 \leq \omega \leq 2\pi \quad (7.50)$$

muestreando la frecuencia

$$X(\omega) \text{ en } \omega_k = \frac{2\pi}{N}k, \quad ; \quad k = 0, 1, 2, \dots, N-1 \quad ; \quad N \geq L \quad (7.51)$$

se obtiene la DFT de $x(n)$

$$X(\omega_k) = X(\omega) \Big|_{\omega = \frac{2\pi k}{N}} \equiv X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} \quad ; \quad k = 0, 1, \dots, N \quad (7.52)$$

por definición de la serie exponencial de Fourier se obtiene la DFT inversa (DFTI)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi kn}{N}} \quad ; \quad n = 0, 1, 2, \dots, N-1 \quad (7.53)$$

7.6 DESARROLLO MATRICIAL DE LA DFT

La DFT también se puede escribir como un arreglo matricial, sean las secuencias de longitud N :

$x_N(n)$ en el tiempo discreto
y $X_N(k)$ en la frecuencia discreta

$$x_N(n) = [x(0) \ x(1) \ x(2) \ \dots \ x(N-1)]^T$$

$$X_N(k) = [X(0) \ X(1) \ X(2) \ \dots \ X(N-1)]^T$$

Al desarrollar la definición de la DFT (ecuación 7.4) tanto para el índice n como para el índice k se obtiene la matriz de factores Twiddle

$$W_N = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^1 & W_N^2 & \dots & W_N^{N-1} \\ 1 & W_N^2 & W_N^3 & \dots & W_N^{2(N-1)} \\ 1 & \vdots & \vdots & \vdots & W_N^{2(N-1)} \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix} \quad (7.54)$$

entonces la DFT para N puntos se puede escribir como $X_N(k) = W_N x_n(k)$, donde W_N es una matriz simétrica de transformación de $x(n)$, si existe su inversa, se cumple que $x_n(n) = W_N^{-1} X_n(k)$ es justamente una expresión de la transformada discreta de Fourier inversa (DFTI), que puede expresarse como $x_N(n) = W_N^* X_N(k) / N$ donde $W_N^{-1} = W_N^* / N$, lo que implica que $W_N^* W_N = N I_N$, donde I_N es la matriz identidad de orden N . Por lo tanto, la matriz W_N en la transformación es una matriz ortogonal unitaria y se conoce como *Matriz de Fourier*.

7.7 PROPIEDADES DE LA DFT

Las propiedades de la transformada discreta de Fourier (DFT) son básicamente las mismas de la transformada de Fourier en el tiempo continuo, ya que la primera es simplemente un caso especial de la segunda.

1) LINEALIDAD

Sea $F_1(k) = DFT\{f_1(n)\}$ y $F_2(k) = DFT\{f_2(n)\}$, las transformadas de Fourier de dos funciones, multiplicadas por las constantes a_1 y a_2 cumplen con las condiciones de superposición

$$DFT \{a_1 f_1(nT) + a_2 f_2(n)\} = a_1 F_1(k) + a_2 F_2(k) \quad (7.55)$$

- Homogeneidad: $DFT\{af(n)\} = aF(k)$

- Aditividad: $DFT\{f(n) + g(n)\} = F(k) + G(k)$

Si una transformación cumple con las dos condiciones anteriores, se dice que la transformación es lineal.

2) INVERSIÓN EN EL TIEMPO (FOLDING)

$$DFT\{x(-n)\} = DFT\{x(N-n)\} \Rightarrow X(-k) = X(N-k) \quad (7.56)$$

3) COMPLEJO CONJUGADO

La DFT de una secuencia conjugada compleja puede obtenerse al tomar el conjugado de ambos lados de la DFT

$$DFT\{x^*(n)\} = X^*(-k) = X^*(N-k) \quad (7.57)$$

$$DFT\{x^*(-n)\} = DFT\{x^*(N-n)\} = X^*(k) = X^*(N-k) \quad (7.58)$$

Por lo tanto la DFT de una secuencia conjugada es igual al conjugado complejo de la DFT de salida de la secuencia pero invertida en orden.

4) DESPLAZAMIENTO CIRCULAR

a) Desplazamiento circular en el tiempo

Sea una función $f(n)$, tal que $DFT\{f(n)\} = F(k)$. Si se desea obtener la transformada de $f[(n-l)]$ entonces

$$DFT\{f[(n-l)]\} = F(k) e^{-j2\pi kl/N} \quad (7.59)$$

b) Desplazamiento circular en las frecuencias

Sea una función $f(n)$, tal que $DFT\{f(n)\} = F(k)$. Si a esta función la definimos ahora como $f(n) e^{j2\pi ml/N}$ entonces su transformada estará definida como $F[(k-l)]$.

5) CORRIMIENTO CIRCULAR EN FRECUENCIA DE LA DFT

Si una función en frecuencia discreta es desplazada por un valor m , entonces la IDFT es una secuencia sin corrimiento modificada por una exponencial compleja que puede expresarse como

$$X(k+m) = DFT\{x(n)W_N^{mn}\}$$

$$IDFT\{X(k+m)\} = x(n)W_N^{mn} \quad (7.60)$$

6) SIMETRÍA

Sea $f(n)$ en el dominio del tiempo una función, tal que $DFT\{f(n)\} = F(k)$ de longitud N en el dominio de la frecuencia discreta, entonces $F(k)$ es simétrica respecto de $N/2$.

7) CORRELACIÓN CIRCULAR

Si $r_{xy}(l)$ es la correlación de dos secuencias discretas $x(n)$ y $y(n)$, entonces

$$r_{xy}(l) \xrightarrow{DFT} R_{xy}(k) = X(k)Y^*(k) \quad (7.61)$$

8) CONVOLUCIÓN CIRCULAR

Si se multiplican dos secuencias discretas en el tiempo, en el espectro se convolucionan circularmente

$$x_1(n)x_2(n) \xrightarrow{DFT} \frac{1}{N} X_1(k) \otimes X_2(k) \quad (7.62)$$

Por otro lado si se realiza la convolución circular de dos secuencias en el tiempo entonces

$$x_3(n) = x_1(n) \otimes x_2(n) \xrightarrow{DFT} X_1(k)X_2(k) \quad (7.63)$$

Donde la convolución circular se define como

$$x_3(m) = \sum_{n=0}^{N-1} x_1(n)x_2(m-n) \quad (7.64)$$

9) TEOREMA DE PARSEVAL

Para dos secuencias discretas $x(n)$ e $y(n)$ se cumple que

$$\sum_{n=0}^{N-1} x(n)y^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)Y^*(k) \quad (7.65)$$

si $x(n) = y(n)$, se tiene la energía de la señal $x(n)$

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 \quad (7.66)$$

la ecuación (7.66) expresa la energía de una secuencia de duración finita $x(n)$, en términos de las componentes de frecuencia discreta de $X(k)$, el símbolo * significa el conjugado complejo.

10) TRANSFORMADA DE FOURIER DE FUNCIONES SIMÉTRICAS

a) Sea una función $f(n)$ real que tiene *simetría par*, entonces su transformada discreta de Fourier se puede expresar como

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-\frac{i2\pi kn}{N}} \quad (7.67)$$

si se desarrolla la función exponencial de la transformada de Fourier

$$F(k) = \sum_{n=0}^{N-1} f(n) \cos\left(\frac{2\pi kn}{N}\right) - i \sum_{n=0}^{N-1} f(n) \sin\left(\frac{2\pi kn}{N}\right) \quad (7.68)$$

por simetría de funciones se tiene que la segunda sumatoria de la ecuación (7.68) es cero, y que la primera sumatoria es la sumatoria de una función de simetría par, por lo tanto

$$F(k) = \sum_{n=0}^{N-1} f(n) \cos\left(\frac{2\pi kn}{N}\right) \quad (7.69)$$

que es una función real, es decir

$$\phi(k) = 0 \quad \text{y} \quad |F(k)| = |R(k)| \quad (7.70)$$

La ecuación (7.70) indica que la transformada de Fourier de una función de simetría par es una función real.

b) Sea una función $f(n)$ real que tiene *simetría impar*, entonces su transformada discreta de Fourier se puede expresar como

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-\frac{i2\pi kn}{N}} \quad (7.71)$$

desarrollando la función exponencial de la transformada de Fourier

$$F(k) = \sum_{n=0}^{N-1} f(n) \cos\left(\frac{2\pi kn}{N}\right) - i \sum_{n=0}^{N-1} f(n) \operatorname{sen}\left(\frac{2\pi kn}{N}\right) \quad (7.72)$$

por simetría de funciones se tiene que la primera sumatoria de la expresión anterior es cero, y que la segunda sumatoria es la sumatoria de una función de simetría impar, por lo tanto

$$F(k) = -i \sum_{n=0}^{N-1} f(n) \operatorname{sen}\left(\frac{2\pi kn}{N}\right) \quad (7.73)$$

que es una función imaginaria pura donde

$$\phi(k) = \frac{\pi}{2} \quad \text{y} \quad |F(k)| = |I(k)| \quad (7.74)$$

11) PERIODICIDAD

La DFT y la IDFT producen resultados periódicos con período N. La periodicidad es resultado de la propiedad de periodicidad de la exponencial compleja, es decir

$$\left. \begin{array}{l} x(n) = x(n+N) \quad n=0,1,2,\dots \\ X(k) = X(k+N) \quad k=0,1,2,\dots \end{array} \right\} \text{ donde } W_N^{nN} = e^{-j2\pi n} = 1 \quad \forall n \quad (7.75)$$

12) DFT DE UNA SECUENCIA REAL

La DFT de una secuencia real produce componentes reales con frecuencias simétricas y componentes imaginarias con frecuencia antisimétricas alrededor de N/2.

En resumen la DFT provee una representación en la frecuencia discreta de una secuencia de duración finita, esto puede ser una herramienta útil para el análisis de sistemas lineales, específicamente filtros lineales, ya que permite el tratamiento digital en una computadora o un sistema de procesamiento digital de señales.

Ejemplo 7.2

Dada la secuencia discreta $x(n) = \begin{cases} 1 & 0 \leq n \leq L-1 \\ 0 & 0 > n > L-1 \end{cases}$

obtener la Transformada de Fourier en el tiempo discreto

$$X(e^{j\omega}) = \sum_{n=0}^{L-1} x(n) e^{-j\omega n}$$

por el desarrollo de la serie geométrica y la factorización de exponenciales, se tiene

$$X(e^{j\omega}) = \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} = \frac{e^{-j\omega \frac{L}{2}} \left(e^{j\omega \frac{L}{2}} - e^{-j\omega \frac{L}{2}} \right)}{e^{-j\omega \frac{1}{2}} \left(e^{j\omega \frac{1}{2}} - e^{-j\omega \frac{1}{2}} \right)} = e^{-j\omega \frac{(L-1)}{2}} \frac{\text{sen}\left(\frac{\omega L}{2}\right)}{\text{sen}\left(\frac{\omega}{2}\right)}$$

discretizando la frecuencia ω , es decir, $\omega_k = \frac{2\pi k}{N}$

$$X(e^{j\omega_k}) = e^{-j\frac{\pi k(L-1)}{N}} \frac{\text{sen}\left(\frac{\pi k L}{N}\right)}{\text{sen}\left(\frac{\pi k}{N}\right)}$$

con magnitud

$$|X(\omega_k)| = \frac{\text{sen}\left(\frac{\pi k L}{N}\right)}{\text{sen}\left(\frac{\pi k}{N}\right)} \quad ; \quad k = 0, 1, 2, \dots, N-1$$

y la fase

$$\phi(\omega_k) = -\frac{\pi k}{N}(L-1) + \text{ang} \left(\frac{\text{sen}\left(\frac{\pi k L}{N}\right)}{\text{sen}\left(\frac{\pi k}{N}\right)} \right)$$

Ejemplo 7.3

A continuación se muestra un programa elaborado para ser ejecutado en Matlab, el programa calcula la magnitud de la DFT y la Fase. Para verificar los resultados se corre el programa para dos señales un escalón y una señal seno que se muestran en las figuras 7.8 y 7.9 respectivamente, en estas figuras se puede apreciar la señal original, su espectro y su fase. Se hace la aclaración que por la gran cantidad de puntos utilizados en ambos ejemplos la gráficas se muestran en forma continua.

```

%
%   TRANSFORMADA DISCRETA DE FOURIER
%
%
clear;
clf;
load senall.dat;           % Lee una señal de un archivo de entrada
x= senall;
N = length(x);
pi = 3.14159;
for k=1:N
    A = 0;
    B = 0;
    for n=1:N
        A=A+x(n)*cos(2*pi*(k-1)*(n-1)/N);
        B=B+x(n)*sin(2*pi*(k-1)*(n-1)/N);
    end
    Magx(k)= sqrt(A^2+ B^2);   % Calcula la Magnitud de X(k)
    Angx(k)= atan2(-B,A);     % Calcula la fase de X(k)
end

%%   Graficación de resultados
subplot(311), plot(x);
title('Señal de entrada');
subplot(312), plot(Magx);
title('Magnitud del Espectro de la Señal');
grid;
subplot(313), plot(Angx);
title('Fase de la Señal de entrada');
grid;

```

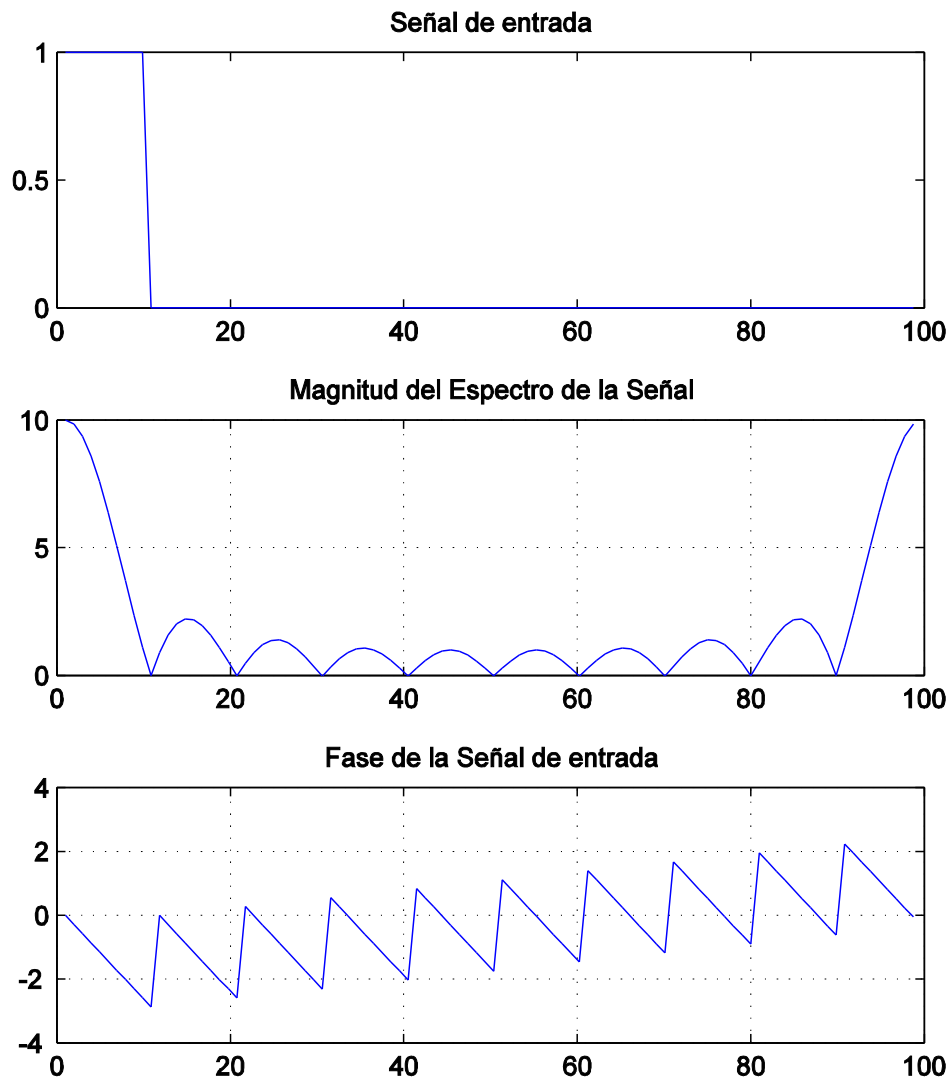


Figura 7.8 Aplicación de la DFT para una señal pulso $p(t)=U(n)-U(n-11)$ para $N=100$.

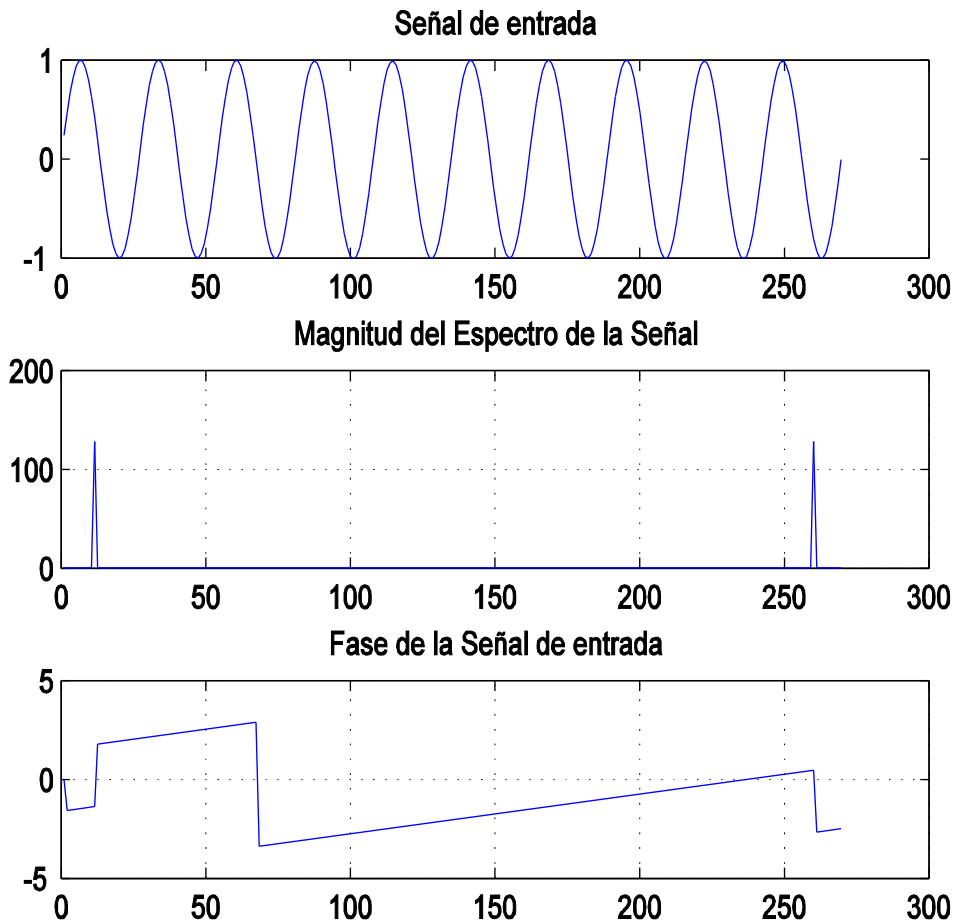


Figura 7.9 La DFT de una señal senoidal de 10 ciclos en N=256 puntos

7.8 ALGORITMO DE GOERTZEL

Se utiliza para el cálculo de la DFT evitando efectuar muchos cálculos con números complejos, este algoritmo explota la periodicidad del factor W_N^k y permite expresar a la DFT como la operación de un filtro lineal de Respuesta Infinita al Impulso (IIR).

Dado que el factor $W_N^{-kN} = 1$ (7.76)

entonces multiplicando a la definición de la DFT por el factor de (7.76)

$$X(k) = W_N^{-kN} X(k) = \sum_{m=0}^{N-1} x(m)W_N^{-k(N-m)} \tag{7.77}$$

la ecuación (7.77) tiene la forma de una convolución lineal.

Si $n=N$ definimos:

$$y_k(n) = \sum_{m=0}^{N-1} x(m)W_N^{-k(n-m)} \quad (7.78)$$

donde $y_k(n)$ es la convolución de una secuencia finita $x(n)$ de longitud N con la respuesta al impulso de un filtro

$$h(n) = W_N^{-kn}U(n) \quad (7.79)$$

aplicando la TZ a la ecuación (7.79) obtenemos la función de transferencia

$$H_K(z) = \frac{1}{1 - W_N^{-k}z^{-1}} \quad (7.80)$$

la ecuación (7.80) se puede interpretar como la salida de un filtro IIR a tiempo $n=N$ que produce el valor de la DFT de $x(n)$ a la frecuencia $\omega_K = \frac{2\pi k}{N}$, es decir, que la DFT de $x(n)$ es $X(k) = y_k(n)|_{n=N}$

es evidente que $H_K(z)$ tiene un polo sobre el círculo unitario a la frecuencia $\omega_k = \frac{2\pi k}{N}$

Multiplicando $H_K(z)$ por $1 - W_N^K z^{-1}$ arriba y abajo de (7.80), se obtiene (7.81)

$$H_K(z) = \frac{1 - W_N^K z^{-1}}{(1 - W_N^{-k} z^{-1})(1 - W_N^K z^{-1})} = \frac{1 - W_N^K z^{-1}}{1 - 2 \cos\left(\frac{2\pi k}{N}\right)z^{-1} + z^{-2}} = \frac{1 - W_N^K z^{-1}}{1} \cdot \frac{1}{1 - 2 \cos\left(\frac{2\pi k}{N}\right)z^{-1} + z^{-2}} \quad (7.81)$$

Es decir, que la DFT puede calcularse introduciendo un bloque de N datos de entrada a un banco paralelo de "N" filtros simples (resonadores figura 7.10), donde cada filtro tiene un polo a la frecuencia correspondiente ω_k de la DFT, en vez de efectuar la DFT vía la convolución, se pueden utilizar ecuaciones en diferencias que evitan las multiplicaciones y sumas complejas al combinar dos resonancias con polos conjugados complejos. Para calcular la salida $y_k(n)$ se reordena la ecuación (7.81) y se separa en dos funciones de transferencia en cascada $H_1(z)$ y $H_2(z)$ como en las ecuaciones (7.82) y (7.83)

$$H(z) = H_1(z)H_2(z) = \left[\frac{Y(z)}{V(z)} \right] \left[\frac{V(z)}{X(z)} \right] \quad (7.82)$$

donde

$$H_1(z) = \frac{Y(z)}{V(z)} = 1 - W_N^K z^{-1} \quad (7.83)$$

$$H_2(z) = \frac{V(z)}{X(z)} = \frac{1}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}} \quad (7.84)$$

definiendo condiciones iniciales $y_k(-1)=0$ y $v_k(-1)=v_k(-2)=0$, aplicando la TZI a las ecuaciones (7.83) y (7.84) y despejando las variables $v_k(n)$ y $y_k(n)$

$$v_k(n) = X(n) + 2\cos\left(\frac{2\pi k}{N}\right)v_k(n-1) - v_k(n-2) \quad (7.85)$$

$$y_k(n) = v_k(n) - W_N^k v_k(n-1) \quad (7.86)$$

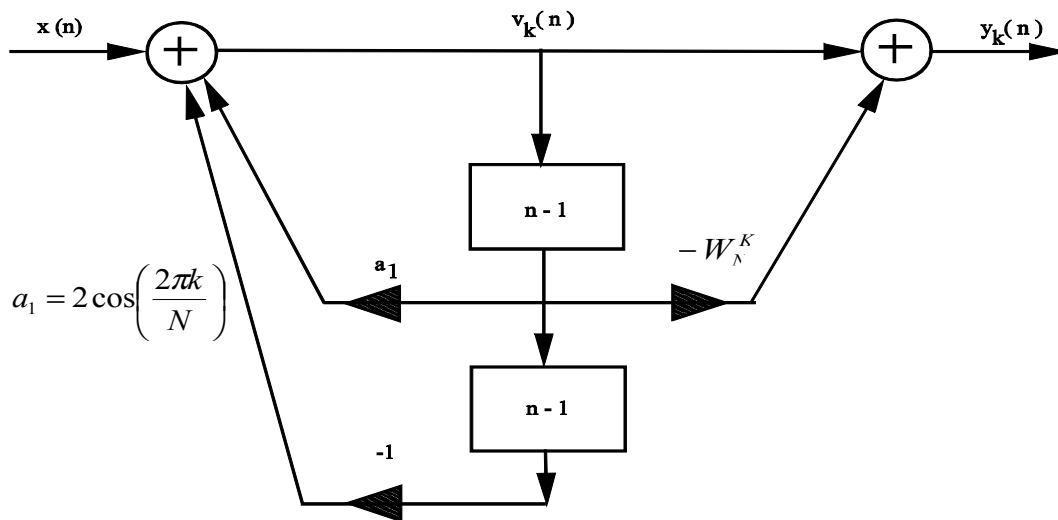


Figura 7.10 Diagrama del Algoritmo de Goertzel

Es importante hacer notar que la salida deseada $y_k(n)$ realmente es la transformada discreta de la entrada $x(n)$, es decir que $y_k(n=N) = X(k)$. Cada valor de salida $X(k)$ requiere $2N+2$ sumas reales y $N+2$ multiplicación reales, es decir que en total se requieren $2(N+1)N$ sumas reales y $N(N+2)$ multiplicaciones reales, en el caso de la DFT se tiene que realizar N^2-N sumas y N^2 productos complejos, es decir que este algoritmo reduce las operaciones en casi un 50%.

La conclusión para el algoritmo de Goertzel es que la DFT completa puede calcularse pasando un bloque de datos de entrada a un banco de N filtros de un polo simple en paralelo (resonadores), donde cada filtro tiene un polo en la frecuencia correspondiente de la DFT $\omega_k=2\pi k/N$.

Ejemplo 7.4

Programa en Matlab para realizar el algoritmo de Goertzel.

```

%
%   ALGORITMO DE GOERTZEL
%
clear;
clf;
load senall.dat   % Lee señal de un archivo
x= senall;
N = length(x);
pi = 3.14159;
for k=1:N
    v1 = 0;   % Condiciones iniciales cero
    v2 = 0;
    clear n;
    for n=1:N
        vk = x(n)+ v1*2*cos(2*pi*(k-1)/N) -v2;
        v2 = v1; % Actualiza variable v2
        v1 = vk;
    end
    yr = v2 - v2*cos(2*pi*(k-1)/N);
    yi = v1*sin(2*pi*(k-1)/N);
    Magx(k)= sqrt(yr^2+ yi^2);
end
    subplot(211), plot(x);
    title('Señal de entrada');
    subplot(212), plot(Magx);
    title('Magnitud del Espectro de la Señal');
    grid;

```

7.9 RESUMEN

En este capítulo se ha introducido el análisis espectral de señales discretas, analizando desde varios puntos de vista la transformada discreta de Fourier, su interpretación gráfica y analítica, se presentó el algoritmo de Goertzel que calcula la DFT como un sistema recursivo y reduce el número de operaciones matemáticas. También se han mostrado ejemplos de programas que permiten calcular la DFT con el fin de hacer más ilustrativos los algoritmos.

7.10 PROBLEMAS PROPUESTOS

1. Demostrar todas las propiedades de la DFT expuestas en este capítulo.

2. Modificar el programa de la DFT para calcular la DFTI.

CAPITULO 8

TRANSFORMADA RÁPIDA DE FOURIER

El cálculo del espectro de una señal discreta o la respuesta en frecuencia discreta de un sistema es de gran importancia en el procesamiento digital de señales (PDS), ya que a través de los espectros podemos analizar características de las señales o los sistemas, obtener información, analizar las señales y sistemas, etc. En el capítulo anterior se estudió la transformada Discreta de Fourier (DFT) que nos permite calcular el espectro discreto de una señal o sistema discreto, sin embargo, el cálculo de la DFT involucra cálculos matemáticos intensivos del orden de N^2 operaciones complejas haciendo a veces imposible su cálculo para procesos en tiempo real. Por tanto los investigadores que se dedican al PDS han desarrollado algoritmos rápidos que permitan reducir el número de operaciones. Para el caso del cálculo del espectro de una señal discreta se desarrolló la transformada rápida de Fourier (FFT) que explota al máximo las propiedades de simetría y periodicidad de la DFT, la FFT como se verá en este capítulo, reduce las operaciones hasta en un 5% manteniendo el desempeño en el resultado.

En este capítulo se desarrollará la FFT “radix” dos en diferentes enfoques, se muestran algunos ejemplos ilustrativos y al final se presenta un ejemplo desarrollado en lenguaje Matlab y ensamblador del DSP TMS320C50 de Texas Instruments.

8.1 DESARROLLO DE LA FFT

Como se explica en seguida, en esencia la FFT se desarrolla a partir de la definición de la DFT, la FFT es una forma eficiente de calcular la DFT y su éxito se debe a la reducción del número de adiciones y multiplicaciones requeridas para su cálculo, explotando las propiedades de simetría y periodicidad de la W_N^k DFT y los factores

Por simetría

$$W_N^{\left(k+\frac{N}{2}\right)} = -W_N^k \quad \text{y} \quad X(k) = X(N-k) \quad (8.1)$$

Por periodicidad

$$W_N^{(k+N)} = W_N^k \quad \text{y} \quad X(k) = X(k+N) \quad (8.2)$$

DESARROLLO MATRICIAL

Dada una secuencia discreta $x(n)$ de longitud $N=4$ y desarrollando las operaciones de la definición de ecuación la DFT se tiene

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad (8.3)$$

donde $k = 0, 1, 2, 3$ y $W = e^{-j\frac{2\pi}{N}}$

se obtiene una función en variable k dada por la ecuación (8.4)

$$X(k) = x(0) W_N^0 + x(1) W_N^k + x(2) W_N^{2k} + x(3) W_N^{3k} \quad (8.4)$$

desarrollando la ecuación (8.4) para $k=0,1,2,3$, se genera el conjunto de cuatro ecuaciones lineales con coeficientes complejos dados por

$$\begin{aligned} X(0) &= x(0) W^0 + x(1) W^0 + x(2) W^0 + x(3) W^0 \\ X(1) &= x(0) W^0 + x(1) W^1 + x(2) W^2 + x(3) W^3 \\ X(2) &= x(0) W^0 + x(1) W^2 + x(2) W^4 + x(3) W^6 \\ X(3) &= x(0) W^0 + x(1) W^3 + x(2) W^6 + x(3) W^9 \end{aligned} \quad (8.5)$$

(por simplicidad se ha eliminado el subíndice N de los factores W)

Se observa de las ecuaciones (8.5) que se puede expresar en forma matricial de la forma $X_N(k) = W_N X_N(n)$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & W^4 & W^6 \\ 1 & W^3 & W^6 & W^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (8.6)$$

donde W_N es una matriz simétrica de la transformación lineal y si existe su inversa, se cumple que $x_N(n) = W_N^{-1} X_N(k)$, es decir, la Transformada Discreta de Fourier Inversa (IDFT), y se puede expresar como $x_N(n) = W_N^* X_N(k) / N$ donde $W_N^{-1} = W_N^* / N$, lo que implica que $W_N^* W_N = N I_N$, donde I_N es la matriz identidad de orden N . Por lo tanto, W_N es una matriz ortogonal unitaria y se conoce como Matriz de Fourier.

Escribiendo los valores numéricos correspondientes a los factores “Twiddle” W_N^k

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (8.7)$$

factorizando la matriz W_N

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (8.8)$$

La ecuación (8.8) se puede escribir como dos ecuaciones matriciales independientes (8.9) y (8.10)

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} \quad (8.9)$$

$$\begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \tag{8.10}$$

donde $x_1(0)$, $x_1(1)$, $x_1(2)$, $x_1(3)$ es una secuencia de puntos intermedios del proceso, es decir que se tienen las ecuaciones lineales reducidas de (8.9)

$$\begin{aligned} X(0) &= x_1(0) + x_1(2) \\ X(1) &= x_1(1) - jx_1(3) \\ X(2) &= x_1(0) - x_1(2) \\ X(3) &= x_1(1) + jx_1(3) \end{aligned} \tag{8.11}$$

de (8.10)

$$\begin{aligned} x_1(0) &= x(0) + x(2) \\ x_1(1) &= x(0) - x(2) \\ x_1(2) &= x(1) + x(3) \\ x_1(3) &= x(1) - x(3) \end{aligned} \tag{8.12}$$

de las ecuaciones (8.11) y (8.12) se deduce que para el cálculo de $x_1(k)$ sólo son necesarias ocho sumas, lo que implica una reducción de la cantidad de operaciones si se realizara la DFT para $N=4$. La ejecución de operaciones de las ecuaciones de la DFT se observan en la figura 8.1 y la ejecución de la FFT se ilustra en la figura 8.2

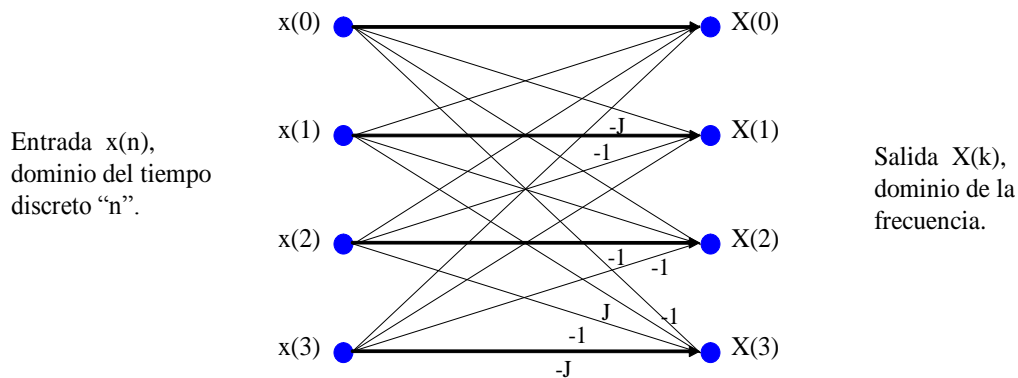


Figura 8.1 Diagrama de operaciones de la DFT para $N=4$

En la figura 8.1 se observa que para cada salida de $X(k)$ se utilizan todas las entradas de $x(n)$ multiplicadas por su respectiva constante, lo que ocasiona que se realicen cuatro multiplicaciones por cada salida $X(k)$.

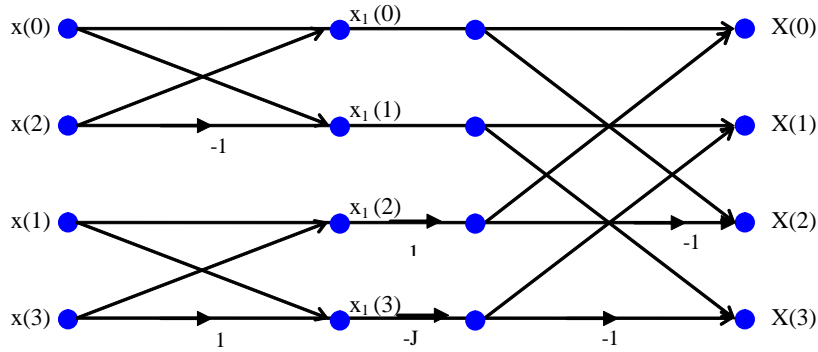
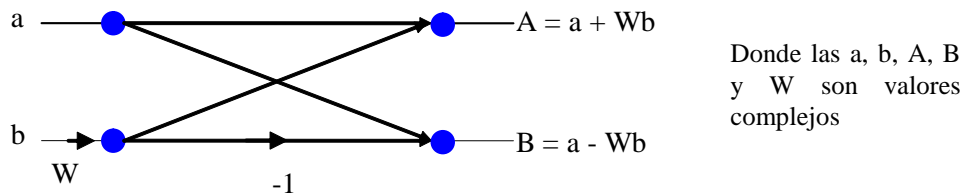


Figura 8.2 Diagrama de operaciones de la FFT para $N=4$

En la figura 8.2 se puede observar que el número de operaciones se ha reducido cuando se opera con la FFT. En el diagrama de la FFT (figura 8.2), obsérvese que cada punto donde convergen las flechas, es un punto de suma; las entradas no están ordenadas, a la forma de ordenar $x(n)$ a la entrada de la FFT se le conoce como decimación en el tiempo, mientras que la salidas $X(k)$ está en orden natural. Además en el cálculo de la FFT existe una operación básica llamada mariposa (Butterfly), que se repite muchas veces y se ilustra en la figura 8.3.



Donde las a , b , A , B y W son valores complejos

Figura 8.3 La mariposa, operación básica de la FFT

8.2 DESARROLLO DE LA FFT RADIX 2 A PARTIR DE LA DEFINICIÓN DE LA DFT

Recordando la definición de la DFT para una secuencia discreta $x(n)$ de longitud N

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}} \quad k = 0, 1, 2, \dots, N-1 \quad (8.13)$$

La FFT se puede desarrollar partiendo de (8.13) basándose en el principio de “*divide y vencerás*”, esto es, dada una señal discreta $x(n)$ de longitud N (número no primo), expresada como un vector

$$x_N^T[n] = \underbrace{[x(0), x(1), x(2)]}_L, \underbrace{[x(3), x(4), x(5)]}_L, \dots, \underbrace{\dots, x(N-1)}_L \quad (8.14)$$

este vector se puede dividir en M partes de longitud L , es decir, $N = ML$; el almacenamiento de $x(n)$ se expresa en forma de columnas ($n = \ell + Lm$):

$m \backslash \ell$	0	1	2	3	...	$M-1$
0	$x(0)$	$x(L)$	$x(2L)$	$x(3L)$...	$x((M-1)L)$
1	$x(1)$	$x(L+1)$	$x(2L+1)$	$x(3L+1)$...	\vdots
2	$x(2)$	$x(L+2)$	$x(2L+2)$	$x(3L+2)$...	\vdots
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$L-1$	$x(L-1)$	$x(2L-1)$	$x(3L-1)$	$x(4L-1)$...	$x(N-1)$

es decir, M columnas de longitud L .

Ordenando en forma de filas ($n = M\ell + m$) se tiene

$\ell \backslash m$	0	1	2	3	...	M-1
0	x(0)	x(1)	x(2)	x(3)	...	x(M-1)
1	x(M)	x(M+1)	x(M+2)	x(M+3)	...	x(2M-1)
2	x(2M)	x(2M+1)	x(2M+2)	x(3M+3)	...	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
L-1	x((L-1)M)	x((L-1)M+1)	x((L-1)M+2)	x((L-1)M+3)	...	x(N-1)

es decir, L filas de longitud M. Desarrollando la FFT a partir de la DFT para la matriz ordenada de M columnas de longitud L

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} = \sum_{n=0}^{L-1} x(n) W_N^{kn} + \sum_{n=L}^{2L-1} x(n) W_N^{kn} + \cdots + \sum_{n=(M-1)L}^{N-1} x(n) W_N^{kn} \quad (8.15)$$

para una salida ordenada en forma matricial

$$X(p, q) = \sum_{m=0}^{M-1} \sum_{\ell=0}^{L-1} x(\ell, m) W_N^{(\ell+mL)(q+pM)} \quad (8.16)$$

desarrollando el exponente W_N

$$W_N^{(\ell+mL)(q+pM)} = W_N^{\ell q} W_N^{\ell p M} W_N^{m L q} W_N^{m L p M} \quad (8.17)$$

$$W_N^{m p M} = \left(e^{-j \frac{2\pi}{N}} \right)^{m p M} = 1 \Rightarrow W_N^{\ell p M} = W_L^{\ell p} = e^{-j \frac{2\pi \ell p}{M}} \Rightarrow L = \frac{N}{M} \quad (8.18)$$

$$W_N^{m q L} = e^{-j \frac{2\pi}{N} m q L} = e^{-j \frac{2\pi m q}{L}} = e^{-j \frac{2\pi m q}{M}} = W_M^{m q} \quad (8.19)$$

sustituyendo (8.17), (8.18) y (8.19) en (8.16)

$$X(p, q) = \sum_{m=0}^{M-1} \sum_{\ell=0}^{L-1} x(\ell, m) W_M^{m q} W_L^{\ell p} W_N^{\ell q} = \sum_{\ell=0}^{L-1} \left\{ W_N^{\ell q} \sum_{m=0}^{M-1} x(\ell, m) W_M^{m q} \right\} W_L^{\ell p} \quad (8.20)$$

La ecuación (8.20) implica la ejecución de los siguientes pasos

1. Calcular "L" DFTs de "M" puntos cada una, es decir

$$F[\ell, q] = \sum_{m=1}^{M-1} x(\ell, m) W_M^{mq} ; \begin{cases} \ell = 0, 1, 2, 3, \dots, L-1 \\ 0 \leq q \leq M-1 \end{cases} \quad (8.21)$$

Lo que implica realizar LM^2 multiplicaciones complejas y $LM(M-1)$ sumas complejas.

2. Calcular el siguiente arreglo

$$G[\ell, q] = W_N^{\ell q} F[\ell, q] ; \begin{cases} \ell = 0, 1, 2, 3, \dots, L-1 \\ q = 0, 1, 2, 3, \dots, M-1 \end{cases} \quad (8.22)$$

que requiere LM multiplicaciones complejas.

3. Calcular “M” DFT’s de “L” puntos cada una

$$X[\ell, q] = \sum_{\ell=0}^{L-1} G[\ell, q] W_L^{\ell p} ; \begin{cases} p = 0, 1, 2, 3, \dots, L-1 \\ q = 0, 1, 2, 3, \dots, M-1 \end{cases} \quad (8.23)$$

Implica ML^2 multiplicaciones complejas y $ML(L-1)$ sumas complejas.

En resumen se efectúan $N(M+L+1)$ multiplicaciones complejas y $N(M+L-2)$ sumas complejas, donde $N=ML$. Es decir, que se han reducido las operaciones de N^2 a $N(M+L+1)$ multiplicaciones complejas y de $N(N-1)$ a $N(M+L-2)$ sumas complejas.

8.2.1 ALGORITMO FFT RADIX 2

Dividiendo “N” en “m” factores “r” $N = r_1 r_2 r_3 r_4 \cdots r_m$, los factores r_j son números primos si $r_1 = r_2 = r_3 = r_4 = \cdots = r_m = r$, entonces $N = r^m$ en cuyo caso las DFT’s serán de tamaño r , de tal forma que el cálculo de la DFT de N puntos tiene una forma regular. A “r” se le conoce como el “radix” del algoritmo de la FFT. Para $N = 2^m$ ($r = 2$) se tiene el algoritmo FFT radix 2, con $N = LM$ lo que implica que $M = \frac{N}{2}$ ó $L = \frac{N}{2}$. Ahora se substituyen estos resultados en la ecuación general (8.23), si $L=2$ y $M=N/2$

$$X[p, q] = \sum_{\ell=0}^{2-1} \left\{ W_N^{\ell q} \left[\sum_{m=0}^{\frac{N}{2}-1} X[\ell, m] W_{\frac{N}{2}}^{mq} \right] \right\} W_2^{\ell p} \quad (8.24)$$

almacenando el arreglo $x(n)$ en forma de filas

$$\begin{array}{l}
 n = \ell + mL = \ell + 2m \\
 k = Mp + q = \frac{N}{2}p + q
 \end{array}
 \begin{array}{|c|cccc|}
 \hline
 & m & & & & \\
 \hline
 \ell & \backslash & 0 & 1 & 2 & \dots & \frac{N}{2}-1 \\
 \hline
 0 & & x(0) & x(2) & x(4) & \dots & x(N-2) & \leftarrow f_1(n) = x(2n) \\
 1 & & x(1) & x(3) & x(5) & \dots & x(N-1) & \leftarrow f_2(n) = x(2n+1) \\
 \hline
 \end{array}$$

se observa que las secuencias $f_1(n)$ y $f_2(n)$ son la parte par e impar de la secuencia $x(n)$ respectivamente.

Obteniendo la DFT de $x(n)$ como la suma de las DFT's de $f_1(n)$ y $f_2(n)$

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} = \sum_{m=0}^{\frac{N}{2}-1} x(2m) W_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x(2m+1) W_N^{(2m+1)k} \tag{8.25}$$

como $W_N^2 = W_{\frac{N}{2}}$ $W_N = e^{-j\frac{2\pi}{N}}$ $W_N^2 = e^{-j\frac{2\pi}{N}(2)} = e^{-j\frac{2\pi}{\frac{N}{2}}} = W_{\frac{N}{2}}$

$$X(k) = \sum_{m=0}^{\frac{N}{2}-1} x(2m) W_{\frac{N}{2}}^{mk} + W_N^k \sum_{m=0}^{\frac{N}{2}-1} x(2m+1) W_{\frac{N}{2}}^{mk} \tag{8.26}$$

$$X(k) = \sum_{m=0}^{\frac{N}{2}-1} f_1(m) W_{\frac{N}{2}}^{mk} + W_N^k \sum_{m=0}^{\frac{N}{2}-1} f_2(m) W_{\frac{N}{2}}^{mk} \tag{8.27}$$

que se puede expresar como series de DFT's de longitud $N/2$

$$X(k) = F_1(k) + W_N^k F_2(k) \quad ; \quad k = 0, 1, 2, \dots, \frac{N}{2}-1 \tag{8.28}$$

donde se ha definido

$$F_1(k) = DFT\{f_1(n)\} \tag{8.29}$$

$$F_2(k) = DFT\{f_2(n)\} \tag{8.30}$$

que se calculan para $\frac{N}{2}$ puntos. Como $F_1(k)$ y $F_2(k)$ son funciones periódicas en $\frac{N}{2}$, entonces

$$F_1(k + \frac{N}{2}) = F_1(k) \tag{8.31}$$

$$F_2(k + \frac{N}{2}) = F_2(k) \tag{8.32}$$

Además se puede demostrar que W_N es simétrico

$$W_N^{k+\frac{N}{2}} = -W_N^k = e^{-j\frac{2\pi k}{N}} \underbrace{e^{-j\frac{2\pi}{N} \frac{N}{2}}}_{-1} = -e^{-j\frac{2\pi k}{N}} \tag{8.33}$$

entonces $X(k) = F_1(k) + W_N^k F_2(k)$; $k = 0, 1, 2, \dots, \frac{N}{2} - 1$

$$X(k + \frac{N}{2}) = F_1(k + \frac{N}{2}) - W_N^k F_2(k + \frac{N}{2}) ; k = 0, 1, 2, \dots, \frac{N}{2} - 1 \tag{8.34}$$

$$X(k + \frac{N}{2}) = F_1(k) - W_N^k F_2(k) ; k = 0, 1, 2, \dots, \frac{N}{2} - 1 \tag{8.35}$$

gráficamente se expresa como se ilustra en la figura 8.4

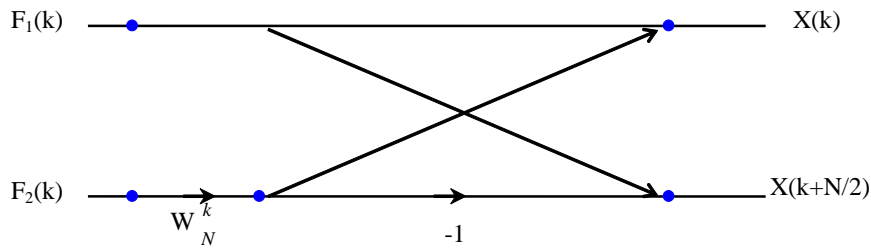


Figura 8.4 Operación mariposa

Número de operaciones complejas

$$2\left(\frac{N}{2}\right)^2 = \left(\frac{N^2}{2}\right) \text{ para el cálculo de } F_1(k) \text{ y } F_2(k) \tag{8.36}$$

Para el cálculo de $X(k)$ se requieren $\frac{N^2}{2}$ más $\frac{N}{2}$ del proceso $W_N^k F_2(k)$, es decir, $\frac{N^2}{2} + \frac{N}{2}$ multiplicaciones, aproximadamente la mitad si se calcula la DF directa de $x(n)$.

$$\tag{8.37}$$

Si

$$G_1(k) = F_1(k) \quad ; \quad k = 0, 1, 2, \dots, \frac{N}{2} - 1$$

y

$$G_2(k) = F_2(k) W_N^k \quad (8.38)$$

La DFT de N puntos se expresa como

$$X(k) = G_1(k) + G_2(k) \quad (8.39)$$

o

$$X(k + \frac{N}{2}) = G_1(k) - G_2(k) \quad (8.40)$$

Si se continúa el proceso de dividir las secuencias en una parte par y otra impar, entonces:

$$\text{para } f_1(n) \text{ se tiene } v_{11}(n) = f_1(2n) \text{ y } v_{12}(n) = f_1(2n+1) \quad ; \quad n = 0, 1, 2, \dots, \frac{N}{4} - 1$$

$$\text{para } f_2(n) \text{ se tiene } v_{21}(n) = f_2(2n) \text{ y } v_{22}(n) = f_2(2n+1)$$

entonces se calculan las DFT's para $\frac{N}{4}$ puntos de las secuencias $v_{ij}(n)$.

Haciendo el desarrollo similar a las ecuaciones (8.26) a (8.40) para $\frac{N}{2}$

$$F_1(k) = V_{11}(k) + W_{\frac{N}{2}}^k V_{12}(k) \quad ; \quad k = 0, 1, 2, \dots, \frac{N}{4} - 1 \quad (8.41)$$

$$F_1(k + \frac{N}{4}) = V_{11}(k) - W_{\frac{N}{2}}^k V_{12}(k) \quad (8.42)$$

$$F_2(k) = V_{21}(k) + W_{\frac{N}{2}}^k V_{22}(k) \quad (8.43)$$

$$F_2(k + \frac{N}{4}) = V_{21}(k + \frac{N}{4}) - W_{\frac{N}{2}}^k V_{22}(k) \quad (8.44)$$

Para este último caso, se efectúan $4(\frac{N}{4})^2 + 2(\frac{N}{4})$ multiplicaciones complejas, es decir, $\frac{N^2}{4} + \frac{N}{2}$ operaciones, y adicionalmente se requieren $\frac{N}{2}$ multiplicaciones complejas para el cálculo $F_1(k)$ y $F_2(k)$, entonces se tienen $\frac{N^2}{4} + N$ multiplicaciones. El proceso de dividir las secuencias $v_{ij}(n)$ se puede seguir hasta que la última secuencia quede de longitud 2 (el “radix” del algoritmo). Para $N = 2^m$ implica que $m = \log_2 N$ donde “m” es el número de veces que se hace las división de secuencias, a este proceso se le conoce como decimación de $x(n)$ en el tiempo. El número de operaciones complejas a realizar en el cálculo de la FFT “radix” 2 de longitud N es de: $(N/2) \log_2 N$ multiplicaciones y de $N \log_2 N$ sumas; en estas dos últimas expresiones es muy claro la reducción del número de operaciones del algoritmo.

Ejemplo 8.1

Si $x(n)$ es una secuencia de 8 puntos, es decir, $N = 8 = 2^3$; para realizar la decimación de $x(n)$ se divide en subsecuencias índice par e impar hasta que la longitud de las subsecuencias sea de longitud dos (el “radix” del algoritmo), este procedimiento se muestra en la figura 8.5

$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	$x(7)$
$x(0)$	$x(2)$	$x(4)$	$x(6)$	$x(1)$	$x(3)$	$x(5)$	$x(7)$
$x(0)$	$x(4)$	$x(2)$	$x(6)$	$x(1)$	$x(5)$	$x(3)$	$x(7)$

Figura 8.5 Decimación de una secuencia de longitud $N=8$

Para el cálculo de la FFT se deben calcular:

- 4 DFT's de 2 puntos
- 2 DFT's de 4 puntos
- 1 DFT de 8 puntos
- $m = \log_2 N = 3$, número de etapas del algoritmo

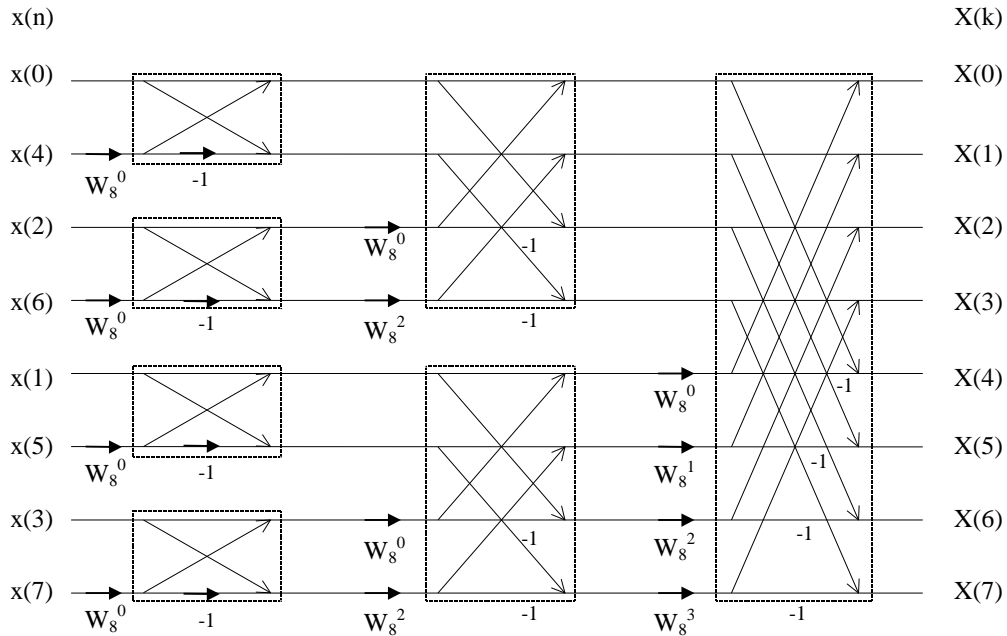


Figura 8.6 FFT de longitud $N=8$

Como se observa en la figura 8.6, el algoritmo de la FFT está en función de “ n ” ($N = 2^n$), además se utilizó sólo la mitad de los factores “Twiddle” y es aquí donde se aprovecha la simetría de la DFT; también se observa que se utiliza la mitad de los coeficientes W_N hasta la

última etapa, en las etapas intermedias se utilizan alternados. Se puede verificar que si se sabe construir un diagrama de mariposas con $N=4$, es posible continuar con uno de $N=8$, para $N=16$, y así sucesivamente para cualquier orden superior N .

8.3 DECIMACIÓN

La decimación de una secuencia consiste en darle un orden especial a la secuencia. Esta se puede efectuar de varias formas:

1. Dada una secuencia de longitud $N = 2^n$, se va subdividiendo en secuencias con índice par e impar hasta que tengan secuencias de longitud 2 (ver ejemplo anterior).
2. Girar los bits del índice respecto del centro del número total de bits, como ejemplo en la figura 8.7 se ilustra el giro de los bits para los índices binarios con $N=8$

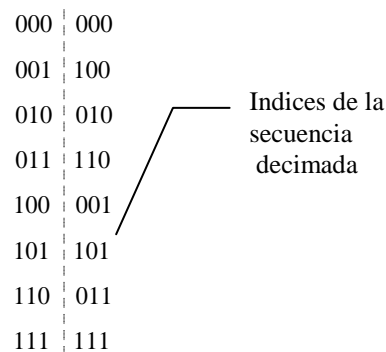


Figura 8.7 Decimación girando los bits del índice

3. Direccionamiento por “*acarreo inverso*”

Algunos DSP's modernos tienen este tipo de direccionamiento. Se comienza desde el número “0” en forma binaria y se le suma $\frac{N}{2}$ (binario) efectuando las sumas de izquierda a derecha y arrastrando el “*carry*” a la derecha, entonces se obtiene el siguiente índice decimado, a éste número se le suma de nuevo $\frac{N}{2}$, el procedimiento continúa de forma similar hasta obtener un resultado de sólo unos en n bits. Por ejemplo, si $N = 8$ entonces $\frac{N}{2} = 4$, ver figura 8.8

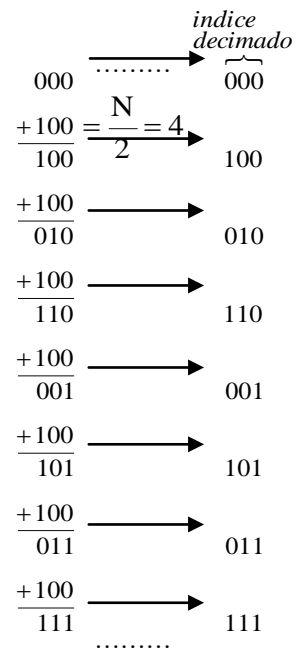


Figura 8.8 Decimación por acarreo inverso

4. Decimación por hardware:

Para direccionar en forma decimada una memoria que contenga los datos $x(n)$ de longitud “N”, basta con conectar un buffer de tres estados cuyas direcciones están giradas respecto del centro de la longitud de palabra (figura 8.9).

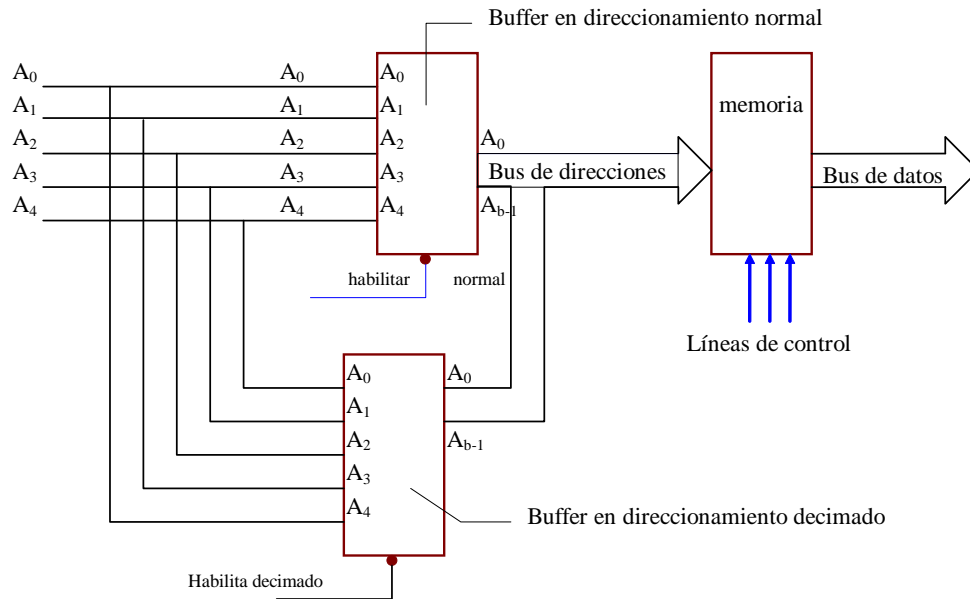


Figura 8.9 Decimación por hardware

Ejemplo 8.2

Dada la secuencia $x(n) = \{1, 1, 1, 1, 0, 0, 0, 0\}$, obtener $X(k)$ utilizando la FFT. En la figura 8.9 se observa el diagrama de la FFT para $N=8$, además se escriben en el diagrama los coeficientes W_8 correspondientes. Los cálculos se efectúan manualmente siguiendo las líneas del diagrama y realizando las operaciones mariposa en cada caso.

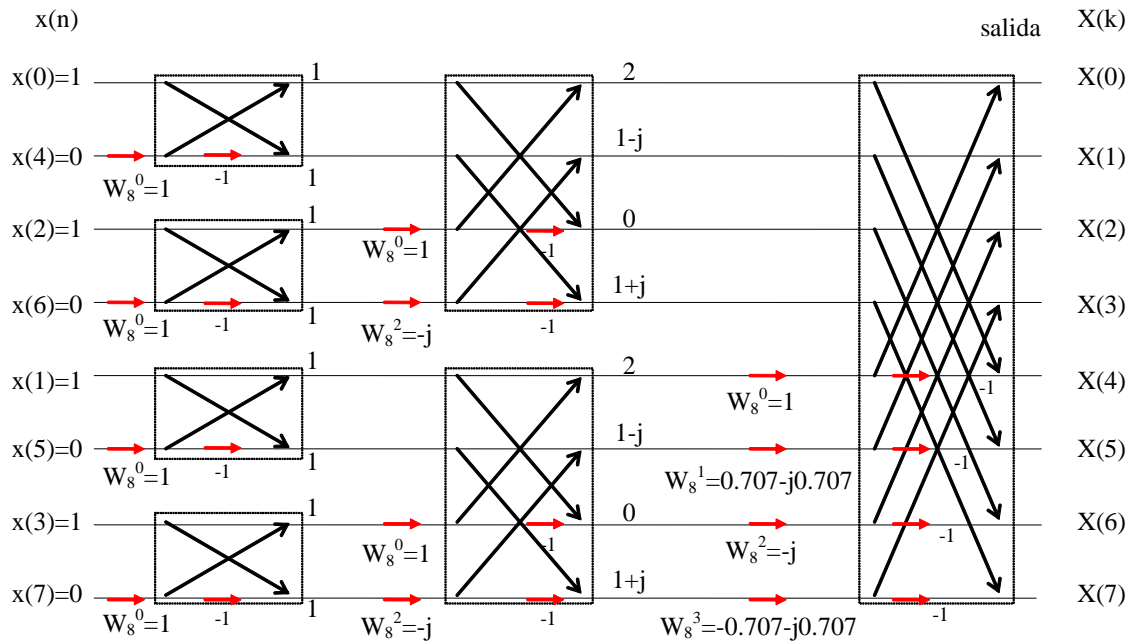


Figura 8.10 FFT para N=8

Valores calculados de X(k):

- $X(0) = 4$
- $X(1) = 2.605 \angle -67.5^\circ$
- $X(2) = 0$
- $X(3) = 1.082 \angle -22.5^\circ$
- $X(4) = 0$
- $X(5) = 1.082 \angle 22.5^\circ$
- $X(6) = 0$
- $X(7) = 2.609 \angle 67.46^\circ$

8.4 DESARROLLO ALTERNATIVO DE LA FFT

Aplicando la definición de la DFT sobre una secuencia $x_0(n)$

$$X(k) = \sum_{n=0}^{N-1} x_0(n) e^{-\frac{j2\pi kn}{N}}; \quad k = 0, 1, 2, \dots, N-1 \quad (8.45)$$

si $W = e^{-\frac{j2\pi}{N}}$ entonces la ecuación (8.45) se puede reescribir

$$X(k) = \sum_{n=0}^{N-1} x_0(n) W^{nk}; \quad k = 0, 1, 2, \dots, N-1 \quad (8.46)$$

como $N = 2L$, desarrollando para el caso en que $N = 4$, entonces $L = 2$, por lo que se puede representar k y n como

$$\begin{aligned} k &= 0, 1, 2, 3 \quad \text{ó} \quad k = (k_1, k_0) = 00, 01, 10, 11 \\ n &= 0, 1, 2, 3 \quad \text{ó} \quad n = (n_1, n_0) = 00, 01, 10, 11 \end{aligned}$$

otra forma de escribir k y n es $k = 2k_1 + k_0$ y $n = 2n_1 + n_0$, donde k_0, k_1, n_0 y n_1 sólo pueden tomar valores entre 0 y 1, reescribiendo $X(k)$ como función de k_1 y k_0 , se tiene

$$X(k_1, k_0) = \sum_{n_0=0}^1 \sum_{n_1=0}^1 x_0(n_1, n_0) W^{(2n_1+n_0)(2k_1+k_0)} \quad (8.47)$$

donde

$$\begin{aligned} W^{(2n_1+n_0)(2k_1+k_0)} &= W^{(2n_1+n_0)2k_1} W^{(2n_1+n_0)k_0} \\ &= W^{4n_1k_1} W^{2n_0k_1} W^{(2n_0+n_1)k_0} \\ &= W^{2n_0k_1} W^{(2n_0+n_1)k_0} \end{aligned} \quad (8.48)$$

ya que

$$W^{4n_1k_1} = [W^4]^{n_1k_1} = \left[e^{-\frac{j2\pi 4}{4}} \right]^{n_1k_1} = 1 \quad (8.49)$$

entonces

$$X(k_1, k_0) = \sum_{n_0=0}^1 \left[\sum_{n_1=0}^1 x_0(n_1, n_0) W^{2k_0n_1} \right] W^{(2k_1+k_0)n_0} \quad (8.50)$$

La ecuación (8.50) es el fundamento del algoritmo de la FFT. Considérese ahora cada una de las sumatorias de manera individual, cuando la sumatoria interior toma valores de n_0 y k_0

$$X_1(k_0, n_0) = \sum_{n_1=0}^1 x_0(n_1, n_0) W^{2k_0 n_1} \quad (8.51)$$

al desarrollar la ecuación (8.51) se obtiene el conjunto de ecuaciones (8.52)

$$\begin{array}{ll} \text{para } n_0 = 0, k_0 = 0 & x_1(0,0) = x_0(0,0) + x_0(1,0) W^0 \\ \text{para } n_0 = 0, k_0 = 1 & x_1(0,1) = x_0(0,1) + x_0(1,1) W^0 \\ \text{para } n_0 = 1, k_0 = 0 & x_1(1,0) = x_0(0,0) + x_0(1,0) W^2 \\ \text{para } n_0 = 1, k_0 = 1 & x_1(1,1) = x_0(0,1) + x_0(1,1) W^2 \end{array} \quad (8.52)$$

que se puede expresar en forma matricial

$$\begin{bmatrix} x_1(0,0) \\ x_1(0,1) \\ x_1(1,0) \\ x_1(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{bmatrix} \begin{bmatrix} x_0(0,0) \\ x_0(0,1) \\ x_0(1,0) \\ x_0(1,1) \end{bmatrix} \quad (8.53)$$

de manera similar se puede escribir la sumatoria externa de la ecuación (8.50)

$$X_2(k_0, k_1) = \sum_{n_0=0}^1 x_1(k_0, n_0) W^{(2k_1 + k_0) n_0} \quad (8.54)$$

que al desarrollar (8.54) obtenemos

$$\begin{array}{ll} \text{para } n_0 = 0, n_1 = 0 & x_2(0,0) = x_1(0,0) + x_1(1,0) W^0 \\ \text{para } n_0 = 0, n_1 = 1 & x_2(0,1) = x_1(0,0) + x_1(0,1) W^2 \\ \text{para } n_0 = 1, n_1 = 0 & x_2(1,0) = x_1(1,0) + x_1(1,1) W^1 \\ \text{para } n_0 = 1, n_1 = 1 & x_2(1,1) = x_1(1,0) + x_1(1,1) W^3 \end{array} \quad (8.55)$$

Arreglando (8.55) en forma matricial

$$\begin{bmatrix} x_2(0,0) \\ x_2(0,1) \\ x_2(1,0) \\ x_2(1,1) \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \begin{bmatrix} x_1(0,0) \\ x_1(0,1) \\ x_1(1,0) \\ x_1(1,1) \end{bmatrix} \quad (8.56)$$

De la ecuación (8.50) y (8.52) se llega a la transformación

$$X(k_1, k_0) = X(k_0, k_1) \quad (8.57)$$

con la ecuación (8.57) se obtiene el conjunto de ecuaciones que representan al algoritmo original de Cooley-Turkey de la transformada rápida de Fourier para $N = 4$

$$X_1(k_0, n_0) = \sum_{n_1=0}^1 x_0(n_1, n_0) W^{2n_0 k_1} \quad (8.58)$$

$$X_2(k_0, k_1) = \sum_{n_0=0}^1 x_1(k_0, n_0) W^{(2n_1+n_0)k_0} \quad (8.59)$$

$$X(k_1, k_0) = X(k_0, k_1) \quad (8.60)$$

Las ecuaciones (8.58) a (8.60) son *recursivas*, ya que la segunda se genera en términos de la anterior. Si se hace extensivo este método, el algoritmo de Cooley-Turkey para $N = 2L$ se representará como [BRI88]

$$\begin{aligned} X(k_{L-1}, k_{L-2}, \dots, k_0) &= \\ &= \sum_{n_0=0}^1 \sum_{n_1=0}^1 \dots \sum_{n_{\gamma-1}=0}^1 x_0(n_{L-1}, n_{L-2}, \dots, n_0) \cdot \left(W^{2L-1(k_0 n_{L-1})} W^{(2k_1+k_0)2^{\gamma-1}n_{L-1}} \right) \dots \left(W^{(2^{L-1}k_L+2^{\gamma-1}k_{L-1}+\dots+n_0)n_0} \right) \end{aligned} \quad (8.61)$$

donde las ecuaciones recursivas están definidas por

$$x_1(k_0, n_{L-2}, \dots, n_0) = \sum_{n_{\gamma-1}=0}^1 x_0(n_{\gamma-1}, n_{\gamma-2}, \dots, n_0) W^{2^{\gamma-1}(k_0 n_{\gamma-1})} \quad (8.62)$$

$$x_2(k_0, k_1, n_{L-3}, \dots, n_0) = \sum_{n_{\gamma-2}=0}^1 x_1(k_0, n_{\gamma-2}, \dots, n_0) W^{(2k_1+k_0)2^{\gamma-2}n_{\gamma-2}} \quad (8.63)$$

$$x_{\gamma}(k_0, k_1, \dots, k_{L-1}) = \sum_{n_0=0}^1 x_{L-1}(k_0, k_1, \dots, n_0) W^{(2^{L-1}k_{L-1}+2^{L-2}k_{L-2}+\dots+k_0)n_0} \quad (8.64)$$

$$X(k_{L-1}, k_{L-2}, \dots, k_0) = x_L(k_0, k_1, \dots, k_{L-1}) \quad (8.65)$$

Este conjunto de ecuaciones recursivas representa la forma en que Cooley-Turkey definieron la FFT para $N = 2^L$.

En la FFT existen L ecuaciones de sumatoria, cada una de las cuales representa N ecuaciones y contiene dos multiplicaciones complejas, pero una de ellas es multiplicada por 1.

Esto se debe a que la primera multiplicación es de la forma $W^{n_{l-1}}$, donde $n_{l-1} = 0$, por tanto sólo se requieren $N \times L$ multiplicaciones. Por otra parte, se puede probar que si $n = p + \frac{N}{2}$

donde p varía entre cero y $\frac{N}{2-1}$, entonces $W^p = -W^{\frac{p+N}{2}}$, por lo que estos datos se pueden manejar en un mismo lugar de memoria y de manera simultánea, el número de multiplicaciones complejas se puede reducir en otro factor de 2, y por lo tanto se realizan $\frac{N \times L}{2}$ multiplicaciones complejas. De manera similar, se puede razonar que se realizan $N \times L$ sumas complejas.

En el caso de que el número de puntos no sea una potencia de 2, se puede llenar con ceros hasta llegar a la siguiente potencia o formular el algoritmo de la FFT para otra base, siguiendo un proceso similar [BRI88].

8.5 ALGORITMO FFT DE DECIMACIÓN EN FRECUENCIA RADIX 2

En este caso se selecciona $M=2$ y $L=N/2$, es decir, que los datos de entrada $x(n)$ se almacena en columnas.

X_0	X_L
X_1	X_{L+1}
X_2	X_{L+2}
X_3	X_{L+3}
.	.
.	.
.	.

Figura 8.11 Arreglo de datos de $x(n)$ como columna

La fórmula de la DFT se divide en dos sumatorias, una sobre los primeros $N/2$ puntos y la otra sobre los últimos $N/2$ puntos.

$$X_1(k) = \sum_{n=0}^{N/2-1} x(n)W_N^{nk} + \sum_{n=N/2}^{N-1} x(n)W_N^{kn} \quad (8.66)$$

donde

$$W_N^{k\left(n+\frac{N}{2}\right)} = W_N^{kn} W_N^{kN/2} = (-1)^k W_N^{kn} \quad (8.67)$$

entonces

$$X_1(k) = \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right] W_N^{kn} \quad (8.68)$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} \left\{ \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^{kn} \right\} W_N^{kn}; k = 0, 1, 2, \dots, \frac{N}{2}-1 \quad (8.69)$$

dividiendo $X(k)$ muestras par e impar

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} g_1(n) W_N^{kn} \tag{8.70}$$

$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} g_2(n) W_N^{kn} \tag{8.71}$$

definiendo las secuencias $g_1(n)$ y $g_2(n)$ de longitud $N/2$

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_N^{kn}; k = 0, 1, 2, \dots, \frac{N}{2} - 1 \tag{8.72}$$

$$g_1(n) = x(n) + x\left(n + \frac{N}{2}\right) \tag{8.73}$$

entonces

$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} \left\{ \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \right\} W_N^{kn}; k = 0, 1, 2, \dots, \frac{N}{2} - 1 \tag{8.74}$$

$$g_2(n) = \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n; n = 0, 1, \dots, \frac{N}{2} - 1 \tag{8.75}$$

El cálculo básico involucra una operación mariposa por decimación en frecuencia, dada en figura 8.12.

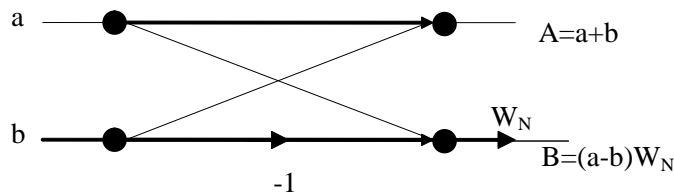
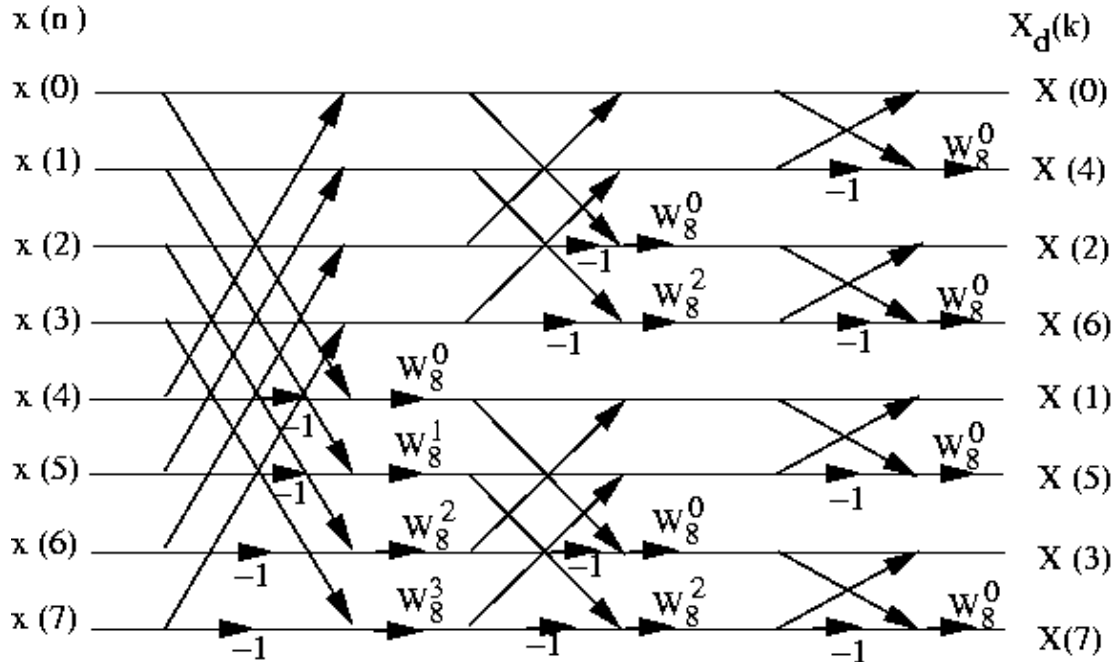


Figura 8.12 Mariposa para decimación en frecuencia

En la figura (8.13) se observa el diagrama del algoritmo, la entrada $x(n)$ está en orden natural y la salida $X(k)$ está ordenada en forma de acarreo inverso. El algoritmo involucra la decimación de la salida $X(k)$ para reordenarla, a este proceso se le llama decimación en frecuencia, el diagrama completo para $N=8$ se ilustra en la figura 8.13.

Figura 8.13 FFT con decimación en frecuencia para $N=8$

8.6 LA FFT INVERSA

Utilizando la misma red de FFT Radix 2 decimada en tiempo se puede calcular la FFT Inversa siguiendo el procedimiento:

1. Introducir $X(k)$ decimada
2. En la red cambiar coeficientes W^n en el orden: $W_N^0 = 1$
Y el resto de coeficientes W cambiar el signo: $-W_N^n$
3. Dividir la secuencia de salida entre N .

Ejemplo 8.3

Dada la secuencia en espectro discreto:

$$X(k) = \{4, 1 - 2.4142j, 0, 1 - 0.4142j, 0, 1 + 0.4142j, 0, 1 + 2.4142j\}$$

aplicando la FFT inversa calcular $x(n)$.

Decimado en frecuencia:

$$\Rightarrow X_d(k) = \{4, 0, 0, 0, 1 - 2.4142j, 1 + 0.4142j, 1 - 0.4142j, 1 + 2.4142j\}$$

se obtiene:

$$x(n) = \{1, 1, 1, 1, 0, 0, 0, 0\}$$

Para efectos de facilitar la forma de realizar la FFT “radix” 2, en la figura 8.14 se muestra un diagrama de la FFT decimada en tiempo para $N=16$.

Como parte final de este tema, en la figura 8.14 se muestra un diagrama de la FFT para $N=16$, éste facilita la comprensión de la programación de la FFT “radix” 2 decimada en tiempo. En la figura 8.15 se presenta un diagrama de flujo para programar la FFT y posteriormente el programa de la FFT realizado en Matlab y en lenguaje ensamblador en el DSP TMS320C50.

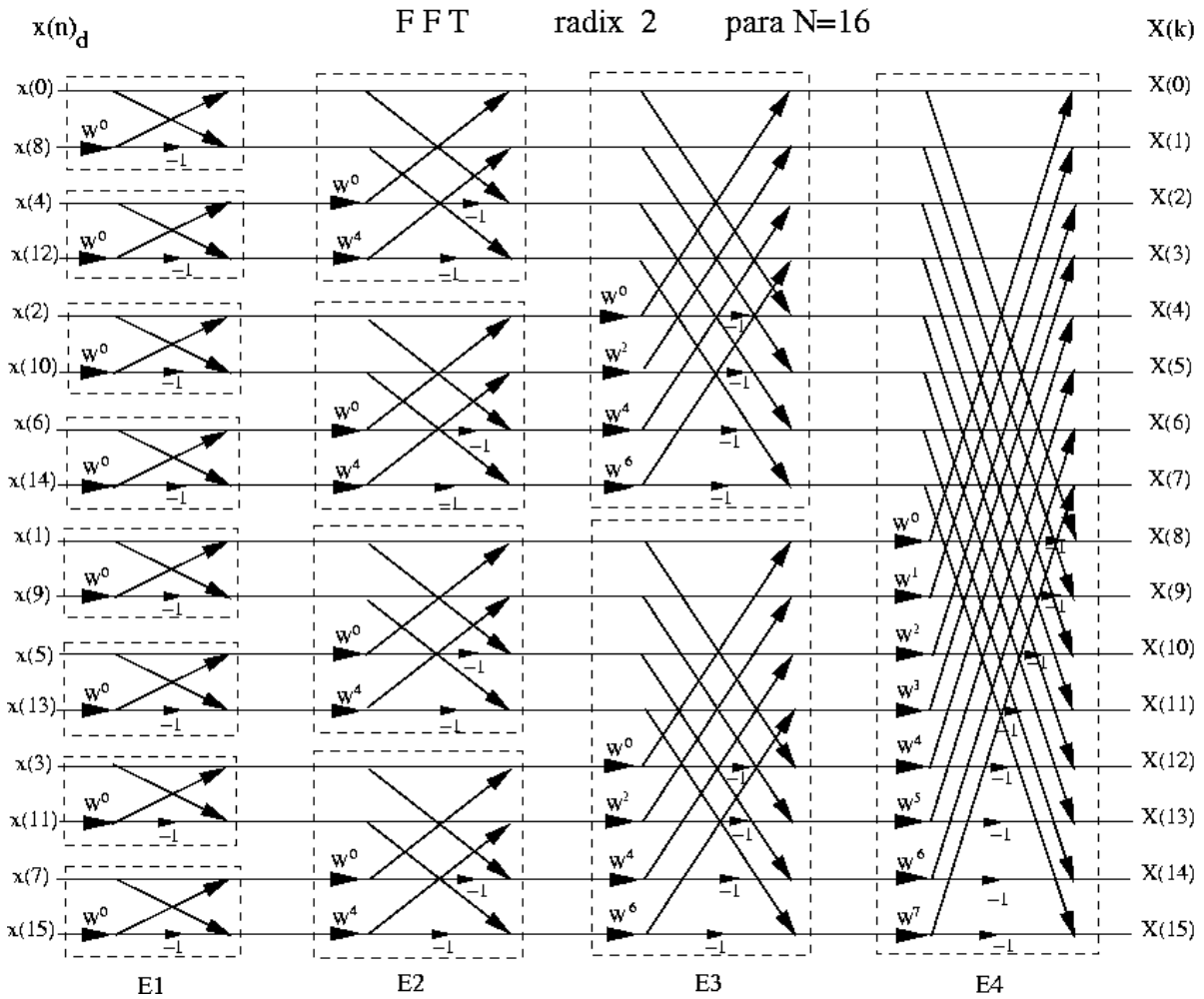


Figura 8.14 Diagrama de FFT para $N=16$

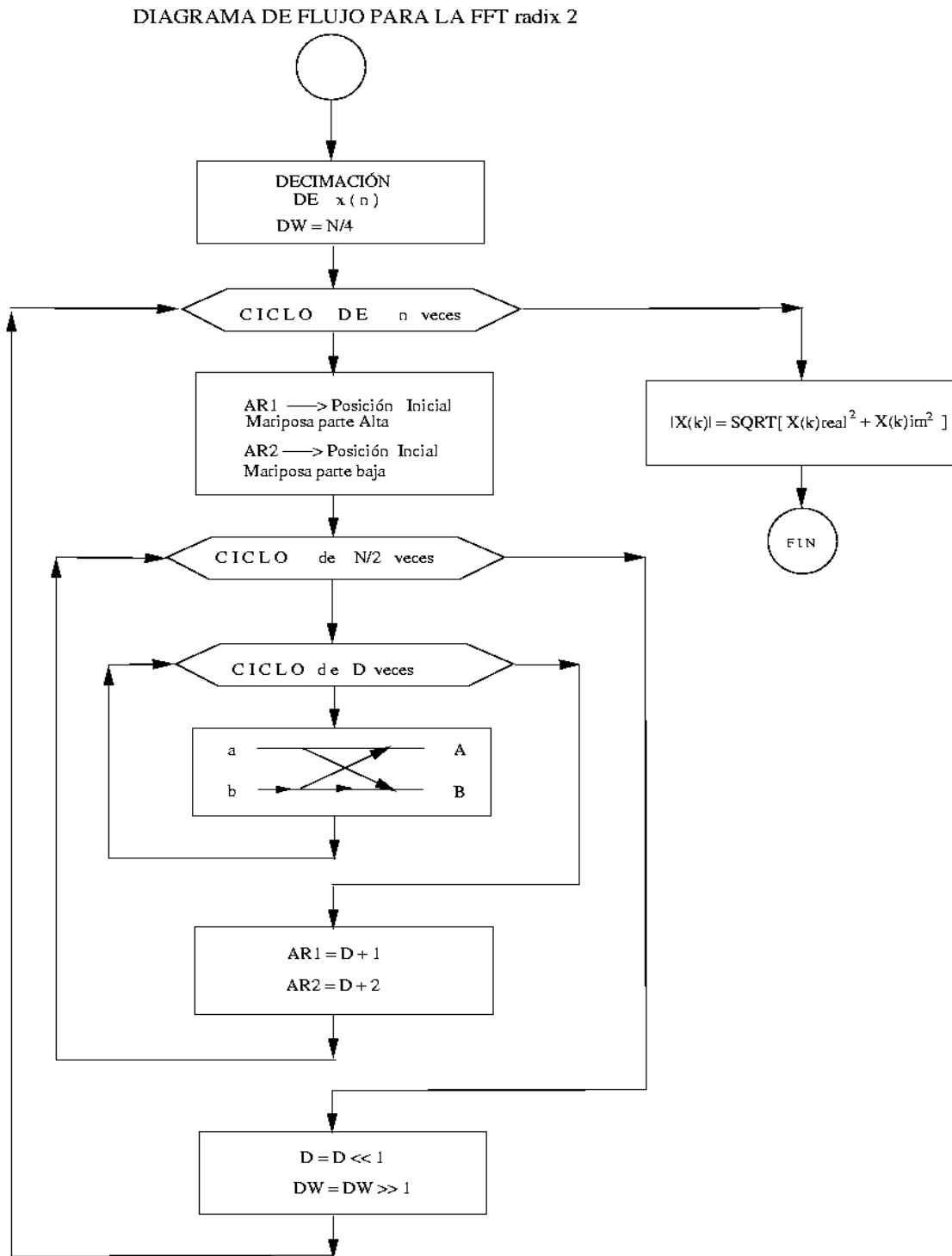


Figura 8.15 Diagrama de flujo de FFT

LA FFT PROGRAMADA EN MATLAB

En este programa se ha desglosado al máximo el algoritmo y se utilizan operaciones elementales para que sea más ilustrativa la comprensión de implementación de la FFT y fácil implementación en otro lenguaje.

```

%   ALGORITMO PARA LA IMPLANTACION DE LA TRANSFORMADA
%   RAPIDA DE FOURIER (FF)
%   Versión para MATLAB
%   @larry
%   INICIO DE LA DECIMACION
%
load señal.dat;           % carga la señal de datos x(n)
x = señal;
z = zeros(1,length(x));
xfft = x;
x = [ xfft; z];
N = length(x);
% tiempo = clock;
n = log(N)/log(2);
ne = n;

%
%   CONVERSION DE UN NUMERO DECIMAL DE ENTRADA ORDENADO
%   A BINARIO INVERTIDO
%
entra = x;
for k=1:N-1
    dec=k;
    for i=1:n+1
        z = dec/2;
        yb(k,i) = rem(dec,2);
        dec = fix(z);
    end
end
for i=1,n
    yb(N,i)=0;
end
    yb(N,n+1)=1;

%
%   CONVERSION DE BINARIO INVERTIDO A DECIMAL DECIMADO
%
for j=1:N
    m = ne;
    for i=1:n
        m = m-1;

```

```

        z(i) = yb(j,i)*2^m;
        end
        yd(j) = sum(z);
    end
%
% PASA VALORES DE X[K] ORDENADO A X[K] DECIMADO
%
for k=1:N-1
    valor = yd(k)+1;
    y2(k+1) = x(valor);
end
x1 = y2;
x1(1) = x(1);
x2 = zeros(1,N);
x = [x1;x2];
%
% CALCULO DE LA TRANSFORMADA RAPIDA DE FOURIER
% PARA N PUNTOS
%
clf;
for d=1:N/2;
    z1(d)=x(2,d);
    entra1(d)=entra(1,d);
end
% variable net, para calcular el número de etapas
net = log(N)/log(2);
% Generacion de los valores de Wn()
ang = 2*pi/N;
for i=1:N
    i1 = i-1;
    w(1,i) = cos(ang*i1);
    w(2,i) = sin(ang*i1);
end
%
% Ejecución de la FFT para n=N/4 etapas de Mariposas
% n inicial: número de etapas
% nm: control de etapa en proceso
% p : número de la etapa en proceso
% k : índice de la del arreglo de valores x() de entrada
% nn: índice que lleva el control de Wn()
% l : control del salto de posición
% de un conjunto de mariposas
% a otro conjunto dentro de la misma etapa
% i: índice que lleva el control del número de mariposas
% a operar en un conjunto dentro de una misma etapa
%

```

```

% PROGRAMA PRINCIPAL DE LA EJECUCION DE LA FFT
%
nm = N/2; n = N/2; p=1;
%
% Entra a una etapa de mariposas
%
for e=1:net
    k=0;
%
% Cambia de posición de un conjunto de mariposas a otro
%
for l=1:nm
    nn=1;
%
% calculo de i mariposas en un conjunto
%
    for i=1:(2^(p-1))
        if l==1
            if i==1
                k=1; nn=1;
            else
                k = k+1;
            end
            else
                k=k+1;
            end
            if i==1
                nn = 1;
            else
                nn = nn+n;
            end
            k1 = k+2^(p-1);
%
% EFECTUA LA OPERACION DE UNA MARIPOSA
%
xr(1,k) = x(1,k) + x(1,k1).*w(1,nn) - x(2,k1).*w(2,nn);
xi(2,k) = x(2,k) + x(1,k1).*w(2,nn) + x(2,k1).*w(1,nn);
xr(1,k1) = x(1,k) - x(1,k1).*w(1,nn) + x(2,k1).*w(2,nn);
xi(2,k1) = x(2,k) - x(2,k1).*w(1,nn) - x(1,k1).*w(2,nn);
x(1,k) = xr(1,k);
x(2,k) = xi(2,k);
x(1,k1) = xr(1,k1);
x(2,k1) = xi(2,k1);
end
    k=2^(p-1)+k;
    end
    p=p+1; n=n/2;

```

```

        nm = nm/2;
%     x=y;
end
%
%     Cálculo del |H(w)|
%
for j=1:N
        c1(j) = sqrt((x(1,j))^2+(x(2,j))^2);
        fase1(j) = atan(x(2,j)/(x(1,j)+0.0001));
end
% tfinal = etime(clock,tiempo);
% disp('TIEMPO DE PROCESO OCUPADO POR LA FFT = '); tfinal
%
%     GRAFICACION DE x[k] de entrada y espectro |H(w)|
%
for j=1:N/2
        co(j) = c1(j);
        fase(j) = fase1(j);
end
save fftr co;
subplot(211);
title('Señal Temporal discreta');
xlabel('Muestreos de k');
ylabel('Magnitud');
grid;
plot(entral);
subplot(212);
title('F      F      T');
xlabel('Frecuencia');
ylabel('Magnitud');
grid;
plot(co,'g')
pause;
x=xfft;

```


LA FFT PROGRAMADA EN UN DSP

```

*      Programa en lenguaje ensamblador para el DSP TMS320C50
*      FFT para N=32 , radix 2
*      @larry
      .mmregs
      .ds 0F00h
*** Variables para operación Mariposa
A      .word 0
AAR    .word 0
AAI    .word 0
BR     .word 0
BI     .word 0
ar     .word 0
ai     .word 0
br     .word 0
bi     .word 0
WR     .word 0
WI     .word 0
Wbr    .word 0
Wbi    .word 0
D0     .word 0
DW     .word 4
      .ds 01000h      ; sección de datos
;                          de señal de entrada
xr     .Q10 1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0
xr1    .Q10 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
xi     .Q10 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
xi1    .Q10 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
xk2    .Q10 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
xk21   .Q10 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATOS  .word 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DAT    .word 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

***** Factores W32/2
SEN    .Q10 0.00000, 0.19509, 0.38268, 0.55557, 0.70711,
SEN1   .Q10 0.83147, 0.92388, 0.98078, 1.00000, 0.98078
SEN2   .Q10 0.92388, 0.83147, 0.70711, 0.55557, 0.38268,
SEN3   .Q10 0.19509, 0.00000
*
COS    .Q10 1.00000, 0.98078, 0.92388, 0.83147, 0.70711
COS1   .Q10 0.55557, 0.38268, 0.19509, 0.00000, -0.19509
COS2   .Q10 -0.38268, -0.55557, -0.70711, -0.83147, -0.92388
COS3   .Q10 -0.98078, -1.00000
*
N      .set 32

```

```

N1      .set  31      ; N-1 dato
N2      .set  16
n1      .set  3      ; n=log2(N)
ONE     .set  1
*
      .ps  0A00h      ; sección de código de programa
      .entry
      SETC SXM
      SPM  #0
      LAR  AR1,#xr
      MAR  *,AR1
      CALL DECIMA      ;----->
      LDP  #D0          ; D0
      LACL #ONE
      SACL D0          ; paso inicial
      LACL #N          ; Salto a posición imaginaria de x(n)
      SAMM INDX        ;
***** Ciclo principal
      LACL #n1
      SAMM BRCCR
      RPTB FFT
      LAR  AR1,#xr ; Ubica apuntadores a mary Inicio
      LAMM AR1      ; AR1 para Posición alta de Mary
      ADD  D0
      SAMM AR2      ; AR2 para Posición Baja de Mary

*****
      LACL #N2
      SUB  #ONE      ; AR7 = 15, => 16 MARYS POR ETAPA
      SAMM AR7
***** loop de N/2 marys *****
LOOP_N2 LACL D0
      SUB  #ONE
      SAMM AR6      ; para loop D0

      LAR  AR3,#COS ; Wreal_0
      LAR  AR4,#SEN ; Wimg_0
** Loop de D0 marys, D0=2^#Etapa
*LOOP_D  LDP  #A
LOOP_D  MAR  *,AR1
      LACL *0+      ; trae valores de x(n) real para mary
      SACL ar
      LACL *0-,AR2
      SACL ai      ; trae valores de x(n) img.

      LACL *0+      ; trae valores de x(n) real
      SACL br

```

```

LACL *0-,AR3
SACL bi      ; trae valores de x(n) img.
****
LACL *,AR4 ; trae valores de COSENOS para WReal.
SACL WR
LACL *,AR1
SACL WI      ; trae valores de SENOS para Wimg.

      CALL MARY ; ----->>
****
MAR  *,AR1
LACL AAR    ; Salva valores Altos de mary en x(n) real
SACL *0+
LACL AAI
SACL *0-,0,AR2 ; Salva valores Altos mary en x(n) img
*de mary.
LACL BR    ; trae valores de x(n) real
SACL *0+
LACL BI
SACL *0-,0,AR1 ; trae valores de x(n) img.
*****
***** Salto en uno de AR1 y AR2, en el mismo conjunto de
Marys
MAR  *,AR2 ; AR1=AR1+1
MAR  *,AR6 ; AR2=AR2+1
*** ***** MODIFICAR LOS WR (AR3) y WR (AR4)
LAMM AR6
BCND  NOC_DW,EQ ; en ETAPA 1 No modifica DW a usar
*(W0)

LAMM AR3 ; AR3=AR3+DW
ADD DW
SAMM AR3
LAMM AR4 ; AR3=AR3+DW
ADD DW
SAMM AR4
NOC_DW NOP
BANZ  LOOP_D,*-,AR1
*****
LAMM AR1 ; Salto a otro conjunto de Marys
ADD D0 ;
***
ADD #ONE
SAMM AR1 ; Posición Alta
LAMM AR2 ; Salto a otro conjunto de Marys
ADD D0 ;
****
ADD #ONE
SAMM AR2 ; Posición baja

```

```

        MAR    *,AR7
        BANZ   LOOP_N2,*-
* *****
        LACC D0
        SFL           ; D0 << 1, para la siguiente ETAPA de Marys
        SACL D0
***** Modifica DW, para siguiente ETAPA
        LACL DW
        SFR           ; DW >> 1
        SACL DW

FFT     NOP
        CALL  MODULO ; Cálculo de X(k)
FIN     NOP
        B    FIN

*****+*****
***** Subrutina de la Decimación Radix 2
DECIMA  LACL #N2
        SAMM INDX           ; INDX = N/2
        RPT  #N1
        BLDD *BR0+,#DATOS ; copia datos y decima
        LAR  AR1,#xr
        RPT  #N1
        BLDD #DATOS,+      ; regresa los datos decimados
        RET

***** Subrutina de la Mariposa *****+****
*MARY   LDP  #A
MARY    LT   WR           ; ver el Qi para operar y salvar
        MPY  br           ; WR*br
        PAC
        LT   WI
        MPY  bi           ; WI*bi
        SPAC
        SACH Wbr,6       ; Wbr=WR*br-WI*bi
        MPY  br           ; WI*br
        PAC
        LT   WR
        MPY  bi           ; WR*bi
        APAC           ; Wbi= WI*br+WR*bi
        SACH Wbi,6      ; Q10
        LACC Wbi
        ADD  ai
        SACL AAI        ; AAI=ai+Wbi
        LACC ar
        ADD  Wbr

```

```

        SACL AAR      ; AAR=ar+Wbr
        LACC ar
        SUB  Wbr
        SACL BR      ; BR=ar-Wbr
        LACC ai
        SUB  Wbi     ; BI=ai-Wbi
        SACL BI
        RET
*****
* SUBROUTINA PARA EL CALCULO DEL MODULO DE X(K)^2
*****
MODULO  MAR *,AR1
        LAR AR1,#xr
        LAR AR2,#xi
        LAR AR3,#xk2
        LACL #N1
        SAMM BRCR
        ZAP
        RPTB FIN_MOD
          LT *
          MPY *+,AR2 ; xr^2
          LTA *
          MPY *+,AR3 ; xi^2
          APAC ; X(k)^2 = xr^2+ xi^2
FIN_MOD SACL *+,0,AR1
        RET
*****
        .end

```

8.7 RESUMEN

En este capítulo se ha analizado la transformada rápida de Fourier que es una transformada muy utilizada en procesamiento digital de señales. Cuando se desarrolló la FFT el PDS logró grandes avances, ya que permitió la implementación de aplicaciones en tiempo real. La FFT en la actualidad se encuentra en un gran número de aplicaciones y es de gran importancia conocerla e implementarla. En este trabajo se han analizado sus características y su forma de implementarla, además se han dado ejemplos de programas que faciliten al lector programar en otro lenguaje de su preferencia.

8.8 PROBLEMAS PROPUESTOS

1. Para $N=8$ escribir la matriz de Fourier y su factorización por procedimientos de la FFT “radix” 2.
2. Para una secuencia de longitud 32, realizar la decimación en el tiempo utilizando los tres primeros métodos expuestos en este capítulo.
3. Elaborar un diagrama de la FFT “radix” 2 para $N=32$.
4. Para una secuencia senoidal con tres ciclos en $N=16$, calcular la FFT efectuando todas las operaciones necesarias sobre un diagrama de FFT para $N=16$.
5. Con base en la reducción sucesiva de operaciones al ir desarrollando la FFT, proponer una ecuación en diferencias para el número de sumas y otra para los productos, y luego obtener las fórmulas que determinan el número de sumas y productos que realiza la FFT “radix” 2.

CAPITULO 9

LA TRANSFORMADA COSENO

Las transformadas matemáticas han sido utilizadas para facilitar la solución de problemas matemáticos. El objetivo de estas transformadas es trasladar una señal de su espacio original a otro espacio matemático, donde el manejo e interpretación de la información se torne más fácil [RAO90].

En años recientes ha habido un crecimiento respecto a las aplicaciones de las transformadas matemáticas en el *Procesamiento Digital de Señales e imágenes*, y se han desarrollado algoritmos rápidos que tienen como consecuencia, una reducción en los costos computacionales y de memoria en la aplicación de las transformadas. Algunas de las aplicaciones de estas transformadas son: procesamiento de imágenes, procesamiento de voz, reconocimiento de patrones, reconocimiento de caracteres, análisis y diseño de sistemas de comunicación, filtrado digital, análisis espectral, compresión de datos, procesos de convolución y correlación, análisis estadístico; además de aplicarse a señales tan variadas como: señales sísmicas, sonar, radar, biológicas y biomédicas, astronomía, ondas oceanográficas, imágenes satelitales, vibraciones estructurales, rayos x, entre otras [ELL82].

En este capítulo se analiza la transformada discreta coseno (DCT) que es la base de muchas aplicaciones como voz, audio, imágenes, video, etc. Al final se presenta como ejemplo los efectos de la DCT en una imagen al reducir a cero algunos coeficientes debajo de un umbral y aplicar la DCT inversa (DCTI).

9.1 TRANSFORMADAS

Para la gran cantidad de aplicaciones actuales existe una diversidad de transformadas, las cuales proporcionan resultados diferentes al momento de su aplicación, los criterios principales para utilizar una transformación son el tipo de señal que se tiene y lo que se desea obtener de ella, por ejemplo la transformada de Fourier (*TF*) permite encontrar el espectro (las componentes frecuenciales que contribuyen a la formación de esa señal en particular) de una señal dada.

La transformada óptima en el sentido estadístico para el desarrollo de la eliminación de la redundancia espacial, es la transformada Karhunen-Loeve (*KLT*), ésta decorrelaciona los coeficientes transformados concentrando la mayor parte de la energía en unos pocos coeficientes, además de minimizar el error cuadrático medio entre las señales original y reconstruida.

Sin embargo, la implementación de la *KLT* involucra la determinación de eigenvalores y sus correspondientes eigenvectores y no existe un algoritmo general para su computación rápida, la *KLT* tiene una propiedad teórica óptima de decorrelación, pero pocas veces es usada en la práctica.

En vista de la complejidad asociada con la implementación de la *KLT*, han sido utilizadas otras transformadas discretas en el procesamiento de imágenes y señales, de hecho algunas transformadas tales como la *DCT* y la transformada discreta de Fourier (*DFT*) son asintóticamente equivalentes a la *KLT* [ELL82]. La *DCT* fue aplicada en el trabajo pionero de Ahmed, Natarajan y Rao, en la cual mostraron que esta transformada en particular es muy cercana a la *KLT*, la cual produce coeficientes decorrelacionados [PEN93].

La *DCT* es una transformación eficiente que descompone una señal en un grupo de señales cosenoidales ortogonales llamadas funciones bases, la transformada toma un conjunto de muestras y nos entrega un conjunto de coeficientes, el número de elementos en ambos conjuntos es el mismo [RAO90].

9.1.1 VENTAJAS DE LA DCT SOBRE LA KLT

Existe una equivalencia asintótica entre la *DCT* y la *KLT*, para longitudes de datos finitos se pueden generar diferentes aproximaciones para la *KLT*. En general existen varias características que son deseables en la *DCT* cuando es usada con el propósito de comprimir datos en sustitución de la *KLT* :

- *Decorrelación de datos:* La transformada ideal debe decorrelacionar completamente los datos en una secuencia, concentra la mayor parte de la energía en pocos coeficientes, de esta manera muchos coeficientes tendrán valores pequeños.
- *Funciones base independientes:* Debido a las grandes variaciones estadísticas en los datos, la transformada óptima depende de los datos, esto es particularmente un problema si los bloques de datos no están muy relacionados, es decir, necesita el uso de más de un conjunto de funciones base para alcanzar una alta decorrelación, por lo tanto es deseable un desarrollo óptimo para las transformadas cuyas funciones base son independientes.
- *Implementación rápida:* Existen algoritmos matemáticos que permiten la implementación de la *DCT* y *DCTI* con una cantidad menor de operaciones, con lo cual se logra optimizar su desarrollo.

9.2 CLASIFICACIÓN DE LAS TRANSFORMADAS

La transformada Coseno y la transformada Seno discreta (*DST*) son miembros de una familia de transformadas unitarias senoidales, que encuentran aplicaciones en procesamiento digital de señales e imágenes y particularmente en sistemas de codificación por transformada para compresión y descompresión de datos.

A pesar de ser real, ortogonal y separable sus propiedades se han utilizado ampliamente en compresión de datos y sus algoritmos rápidos han hecho un gran aporte a aplicaciones en tiempo real. Recientemente la *DCT* ha sido empleada como la herramienta principal de procesamiento para compresión y descompresión de datos en los estándares internacionales de codificación de, voz, audio, imágenes y video. Una transformada alternativa usada en sistemas de codificación por transformada es la *DST*, de hecho el uso alternado de las formas modificadas de la *DCT* y la *DST* han sido adoptado en el estándar internacional de codificación de audio MP3, de imágenes JPEG y de video en MPEG-1, MPEG-2 y MJPEG, etc.

La transformadas *DCT* y *DST* son miembros de una clase de transformaciones unitarias senosoidales desarrolladas por Jain, una transformada unitaria sinusoidal es una transformada lineal invertible cuyo Kernel describe un conjunto completo de funciones coseno y seno ortogonales discretos (funciones base) y son miembros de esta clase de transformadas unitarias.

9.2.1 TRANSFORMADA INTEGRAL COSENO

La transformada Coseno al igual que muchas otras transformadas son definidas de acuerdo a una operación de integración la cual muchas veces se denomina *Kernel* de la transformación,

gracias a esta operación se logra trasladar una señal en su espacio original al espacio de transformación definido por la ecuación (9.1).

Considerando la parte real del núcleo de la transformada de Fourier en el tiempo continuo para una función $x(t)$ que está definida para $0 \leq t < \infty$, entonces la transformada integral Coseno o simplemente transformada Coseno (TC), $X_c(\omega)$ está definida por la ecuación (9.1)

$$X_c(\omega) = F_c\{x(t)\} = \sqrt{\frac{2}{\pi}} \int_0^{\infty} x(t) \cos(\omega t) dt \quad (9.1)$$

Asímismo la transformada inversa Coseno en el tiempo continuo (TCI), se define por la ecuación (9.2)

$$x(t) = F_c^{-1}\{X(\omega)\} = \sqrt{\frac{2}{\pi}} \int_0^{\infty} X(\omega) \cos(\omega t) d\omega \quad (9.2)$$

9.2.2 TRANSFORMADA COSENO DISCRETA UNIDIMENSIONAL

Debido a las definiciones de TC y TCI el kernel de la transformada Coseno está dado por la ecuación (9.3)

$$K_c(\omega, t) = \cos \omega t \quad (9.3)$$

Si se considera $\omega_n = 2\pi m \delta f$ y $t_n = n \delta t$, es decir, muestrear la frecuencia angular y el tiempo respectivamente, donde m y n son enteros, la ecuación (9.3) se puede reescribir como

$$K_c(\omega_n, t_n) = K_c(2\pi m \delta f, n \delta t) = \cos(2\pi mn \delta f \delta t) \quad (9.4)$$

por definición se considera que $\delta f \delta t = 1/2N$, donde N es un número entero, por lo tanto se tiene que la ecuación (9.4) se transforma ahora en la ecuación (9.5)

$$K_c(m, n) = \cos((mn\pi)/N) \quad (9.5)$$

La ecuación (9.5) representa el *kernel* discretizado de la transformada Coseno. De acuerdo a este *kernel* se definen cuatro formas para la transformada Coseno discreta (DCT), las cuales se definen a partir de las ecuaciones (9.6), (9.7), (9.8) y (9.9)

DCT-I

$$[Cn]_{mn} = (2/N)^{1/2} (K_m K_n \cos((mn\pi)/N)) \quad m, n = 0, 1, 2, \dots, N \quad (9.6)$$

DCT-II

$$[Cn]_{mn} = (2/N)^{1/2} (K_m \cos((m(n+0.5)\pi)/N)) \quad m, n = 0, 1, 2, \dots, N \quad (9.7)$$

DCT-III

$$[Cn]_{mn} = (2/N)^{1/2} (K_n \cos((n(m+0.5)\pi)/N)) \quad m, n = 0, 1, 2, \dots, N \quad (9.8)$$

DCT-IV

$$[Cn]_{mn} = (2/N)^{1/2} (\cos(((n+0.5)(m+0.5)\pi)/N)) \quad m, n = 0, 1, 2, \dots, N \quad (9.9)$$

donde

$$K_n = \begin{cases} 1 & \text{Si } n \neq 0 \\ \frac{1}{\sqrt{2}} & \text{Si } n = 0 \end{cases} ; \quad K_m = \begin{cases} 1 & \text{Si } n \neq 0 \\ \frac{1}{\sqrt{2}} & \text{Si } n = 0 \end{cases}$$

y $[C_n]_{mn}$ es el elemento mn-ésimo de la matriz *DCT*.

Las formas *DCT-II* y *DCT-III* proveen una excelente decorrelación estadística y son la mejor aproximación para la optimización de la *KLT* tomando en consideración un número de criterios de desempeño como la eficiencia en el empaquetamiento de energía y razón de distorsión. La importancia de las *DCT-II* y *DCT-III* está acentuada por la existencia de algoritmos “rápidos” para su implementación.

Con base en las definiciones *DCT-II* y *DCT-III* (ecuaciones (9.7) y (9.8)) la transformada discreta Coseno unidimensional para una secuencia discreta $f(x)$ de N muestras, se define por la ecuación (9.10) [PEN93]

$$F(u) = \frac{C(u)}{2} \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \quad (9.10)$$

La definición matemática de la transformada Coseno discreta inversa (*DCTI*), para un conjunto de N muestras de $F(u)$ se define por la ecuación (9.11)

$$f(x) = \sum_{u=0}^{N-1} \frac{C(u)}{2} F(u) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \quad (9.11)$$

donde

$F(u)$ es el valor de un coeficiente DCT unidimensional.

$f(x)$ es el valor de una muestra unidimensional.

$$C(u) = \begin{cases} 1 & \text{Si } u > 0 \\ \frac{1}{\sqrt{2}} & \text{Si } u = 0 \end{cases}$$

Como se ha señalado la *DCT* es una transformación que descompone a una señal en un grupo de señales cosenoidales ortogonales llamadas funciones base, estas funciones base de acuerdo a la definición de la ecuación (9.10) son generadas por el término Coseno y a su vez son

escaladas por el factor $C(u)$, estas funciones junto con los valores originales de las muestras son los encargados de generar los coeficientes *DCT*.

En la figura 9.1 se muestran las funciones base para un conjunto de ocho muestras, el eje de las ordenadas contiene la amplitud de las señales base sin escalar y el eje de las abscisas, al número correspondiente de cada muestra [HER04].

En la Figura 9.1 se pueden ver un conjunto de ocho formas de onda de diferente amplitud, cada una formada por ocho elementos; la primera corresponde al coeficiente *DCT* cero y es simplemente una constante, mientras que las otras siete muestran un comportamiento alternado conforme la frecuencia se incrementa; el coeficiente que corresponde a la función base constante es llamado coeficiente de *DC* (por su similitud a corriente directa al ser constantes), mientras que las otras funciones base corresponden a los denominados coeficientes de *AC* (por su similitud a corriente alterna al ser variables) [PEN93].

Los valores de la primera función base son obtenidos de la siguiente manera: por corresponder al primer coeficiente *DCT* el valor de u es igual cero, debido a esto el término Coseno proporciona un valor de uno, no importando la variación de x , después se debe escalar por el factor $C(u)$ al cual le corresponde el valor de 0.7071 , que debe ser dividido entre 2, lo que da como resultado 0.3536 , para todas las muestras de esta función base. Este procedimiento es el utilizado en la generación de todas las funciones base de la figura 9.1.

Ejemplo 9.1

Considérese el vector $\mathbf{a} = [84 \ 84 \ 90 \ 92 \ 88 \ 66 \ 77 \ 86]$

Que representa el conjunto de ocho muestras que se desea transformar por la aplicación de la *DCT-unidimensional*, la obtención del vector transformado \mathbf{a}_T , se puede lograr por la aplicación de la ecuación (9.10) o bien si se conoce una tabla de los valores de las funciones base escaladas por el factor $C(u)$, en cuyo caso solo es necesario realizar una suma de productos. Para obtener el coeficiente de *DC* se realizan las siguientes operaciones

$$(84 \times 0.3536) + (84 \times 0.3536) + (90 \times 0.3536) + (92 \times 0.3536) + (88 \times 0.3536) + (66 \times 0.3536) \\ + (77 \times 0.3536) + (86 \times 0.3536) = 235.8512$$

Este resultado es el valor del coeficiente *DC*, de igual manera se sigue el mismo procedimiento en la obtención del primer coeficiente de *AC*, solo que ahora se utilizan los valores de la función base uno.

$$(84 \times 0.4904) + (84 \times 0.4157) + (90 \times 0.2778) + (92 \times 0.0975) + (88 \times -0.0975) + (66 \times 0.2778) \\ + (77 \times -0.4157) + (86 \times -0.4904) = 8.9863$$

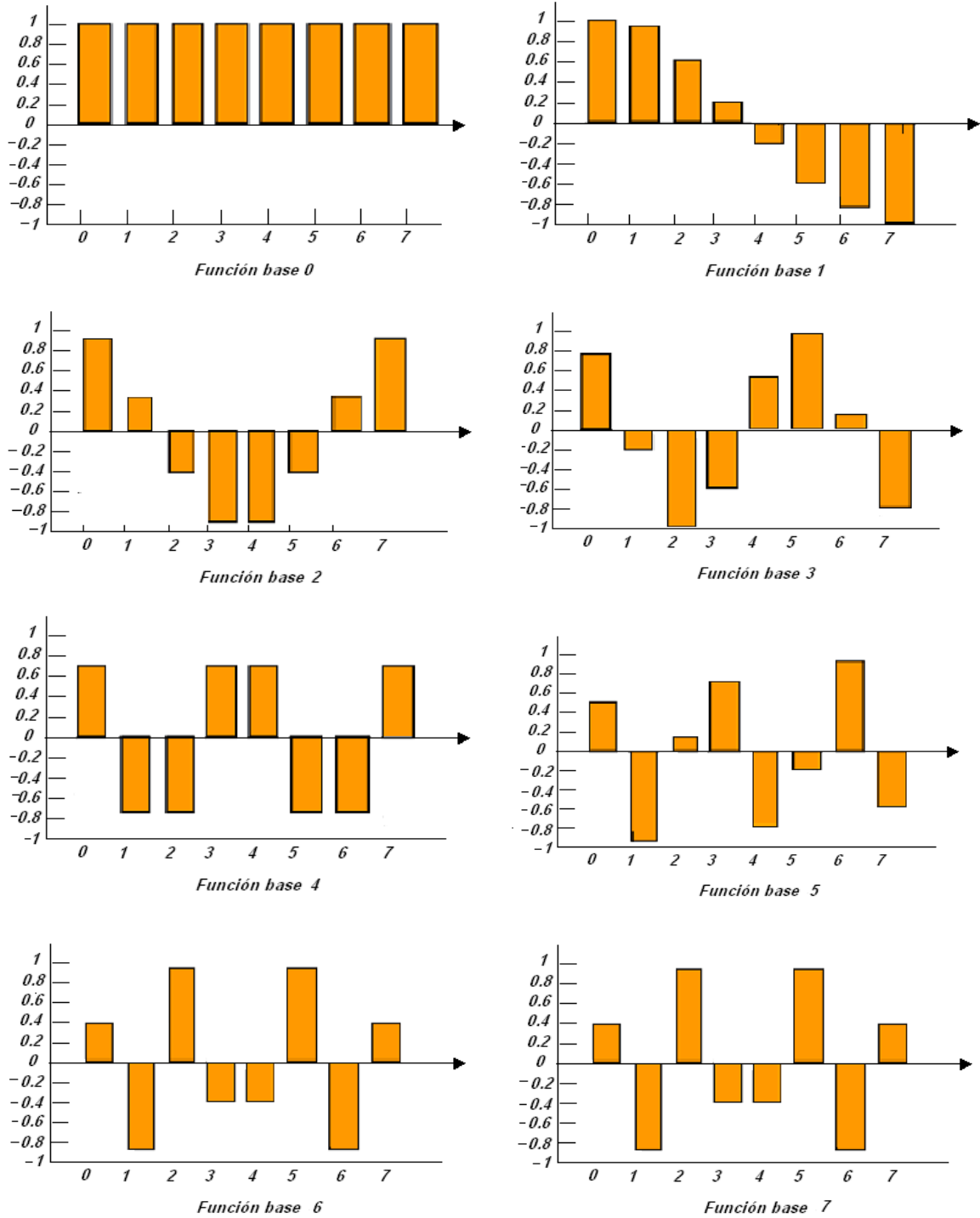


Figura 9.1 Funciones base de la DCT unidimensional

Utilizando el mismo procedimiento, se encuentran todos los componentes del vector de coeficientes transformados

$$\mathbf{a}_T = [235.8512, 8.9863, -3.6627, -14.3948, 11.6673, 0.0157, -4.2231, 5.8765]$$

Al aplicarle la transformada *DCTI* al vector \mathbf{a}_T se obtienen nuevamente los valores originales contenidos en el vector \mathbf{a} . En el vector \mathbf{a}_T es posible apreciar como la mayor parte de la energía del vector \mathbf{a} se concentra en el coeficiente de *DC*, con respecto a sus otros elementos, que presentan valores mucho mas bajos.

El número total de operaciones que realizan la *DCT* y *DCTI unidimensionales* para un conjunto de ocho elementos, siguiendo el criterio de este ejemplo es 120 (56 sumas y 64 multiplicaciones).

9.2.5 TRANSFORMADA COSENO DISCRETA BIDIMENSIONAL

Debido a que en compresión de imágenes y video se utiliza ampliamente la *DCT*, las imágenes se dividen en bloques o matrices de 8x8 pixeles, por lo que en la ecuación (9.12) se define la transformada Coseno discreta bidimensional o simplemente *DCT* en dos dimensiones [RA090]

$$F(u, v) = \frac{C(u)}{2} \frac{C(v)}{2} \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right] \quad (9.12)$$

con $u, v = 0, 1, 2, 3, 4, 5, 6, 7$

A su vez la definición matemática de la *DCTI* en dos dimensiones para un conjunto de 64 coeficientes agrupados en una matriz de 8x8, se muestra en la ecuación (9.13)

$$f(x, y) = \sum_{u=0}^7 \sum_{v=0}^7 \frac{C(u)}{2} \frac{C(v)}{2} F(u, v) \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right] \quad (9.13)$$

donde

$$C(v) = \begin{cases} 1 & \text{Si } v > 0 \\ \frac{1}{\sqrt{2}} & \text{Si } v = 0 \end{cases} \quad C(u) = \begin{cases} 1 & \text{Si } u > 0 \\ \frac{1}{\sqrt{2}} & \text{Si } u = 0 \end{cases}$$

$f(x, y)$ Es el valor de una muestra bidimensional, para los índices x, y de la matriz de elementos.

$F(u, v)$ Es el valor de un coeficiente *DCT bidimensional*, para los índices u, v de la matriz de coeficientes.

Debido a que las imágenes son arreglos en dos dimensiones, la *DCT* en una dimensión se puede extender para transformar arreglos de dos dimensiones que se generan a partir de la multiplicación de dos conjuntos de una dimensión; un conjunto unidimensional representa las frecuencias espaciales horizontales y el otro las frecuencias espaciales verticales. Por convención el coeficiente de *DC* de las frecuencias espaciales horizontales se encuentra en el extremo izquierdo del arreglo y el coeficiente de las frecuencias base verticales en la parte superior, en consecuencia el único término de *DC*, es el que se encuentra en la esquina superior izquierda.

Para la definición de la *DCT* de 64 muestras, se genera un conjunto de 64 funciones base las cuales se muestran en la Figura 9.2, en esa figura se observa la variación de todas las funciones base, así como la función que corresponde al coeficiente de *DC* (esquina superior izquierda posición (0,0)) que presenta un valor constante, los demás cuadros corresponden a los coeficientes de *AC* y en el último en la esquina inferior derecha se puede observar la máxima variación (posición (7,7)). Con estas 64 funciones base se puede generar cualquier arreglo de 8x8 coeficientes de la *DCT*.

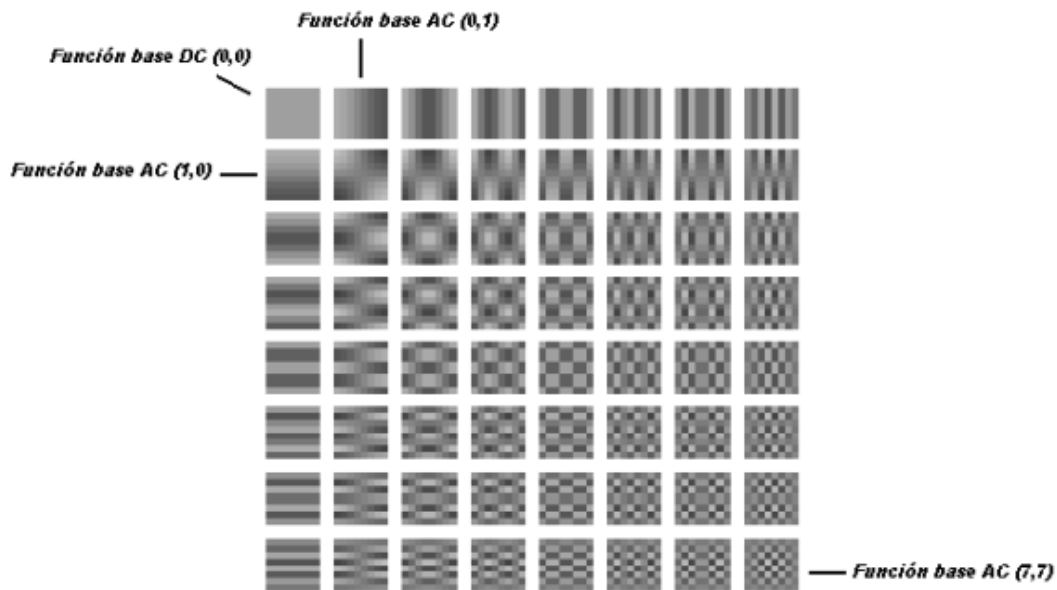


Figura 9.2 Funciones base de la *DCT* bidimensional de 8x8

Los valores de la función base (0,0) de DC resultan del producto de las dos funciones coseno de la ecuación (9.12) evaluadas en $u=0$ y $v=0$, además de ser escalados por los factores $C(u)$, $C(v)$; los valores que corresponden a la Función base DC, son mostrados en la Tabla 9.1.

0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125

Tabla 9.1 Valores Función base DC (0,0)

Ejemplo 9.2

Dada una matriz de datos $A(x,y)$ de la tabla 9.2 que puede corresponder a los valores de un bloque de pixeles de 8×8 de una imagen.

84	84	90	92	88	66	77	86
54	56	72	78	120	124	98	66
55	56	80	120	128	98	74	87
63	55	120	140	245	124	66	88
33	33	45	56	56	87	124	125
88	89	90	100	124	43	46	76
66	56	82	78	33	128	99	67
64	76	74	78	122	121	34	35

Tabla 9.2 Matriz original $A(x,y)$ de 8×8

Para obtener el coeficiente de DC es necesario realizar una suma de productos elemento a elemento de la matriz de datos $A(x,y)$ con la matriz de 8×8 de la función base (0,0). Para obtener la DCT completa de $A(x,y)$ se realiza el mismo procedimiento para las demás funciones base de AC hasta la función (7,7). El procedimiento corresponde a calcular la ecuación (9.12).

Se puede observar que el número de operaciones reales necesarias para el cálculo de cada coeficiente DCT son: 63 sumas y 64 multiplicaciones, es decir 4032 sumas y 4096 multiplicaciones para un bloque de datos de 8×8 .

Para el ejemplo 9.2, todos los coeficientes *DCT* generados se resumen en la tabla 9.3

670.25	-57.49	-106.80	26.83	28.50	-22.82	17.18	16.74
35.51	-20.05	-13.05	-11.75	22.66	-0.28	-16.88	21.21
-36.07	53.30	25.87	19.96	-44.43	8.89	11.50	-30.60
-33.64	11.90	87.80	-42.04	-9.40	26.65	-18.71	-1.95
12.5	-12.85	-14.94	2.77	16.25	-23.86	0.50	17.45
47.67	27.36	-30.65	-12.82	15.01	6.18	13.15	3.77
-0.21	62.21	-11.5	-27.55	49.52	-27.73	-19.37	8.27
-56.74	-55.26	82.42	-20.57	-45.92	51.83	3.31	-44.08

Tabla 9.3 Matriz coeficientes *DCT* de 8x8

El cálculo de la *DCTI bidimensional*, sigue el mismo procedimiento que el de la *DCT*, pero en este caso es necesario utilizar las funciones base que corresponden a la ecuación (9.13). Del ejemplo, se puede observar que para el cálculo de la *DCT* y *DCTI bidimensional* para una matriz de 8x8 se requieren un total de 8128 operaciones (4032 sumas y 4096 multiplicaciones). En aplicaciones de imágenes y video, se pueden utilizar diferentes tamaños de una imagen, por lo tanto dependiendo de este tamaño se tienen muchos bloques de 8x8 a procesar, para imágenes de alta definición el número de operaciones crece extremadamente por lo que se requiere de computadoras de gran potencia para realizar los cálculos.

9.3 ALGORITMOS OPTIMIZADOS PARA LA DCT Y DCTI

El gran número de operaciones que utilizan la *DCT* y *DCTI*, ha generado que los investigadores desarrollen algoritmos que optimizan el desempeño de la transformada, reduciendo el número de operaciones para su ejecución. Se han desarrollado algoritmos rápidos para el cálculo de la *DCT* y *DCTI* unidimensionales y bidimensionales.

Como se ha mencionado en este capítulo, la *DCT* tiene la propiedad de ser una transformación separable, es decir, se puede obtener una *DCT* bidimensional utilizando la *DCT* unidimensional, con este criterio se puede implementar un algoritmo rápido para la *DCT* y *DCTI* unidimensionales y con base en ellos obtener su contraparte bidimensional.

Para el caso unidimensional de la *DCT* optimizada es similar a la transformada rápida de Fourier, un algoritmo optimizado se muestra en la figura 9.3, el algoritmo tiene como entrada ocho muestras de una señal discreta $x(n)$ y salen ocho coeficientes *DCT*. Las flechas indican la dirección en que fluye la información, en el caso en que las flechas estén modificadas por un factor la información se multiplicará por el valor de dicho factor, finalmente cuando dos cantidades se encuentran en un nodo estas se deben sumar.

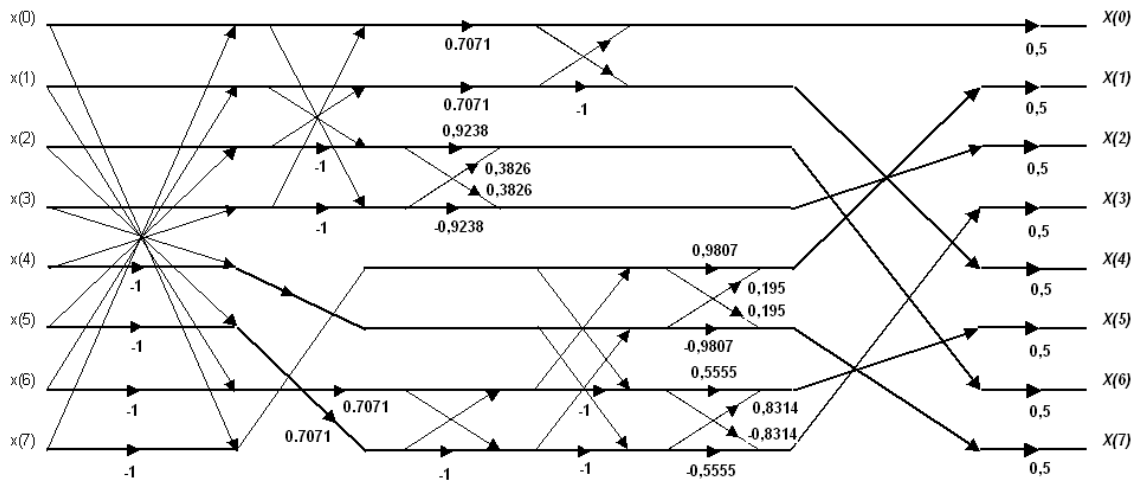


Figura 9.3 Algoritmo DCT unidimensional rápido para N=8.

En la Figura 9.4 se muestra el diagrama de flujo que se utiliza para implementar la *DCTI unidimensional* para un conjunto de ocho muestras, estos dos algoritmos rápidos, fueron desarrollados por Wang, Suehiro y Hatori. En la práctica se han creado otras implementaciones como las hechas por Feing, las cuales requieren operaciones de corrimiento [PEN93], [RAO90].

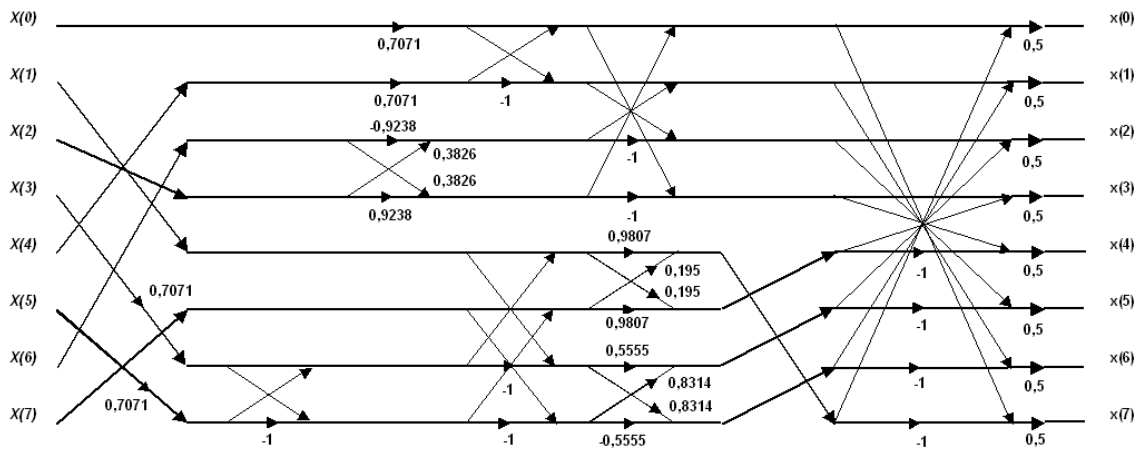


Figura 9.4 Algoritmo DCTI rápido para N=8.

El procedimiento para obtener una *DCT bidimensional* a partir de la utilización de la *DCT optimizada* es el siguiente:

- De la matriz de datos $A(x,y)$ de la tabla 9.2, se toman los elementos del primer renglón, como las muestras de entrada al algoritmo rápido con lo que se obtiene a la salida los coeficientes *DCT unidimensionales*, estos coeficientes

corresponden al primer renglón de una matriz que para fines convencionales se denominada *intermedia*,

- Se aplica la *DCT unidimensional* en los renglones siguientes de la matriz original, con lo cual se obtiene la matriz *intermedia procesada por renglones* (tabla 9.4).

235.85	8.98	-3.66	-14.39	11.66	0.01	-4.22	5.87
236.20	-41.88	-44.06	36.27	-11.31	-5.26	4.47	9.47
246.81	-28.95	-58.14	-0.49	28.99	-5.14	1.89	-1.67
318.59	-28.18	-131.61	21.81	60.46	-45.58	12.04	50.44
197.66	-94.61	26.02	-8.77	-6.71	14.97	-2.74	-1.14
231.96	34.47	-27.33	-15.58	42.43	-23.14	-12.40	20.53
215.34	-26.75	-0.35	13.83	-42.78	35.02	29.61	-29.34
213.57	14.33	-62.91	43.23	-2.12	-35.41	19.94	-6.80

Tabla 9.4 Matriz Intermedia procesada por renglones

- Una vez que se han obtenido todos los renglones de la matriz intermedia, se aplica la *DCT optimizada* tomando como muestras de entrada los elementos de la primera columna de la matriz intermedia, los datos de salida corresponden a la primera columna de la matriz final de coeficientes *DCT*, este proceso continúa hasta que se han procesado todas las columnas de la matriz intermedia.
- El resultado es la matriz final *DCT* que se da en la tabla 9.5, la cual debe tener los mismos valores que la obtenida por el método de las funciones base (tabla 9.3).

670.42	-57.49	-106.80	26.83	28.50	-22.82	17.18	16.74
35.51	-20.05	-13.04	-11.75	22.66	-0.28	-16.88	21.47
-36.07	53.30	25.87	19.96	-44.42	8.89	11.49	-30.60
5-33.65	11.90	87.79	-42.03	-9.41	26.65	-18.71	-1.95
12.50	-12.85	-14.94	2.77	16.25	-23.86	0.50	17.46
47.67	27.36	-30.64	-12.82	15.01	6.18	13.15	3.76
-0.20	62.20	-11.49	-27.55	49.52	-27.72	-19.37	8.27
-56.75	-55.26	82.41	-20.58	-45.93	51.83	3.31	-44.08

Tabla 9.5 Matriz Final DCT $F(u,v)$ de 8x8

Las matrices de las tablas 9.3 y 9.5 son idénticas, aunque si se comparan minuciosamente difieren debido a que la precisión utilizada por el algoritmo rápido es menor. En este ejemplo la *DCT optimizada* se aplicó primero a los renglones y luego a las columnas, el orden de aplicación es arbitrario ya que también es posible procesar primero las columnas y después los

renglones. Para obtener una *DCTI bidimensional* se sigue el mismo procedimiento utilizando el algoritmo de la figura 9.4.

Utilizando los algoritmos optimizados tanto para la *DCT*, como para la *DCTI* es posible reducir de manera considerable el número de operaciones como se enuncia a continuación:

- Para la *DCT unidimensional* optimizada (algoritmo de Wang, Suehiro, Hatori) solo se necesitan 26 sumas y 16 multiplicaciones, para un conjunto de ocho muestras, reduciendo el porcentaje de operaciones hasta el 35%.
- Para la *DCTI unidimensional optimizada* se necesitan un total de 27 sumas y 16 multiplicaciones, para un conjunto de ocho muestras, reduciendo el porcentaje de operaciones hasta el 35.83%.
- Utilizando el criterio descrito en esta sección para el cálculo de la *DCT bidimensional*, solo se necesitan un total de 416 sumas y 256 multiplicaciones, para un bloque de 8x8, reduciendo el porcentaje de operaciones hasta el 8.267%.
- De igual manera para obtener la *DCTI bidimensional* se necesitan 432 sumas y 256 multiplicaciones, para un bloque de 8x8, reduciendo el porcentaje de operaciones hasta el 8.464%.

9.4 EJEMPLO DE APLICACIÓN

En esta sección se incluye un ejemplo de compresión de imágenes aplicando la DCT, en principio se presentan algunos programas con el código en Matlab para calcular la DCT y el proceso de compresión de una imagen. Para realizar el proceso de compresión completo se trabaja con una imagen de 400x400 pixeles, se calcula la de la imagen aplicándola a bloques de 8x8, se cuantizan los coeficientes de la DCT y se recortan a cero aquellos que no rebasen un umbral, se contabiliza la cantidad de memoria que ocupan los coeficientes diferentes de cero, se aplica la DCTI y se rescata la imagen. Los resultados se observan en la figura 9.5.

PROGRAMA DE LA DCT

```

%
%   TRANSFORMADA COSENO
%
%   @Larry

IM = imread('1.bmp');           % lee una imagen de un archivo
[M N] = size(IM);
F= zeros(N,N);
I=I2;
for u=1:M
    u1=u-1;
    for v=1:N
        A=0;
        v1=v-1;
        for x=1:M
            x1=x-1;
            for y=1:N
                y1=y-1;
                A= A + IM(x,y)*cos((2*x1+1)*pi*u1/(2*M))*cos((2*y1+1)*pi*v1/(2*N));

                end    % // y, columnas en I(x,y)
            end        % // x, filas en I(x,y)
            if u == 0 then
                C1 = (1/N)^0.5
            else
                C1 = 1 ;
            end
            if v == 0 then
                C2 = (1/N)^0.5
            else
                C2 = 1 ;
            end
            F(u,v) = 2*A*C1*C2/N;
        end            % // v, columnas en F(u,v)
    end                %//u , filas en F(u,v)
end
F=floor(F)

```

PROGRAMA DE COMPRESIÓN DESCOMPRESIÓN DE UNA IMAGEN UTILIZANDO LA DCT Y DCTI

```

%
%
%   @Larry
%
clear;

% IM = imread('1.bmp');
IM = imread('camilo.bmp');    % lee imagen
[M N] = size(IM);
D = 8;
A = [];
B = [];
M = M/D;
N = N/D;
NC1 = 0;           % Contador de Ceros de Cuantización
for m = 1:M
    for n = 1:N
        for x=1:D
            for y = 1:D
                B(x,y)=IM((m-1)*D+1+x-1, (n-1)*D+1+y-1);
            end
        end
        % OBTENER DCT DE B --->> BEs(u,v)
        % Bdct = dct88(B)
        Bdct = fun_dct_2d(B);
        % CUANTIZAR BEs --->> BEsQ
        % BEsQ = cuantiza(Bdct);
        BQ = cuantiza8(Bdct,600);
        NC1 = NC1 + num_cero(BQ);
        % OBTENER DCTI DE BEsQ
        % Bdcti = dcti88(BEsQ);
        Bo = idct2(BQ);
        % ORDENAR IMAGEN DE SALIDA
        for x=1:D
            for y = 1:D
                IMS((m-1)*D+1+x-1, (n-1)*D+1+y-1)=Bo(x,y);
            end
        end
    end
end
IMS = uint8(IMS);
imshow(IMS);

COMP= (N*D*M*D-NC1)/(N*D*M*D)*100

```

En la figura 9.5 se ha utilizado un umbral arbitrario para hacer cero los coeficientes de la DCT, como se observa, en la medida que se tiene mayor compresión la calidad visual en la recuperación de la imagen original se deteriora y se agregan artefactos tipo cuadrícula.



Imagen original



Imagen comprimida al 50%



Imagen comprimida al 25%



Imagen comprimida al 15%

Figura 9.5 Imagen original e imágenes comprimidas

9.5 RESUMEN

En este capítulo se ha analizado la transformada Coseno discreta directa e inversa, tanto en una dimensión como en dos dimensiones y se ha presentado una aplicación concreta referida a la compresión de imágenes. Se muestra el código utilizado en Matlab para la DCT, como se ve en las imágenes presentadas, en la medida que aumentamos la compresión de una imagen se ha deteriorado la calidad visual cuando se recupera, por tanto el diseñador tiene que tomar decisiones en sacrificar calidad por compresión y viceversa. Se enfatiza que la DCT se ha utilizado ampliamente en estándares de compresión de voz, audio, imágenes y video [VIT07].

9.6 PROBLEMAS PROPUESTOS

1. Desarrollar un programa para aplicar la DCT unidimensional a una señal $x(n)$ discreta, proponer la cantidad de funciones base a utilizar para calcular los coeficientes X_c , luego aplicar la DCTI y verificar la recuperación de $x(n)$, evaluar porcentajes de error en la recuperación de la señal dependiendo del número de funciones base utilizadas.
2. Programar todas las subrutinas faltantes para el programa compresor de imágenes de este capítulo.

CAPITULO 10

FORMATOS NUMÉRICOS BINARIOS

Cuando un sistema de procesamiento digital de señales (PDS) es implementado en una arquitectura dedicada como un procesador digital de señales (DSP), debe existir un compromiso entre el intervalo dinámico de las variables y la precisión de las mismas. Esto también es válido en procesadores de punto flotante, pero es específicamente cierto para procesadores de punto fijo.

Al efectuar operaciones aritméticas en un sistema de PDS representado en formatos numéricos de precisión finita se tienen tres tipos de errores:

- Efecto de conversión de una señal analógica a digital.
- La representación de los coeficientes.
- El truncamiento de los resultados cuando se efectúan productos para reajustar el ancho de palabra cuando se almacena el resultado en L bits.

Los efectos de precisión numérica son no lineales y aleatorios, por lo que su análisis es muy complejo.

Como se ha abordado durante el transcurso de este libro, en PDS existen operaciones básicas tales como suma, multiplicación y retardos; cuando se implementa en una computadora digital un sistema de PDS, y sobre todo para que opere en tiempo real, es necesario conocer como se realizan las operaciones a nivel digital. En este capítulo se abordan los formatos numéricos binarios para la representación de constantes y variables numéricas, la realización de operaciones tales como sumas, multiplicaciones y corrimientos, se analizan los formatos más utilizados, sus dinámicas, precisiones y cómo se realizan las operaciones básicas.

En cualquier sistema digital o una computadora, los números son representados como una combinación finita de números binarios, o bits que toman valores "0" y "1", lo que ocasiona errores por los efectos de precisión finita debido a los efectos de cuantización. Los bits son organizados en conjuntos de ocho llamados *bytes* o 16 bits llamados *words*. La representación más frecuente de los números en una computadora es a través de dos tipos de notación o formatos: *de punto fijo* y *de punto flotante*.

10.1 FORMATO NUMÉRICO DE PUNTO FIJO

La representación numérica en formato digital de punto fijo es similar a la representación de números decimales, es decir como un conjunto de números con un punto decimal, la representación de números decimales en forma binaria se puede realizar en general de tres formas [PRO92]:

1. *Signo magnitud*, el bit más significativo (MSB) representa el signo y el resto de bits la magnitud del número o variable.
2. *Complemento a uno*, el número o variable se representa en forma complementada a uno.
3. *Complemento a dos*, es la forma más utilizada en los cálculos aritméticos en la mayoría de las computadoras, el número o variable se representa en complemento a dos. En este formato se pueden representar números con parte entera y fraccionaria, algunas veces se le conoce como formato Q_i .

Comúnmente en la utilización del formato en punto fijo se representa utilizando el formato llamado " Q_i ", donde "i" significa el número de bits para la parte fraccionaria. En el formato Q_i la cantidad total de bits L de la palabra digital se particiona en:

QE: los bits para la parte entera
 QF= Q_i : los bits para la parte fraccionaria
 S: el bit de signo

es decir que

$$L = 1 + QE + QF \quad (10.1)$$

como se observa en la figura 10.1 entre la parte entera y la parte fraccionaria existe un punto hipotético o *punto fijo*, que sólo lo interpreta el programador.

En general un número X o variable positiva se puede representar en formato de punto fijo como

$$X = \sum_{i=-QE}^{QF} b_i r^{-i} = (b_{-QE}, \dots, b_{-1}, b_0, \dots, b_{QF}) \quad 0 \leq b_i \leq (r-1) \quad (10.2)$$

donde

b_i representa el dígito binario en la posición correspondiente
 r la base igual a 2 para el caso binario.

Además es necesario tener presente que el formato puede tener diferentes formas de interpretar los datos, como no signado, signo magnitud, signo con complemento a uno y complemento a dos, siendo la forma más común de representar a los datos en complemento a dos debido a la eficiencia de los cálculos [PRO92].

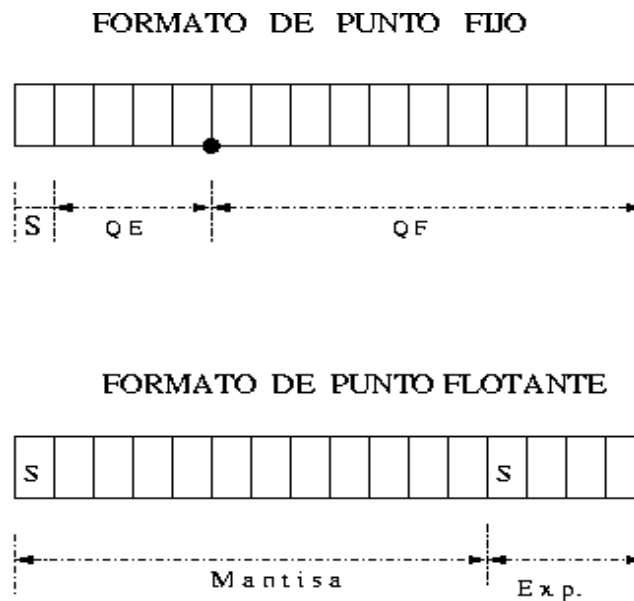


Figura 10.1 Formatos numéricos

Representación de un número real "X" en punto fijo en complemento a dos para $L=QE+QF+1$ se puede escribir como [ESC00]

$$X = (S)^{QE+QF} + \sum_{i=-QE+1}^1 b(QF+|i|)2^{-i} + \sum_{i=1}^{QF} b(QF-i)2^{-i} \quad (10.3)$$

Donde el bit más significativo (MSB) corresponde al signo, $S = 0$ para números positivos, $S = 1$ para números negativos y $b(.)$ es un número binario "0" o "1" en la posición correspondiente.

Ejemplo 10.1

Para el formato de punto fijo Qi expuesto anteriormente, se pueden interpretar los números binarios:

$$0101.1001 \quad \text{y} \quad 1110.1100$$

de la siguiente forma

$$0101.1001 = 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 + 1x2^{-1} + 0x2^{-2} + 0x2^{-3} + 1x2^{-4} = 5.5625$$

$$1110.1100 = -1x2^3 + 1x2^2 + 1x2^1 + 0x2^0 + 1x2^{-1} + 1x2^{-2} + 0x2^{-3} + 0x2^{-4} = -1.25$$

En los formatos de representación numérica es importante tener presente algunos parámetros del sistema para lograr mejores desempeños, estos son, *el intervalo dinámico de las variables, la precisión numérica y la resolución*:

- El intervalo dinámico de un sistema numérico es definido como la diferencia entre el número mayor y el número menor que se pueda expresar en el formato.
- La precisión numérica de un sistema se define como la diferencia entre dos números consecutivos, y es determinado por el valor del bit más a la derecha, es decir, el bit menos significativo (LSB). En el ejemplo 10.1 la precisión numérica está dada por 2^{-4} o 0.0625.
- Resolución, si se utilizan L bits para representar una variable X con un valor máximo X_{max} y un mínimo X_{min} , entonces la resolución es [PRO92].

$$\Delta = \frac{X_{max} - X_{min}}{2^L - 1} \quad (10.3)$$

Por lo tanto en formato de punto fijo debe existir un compromiso entre la precisión y el intervalo dinámico de los números o variables a representar. Es decir, si el intervalo dinámico es muy grande, la precisión no es muy buena y viceversa. Una característica del formato en punto fijo es que su resolución es fija, y el incremento de la resolución es proporcional al incremento del intervalo dinámico.

INTERVALO DINÁMICO Y PRECISIÓN NUMÉRICA

Para el formato Qi, el intervalo dinámico (ID) se puede escribir en función de los valores QE y QF como

$$-2^{QE} < \text{ID} < 2^{QE} - 2^{-QF} \quad (10.4)$$

y la precisión

$$p = 2^{-QF} \quad (10.5)$$

Ejemplo 10.2

Intervalo dinámico y precisión numérica de algunos formatos Qi

- Formato Q0
 $-32,768 < ID < 32,767$
 $p = 1$
- Formato Q15
 $-1 < ID < 0.9999$
 $p = 0.00003051$
- Formato Q12
 $-8 < ID < 7.9999$
 $p = 0.000244$
- Formato Q8
 $-128 < ID < 127.9999$
 $p = 0.0039063$

10.2 TRUNCAMIENTO Y REDONDEO

Cuando se representa una variable que pertenece a los números reales en un formato numérico binario de precisión finita, no se puede representar toda la parte decimal de la variable, es decir que los decimales menos significativos pueden perderse en la representación. En este proceso la pérdida de precisión se puede dar a través de dos formas: por *truncamiento* y *redondeo*.

- **Truncamiento:** un número real en precisión infinita es convertido a formato de precisión finita utilizando los bits disponibles, es decir, que existe un truncamiento de los decimales menos significativos a representar.
- **Redondeo:** se le agrega un "1" binario en una posición más allá del bit LSB a preservar y luego se trunca. Si el bit adicional es "1" entonces el bit LSB se incrementará en uno, si el bit adicional es "0", entonces el redondeo será similar al truncamiento.

Los procesos de truncamiento y redondeo se pueden emplear en asignación de variables y después de efectuar operaciones aritméticas para guardar los resultados en memoria. El proceso de redondeo es más preciso pero requiere más operaciones.

10.3 OPERACIONES BINARIAS

Como se mencionó al inicio de este capítulo, las operaciones aritméticas básicas del PDS que se definieron en el capítulo cuatro son las mismas que realiza una computadora digital, por tanto un diseñador de sistemas de PDS y sobre todo si los sistemas van a operar en tiempo real, debe conocer los recursos del sistema y como se realizan las operaciones digitales a nivel de hardware. En esta sección se estudian las operaciones básicas que realiza una computadora y que están fuertemente relacionadas con el PDS, entre ellas se tiene, la asignación de variables, la suma, la multiplicación y corrimientos.

10.3.1 ASIGNACIÓN DE VARIABLES

En la asignación de variables el programador debe estar conciente del formato de representación para que la operación de asignación tanto para cargar o salvar variables sea coherente y reproduzca el formato deseado. Es por eso que en un DSP existen varias unidades de corrimientos que permiten flexibilizar estas operaciones [TI50]. En la figura 10.2 se observan dos variables X e Y en formatos Q_{ix} y Q_{iy} respectivamente, si se escribe X en la localidad de Y y se requiere representarla en el formato de Y , entonces hay que efectuar los corrimientos $Q_{ix} - Q_{iy}$ a la derecha y por lo tanto tenemos más bits para representar la parte entera, pero existe un truncamiento en los bit LSB, por lo tanto existe una pérdida de precisión, pero un aumento en el intervalo dinámico de X . El caso contrario es cuando la variable Y se quiere escribir con el mismo formato de X , en este caso se tiene un problema de sobreflujo, ya que la parte entera de Y no cabe en el formato que tiene la variable X .

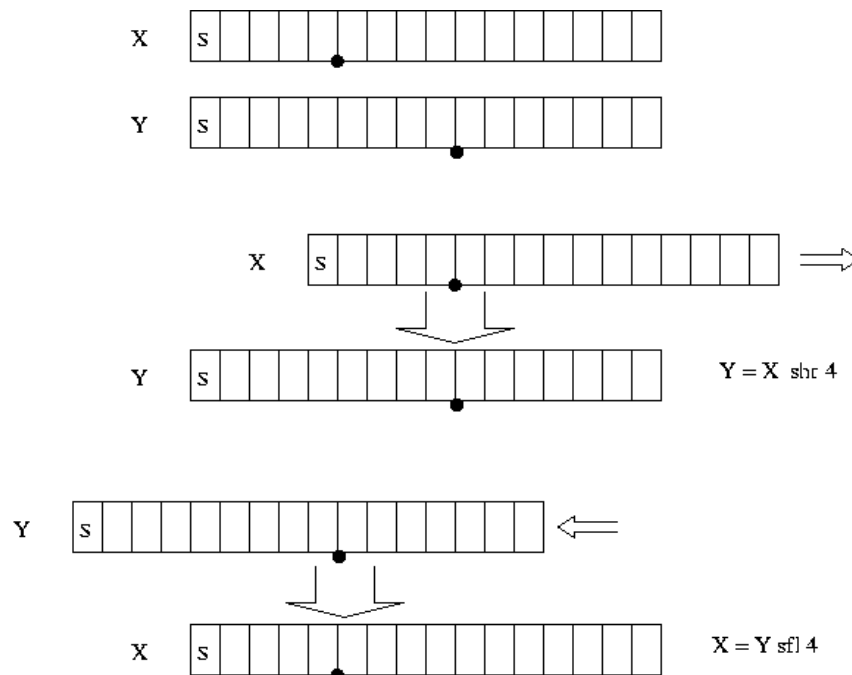


Figura 10.2 Asignación numérica en formato de punto fijo

10.3.2 OPERACIÓN SUMA EN PUNTO FIJO

Para efectuar una suma entre dos variables representadas en punto fijo se efectúan los corrimientos necesarios en una o las dos variables para alinear el punto fijo y posteriormente efectuar la suma, es decir tener las dos variables en el mismo formato Q_i . Obviamente hay que tener cuidado en el formato de representación de las variables y del resultado de la suma para evitar el sobreflujo (figura 10.3).

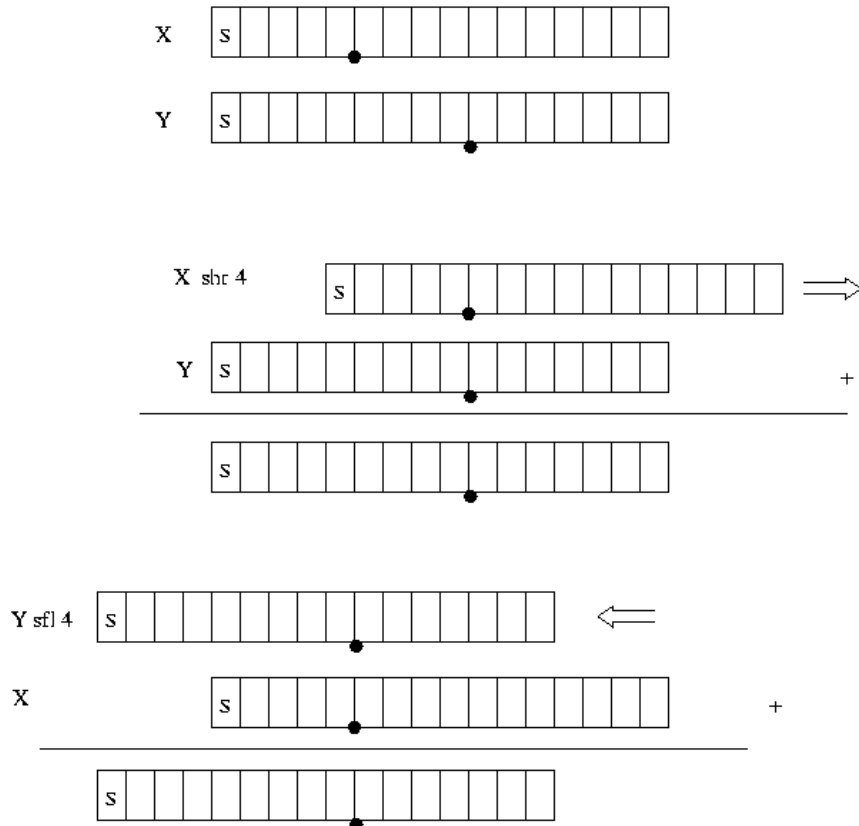


Figura 10.3 Suma en Aritmética de punto fijo

10.3.3 OPERACIÓN MULTIPLICACIÓN EN PUNTO FIJO

Cuando se efectúa una multiplicación de dos variables representadas en “L” bits, el resultado queda en “2L” bits. Si se multiplican dos valores con formato Q_{ix} y Q_{iy} el resultado queda en $Q_{(ix+iy)}$, además se generan dos bits de signo (figura 10.4).

En la figura 10.4 se ha realizado la multiplicación entre la variable X en Q_{11} y la variable Y en Q_7 , ambas con la misma longitud de palabra $L=16$ bits, el resultado de la multiplicación queda en Q_{18} con doble signo S, para efectos de salvar el resultado a memoria se puede

eliminar un signo, salvar la parte alta en longitud de palabra L y el resultado quedaría en formato Q2. En algunos DSPs existe una forma de programación de la salida del multiplicador para correr un bit a la izquierda y eliminar un bit de signo, y así ajustar la salida a formato Q15, o también se tiene la posibilidad de un corrimiento de cuatro bits a la izquierda para trabajar en formato Q12 [TI50].

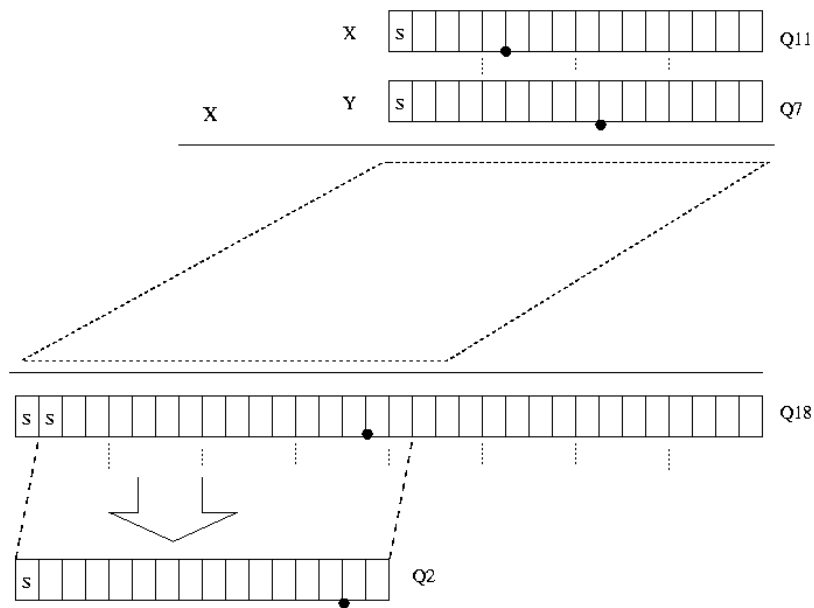


Figura 10.4 Multiplicación en Aritmética de punto fijo

En resumen cuando se realizan operaciones aritméticas en punto fijo, es necesario hacer una simulación para determinar los valores máximos y mínimos que toman los resultados de la aplicación para determinar la dinámica de las variables y elegir los formatos numéricos adecuados. La aritmética de punto fijo es la más utilizada en DSPs debido a su velocidad de operación y bajo consumo de potencia.

10.4 FORMATO NUMÉRICO DE PUNTO FLOTANTE

El formato numérico en punto flotante puede ser empleado para cubrir un amplio intervalo dinámico, sin embargo, la resolución decrece con un incremento en el tamaño de los números sucesivos, es decir que la distancia entre números sucesivos en punto flotante se incrementa. La resolución variable resulta en un gran intervalo dinámico [PRO92].

La representación numérica en punto flotante también es llamada *notación real*, cualquier número X puede expresarse como

$$X = M2^E \quad (10.6)$$

donde: M es la *mantisa*, E el *exponente* y se expresan en forma binaria tal como se tiene en la figura 10.5, es decir, si se tiene un ancho de palabra L , se asignan b_M bits para la mantisa con su signo y b_E bits para el exponente que puede ser negativo o positivo, entonces $L=b_M+b_E$, la mantisa es representada en forma fraccionaria y está en el intervalo $1/2 \leq M < 1$.

Algunos formatos en punto flotante utilizan un *bit implícito* igual a uno que se agrega a la mantisa para ganar un bit de precisión. Esto consiste en que un número expresado en una palabra binaria de longitud L se hacen los corrimientos necesarios del número para obtener la mantisa normalizada, entre $1/2 \leq M < 1$ y esos corrimientos se toman en cuenta para escribirlos en el campo del exponente, en un número positivo, el primer uno original era el valor más significativo y éste siempre va a existir, por lo que pasa a ser el bit implícito.

El programador del hardware decide la ubicación de la mantisa y el exponente dentro de la palabra digital, o su representación en varias palabras, comúnmente se utiliza la proporción $b_M=3L/4$. El número de bits para el exponente determinan el intervalo dinámico y el número de bits de la mantisa.

10.4.1 SUMA EN PUNTO FLOTANTE

Para efectuar la suma entre dos variables representadas en formato de punto flotante se requiere que los exponentes sean iguales, es decir, hay que hacer un ajuste de uno de los exponentes. Esto se realiza por el corrimiento de la mantisa del número más pequeño a la derecha compensando el incremento de su exponente. En este proceso de reajuste del exponente se tiene una pérdida de precisión. En resumen, la representación numérica en punto flotante provee una resolución fina para números pequeños, pero para números grandes la resolución es burda, en contraste con el formato en punto fijo que provee una resolución uniforme a través de todo el intervalo de valores para un mismo Q_i .

10.4.2 MULTIPLICACIÓN EN PUNTO FLOTANTE

En este caso se multiplican las mantisas y los exponentes se suman. Puede existir sobreflujo cuando la suma de los exponentes excede el intervalo dinámico de la representación en punto fijo del exponente.

10.4.3 DIVISIÓN EN PUNTO FLOTANTE

Se dividen las mantisas normalizadas y se restan los exponentes. En este caso también puede haber problemas en la representación del exponente del resultado.

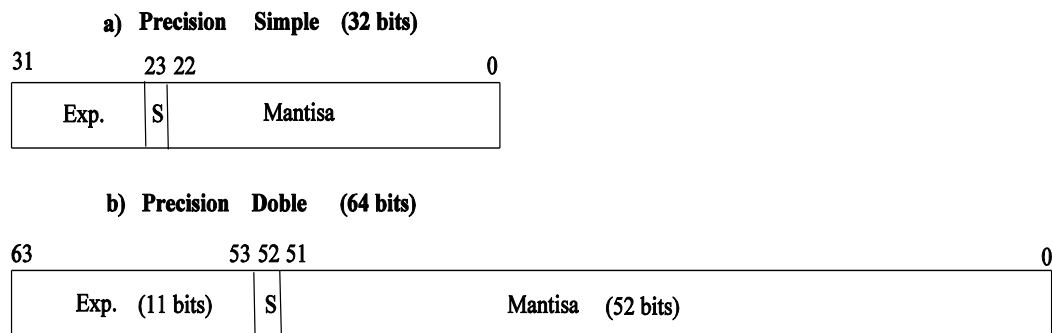
El formato de precisión doble es similar sólo que para la mantisa se utilizan 52 bits más el bit implícito, el exponente se representa en 11 bits con desplazamiento aditivo de 1023.

10.6 FORMATO DE MICROSOFT

En este formato el exponente tiene una desviación aditiva (offset) de 129, por lo tanto para obtener el exponente verdadero, se debe restar al exponente de 129. El signo de la mantisa está en el bit 23 y la mantisa tiene un bit implícito entre los bits 22 y 23 en precisión simple [GOD91].

Para el formato de precisión doble la mantisa se utilizan 52 bits, el exponente se representa en 11 bits con desplazamiento aditivo de 1025, como se observa en la figura 10.6.

$$X = (-1)^s M 2^{E-129} \quad (10.9)$$



10.7.1 FORMATOS DE PUNTO FLOTANTE DE TI

Todos los formatos de punto flotante del TMS320C30 consisten de tres campos:

- exponente (E) : es un número en complemento a dos.
- signo (S) de la mantisa.
- mantisa (M) : el campo de signo y mantisa son considerados como una unidad y se refiere como campo de mantisa.

La mantisa: está compuesta de una fracción en complemento a dos combinada con el bit de signo y un uno del bit más significativo que está implícito. En la representación normalizada, existe el bit más significativo no signado implícitamente, esto se hace con el objeto de proveer un bit más de precisión.

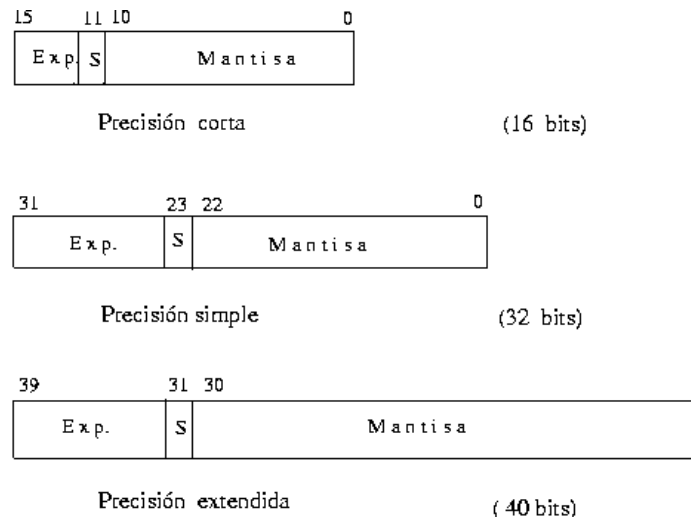


Figura 10.7 Formatos numéricos en punto flotante de TI

10.7.2 FORMATOS DE PUNTO FLOTANTE SOPORTADOS POR DSPs TMS320C30 Y TMS320C40

1) Formato corto de punto flotante

Un número en punto flotante es representado en complemento a dos en 16 bits, donde el campo para el exponente E es de 4 bits, la mantisa en complemento a dos está en 12 bits con el bit más significativo no signado implícito.

El valor de un número X en punto flotante corto en complemento a dos está dado por:

$$\begin{aligned} X &= 01.M x 2^E & \text{si } S &= 0 \\ X &= 10.M x 2^E & \text{si } S &= 1 \\ X &= 0 & \text{si } E &= -8 \end{aligned}$$

Las operaciones son efectuadas con un bit implícito entre los bits 10 y 11. Cuando el bit MSB implícito se hace explícito, este se localiza inmediatamente a la izquierda del punto binario.

Intervalos de precisión:

$$\begin{aligned} \text{Más positivo : } & X = (2-2^{-11}) x 2^7 = 2.5594 x 10^2 \\ \text{Menos positivo: } & X = 1 x 2^{-7} = 7.8125 x 10^{-3} \\ \text{Menos negativo: } & X = (-1-2^{-11}) x 2^{-7} = -7.8163 x 10^{-3} \\ \text{Más negativo: } & X = -2 x 2^7 = -2.56000 x 10^2 \end{aligned}$$

Se deben reservar los siguientes valores para representar el cero en este formato es:

$$\begin{aligned} E &= -8 \\ S &= 0 \\ M &= 0 \end{aligned}$$

2) Formato de precisión simple de punto flotante

Un número en punto flotante es representado por un exponente en un campo de 8 bits y la mantisa en complemento a dos en un campo de 24 bits con el MSB implícito no signado. El valor de un número X en punto flotante de precisión simple en complemento a dos está dado por:

$$\begin{aligned} X &= 01.M x 2^E & \text{si } S &= 0 \\ X &= 10.M x 2^E & \text{si } S &= 1 \\ X &= 0 & \text{si } E &= -128 \end{aligned}$$

Las operaciones son efectuadas con un bit implícito entre los bits 22 y 23. Cuando el bit MSB implícito se hace explícito, éste se localiza inmediatamente a la izquierda del punto binario.

Intervalos de precisión:

$$\begin{aligned} \text{Más positivo : } & X = (2-2^{-23}) x 2^{127} = 3.4028234 x 10^{38} \\ \text{Menos positivo: } & X = 1 x 2^{-127} = 5.8774717 x 10^{-39} \\ \text{Menos negativo: } & X = (-1-2^{-23}) x 2^{-127} = -5.8774724 x 10^{-39} \\ \text{Más negativo: } & X = -2 x 2^{127} = -3.4028236 x 10^{38} \end{aligned}$$

Se deben reservar los siguientes valores para representar el cero en este formato es:

$$E = -128$$

$$S = 0$$

$$M = 0$$

3) Formato de precisión extendida de punto flotante

Un número en punto flotante es representado por un exponente en un campo de 8 bits y la mantisa en complemento a dos en un campo de 32 bits con el MSB implícito no signado. El valor de un número X en punto flotante de precisión extendida en complemento a dos está dado por:

$$\begin{array}{lll} X = 01.M \times 2^e & \text{si} & S = 0 \\ X = 10.M \times 2^e & \text{si} & S = 1 \\ X = 0 & \text{si} & E = -128 \end{array}$$

Las operaciones son efectuadas con un bit implícito entre los bits 31 y 30. Cuando el bit MSB implícito se hace explícito, éste se localiza inmediatamente a la izquierda del punto binario.

Intervalos de precisión:

$$\begin{array}{ll} \text{Más positivo :} & X = (2-2^{-31}) \times 2^{127} = 3.4028236683 \times 10^{38} \\ \text{Menos positivo:} & X = 1 \times 2^{-127} = 5.8774717541 \times 10^{-39} \\ \text{Menos negativo:} & X = (-1-2^{-31}) \times 2^{-127} = -5.87747175569 \times 10^{-39} \\ \text{Más negativo:} & X = -2 \times 2^{127} = -3.402823669 \times 10^{38} \end{array}$$

Se deben reservar los siguientes valores para representar el cero en este formato es:

$$E = -128$$

$$S = 0$$

$$M = 0$$

10.8 OTROS FORMATOS

Intel diseñó coprocesadores matemáticos (familia 80xx7) de tal forma que pudieran manejar operaciones aritméticas con un formato de 80 bits con un intervalo dinámico de $10^{-4,932}$ a $10^{4,932}$ [GOD91]. La dificultad de este formato es la conversión adecuada para utilizar los resultados de las operaciones por otros dispositivos en aplicaciones reales.

10.9 RESUMEN

En este último capítulo se ha incluido la representación numérica en forma binaria, ya que la intención final del diseño de un sistema de PDS es probarlo e implementarlo en alguna arquitectura digital. Cuando se simula una aplicación en algún programa de PDS, normalmente todas las operaciones se realizan en aritmética de punto flotante con precisión doble, donde las dinámicas y precisiones numérica son muy grandes, sin embargo, en los sistemas reales las longitudes de palabras son pequeños por razones de tamaño y ahorro de energía en la actualidad se utilizan 16 bits, por lo tanto las aplicaciones se deben restringir a estas limitaciones. Por estas razones el diseñador de sistemas de PDS debe tener conocimientos del manejo de formatos numéricos y las arquitecturas digitales existentes, el lector puede encontrar implementación de aplicaciones a 16 bits en punto fijo en [ESC05].

10.10 PROBLEMAS PROPUESTOS

1. Dado el número π representarlo y escribirlo en los formatos:

Fijo con $L=16$ bits y $Q12$
Flotante IEEE 754 a 32 bits
Flotante de Microsoft a 32 bit

2. Realizar un programa en cualquier lenguaje que permita representar cualquier número en L bits en un formato Q_i .
3. Realizar un programa en cualquier lenguaje que efectúe la conversión entre formatos Q_i .
4. Realizar un programa en cualquier lenguaje que emule las operaciones suma, multiplicación y corrimiento utilizando aritmética de punto fijo.
5. Realizar un programa en cualquier lenguaje que efectúe la operación convolución entre dos secuencias discretas en punto fijo, el programa debe de elegir la mejor precisión numérica para representar las secuencias y realizar las operaciones, además el resultado lo debe mostrar en el formato que conserve la mejor precisión.

BIBLIOGRAFÍA

- [ALC98] Alcántara S. R. & Escobar S. L. *Dynamic Range and Scaling Evaluation in Adaptive Filtering Algorithms for DSP Fixed-Point Implementation*. International Symposium on Information Theory and its Applications ISITA98. México, D.F., Octubre 14-16, 1998.
- [ALC99] Alcántara S. R. & Escobar S. L. *A Comparative TMS DSP Fixed-Point Implementation of FRLS Adaptive Filtering Algorithms Family*. The International Conference on Signal Processing Application and Technology and DSP World Workshop, ICSPAT99. Orlando, Florida, USA, noviembre 1999.
- [BRI88] Brigham E. O. *The Fast Fourier Transform and its applications*. Prentice Hall. USA 1988.
- [CHA02] Chassaing R. *DSP Applications Using C and TMS320C6x DSK*. John Wiley & Sons, Inc. USA 2002
- [CHU77] Churchill R. V. *Series de Fourier y problemas de contorno*. McGraw-Hill, México 1977.
- [ESC00] Escobar S. L. *Arquitecturas de DSP's, Familia TMS320 y el TMS320C50*. Facultad de Ingeniería UNAM, México, D.F., agosto 2000.
- [ESC00_1] Escobar S. L. & Alcántara S.R. *Fixed Point Arithmetic Using Digital Signal Processors*. International Conference on Telecommunications, ICT2000, Acapulco, Guerrero., México, mayo 2000.
- [ESC02] Escobar S. L. *Laboratorio de DSP's Familias TMS320C5x y TMS32054x*. Facultad de Ingeniería UNAM, junio 2002.

-
- [ESC03] Escobar S. L. *Cancelación de Ruido y Eco Utilizando Filtrado Adaptable*. Electro 2003, Instituto Tecnológico de Chihuahua, Chih., México, octubre 2003.
- [ESC97] Escobar S. L. *Algoritmos de Filtrado Adaptable: Implementación, Evaluación, Comparación y Aplicaciones en Telecomunicaciones*. Tesis de Maestría en Ingeniería Eléctrica, UNAM, México, D.F., noviembre 1997.
- [ELL82] Elliot D. F & Rao R. *Fast Transforms algorithms, advantage and applications*. Academic Press, Orlando Florida, USA 1982.
- [GOD91] Godfrey J. T. *Lenguaje ensamblador para microcomputadoras IBM*. Prentice Hall, México 1991.
- [HSU87] Hsu H. P. *Análisis de Fourier*. Addison-Wesley Iberoamericana. USA 1987.
- [KUU01] Kuo S. M. & Lee B. M. *Real Time Digital Signal Processing*. John Wiley & Sons. Great Britain 2001.
- [KUU05] Kuo S. M. & Gan. *Digital Signal Processors, Architectures, Implementations and Applications*. Pearson Prentice-Hall. New Jersey 2005.
- [MOT56] *Fractional and Integer Arithmetic Using the DSP56000 Family of General-Purpose Digital Signal Processors*. Motorola, USA 1988.
- [OPP99] Oppenheim A.V., Schaffer R. W. & Buck J. R. *Discrete Time Signal Processing*. Prentice-Hall, USA, 1999.
- [PEN93] Pennebaker W. B., Mitchell J. L, Chapman & Hall. *JPEG still data compression standard*. Digital multimedia standards, New York 1993.
- [PRO92] Proakis J. G. & Manolakis. *Digital Signal Processing, Principles, Algorithms and Applications*. Macmillan. Singapore 1992.
- [PRO95] Proakis J. G. *Digital Communications*. 3rd. Edition, McGraw-Hill, Singapore, 1995.
- [PSE98] Psenicka B. & Escobar S. L. *Procesamiento Digital de Señales, Segunda parte, Microprocesadores y Realización de los Filtros digitales con TMS320Cxx*. Facultad de Ingeniería, UNAM, julio 1998.
-

- [RAO90] Rao K. R. & Yip P. *Discrete cosine transform algorithms, advantage and applications*. Academic Press, San Diego, CA. USA 1990.
- [HER04] Hernández R. J. *Compresión de imágenes utilizando la transformada Coseno en un DSP*. Tesis de licenciatura, Facultad de Ingeniería, UNAM, México D. F. 2004.
- [TI30] Texas Instruments. *User Guide, TMS320C3x*. USA 1994.
- [TI40] Texas Instruments. *User Guide, TMS320C4x*. USA 1993.
- [TI50] Texas Instruments. *User Guide, TMS320C5x*. USA 1997.
- [TI6000] Texas Instruments. *TMS320C6000 CPU and Instruction Set, Reference Guide*. USA 2000.
- [TI89] Texas Instruments. *Digital Signal Processing. Applications with the TMS320 Family, Theory, Algorithms and Implementation*, Vol. 1, 2 y 3. USA, 1989.
- [VIT07] Vitela R. M. *Compresión MJPEG de Video digital en un DSP*. Tesis de Maestría. F.I. UNAM. México 2007.
- [WYL94] Wylie C. R. *Matemáticas superiores para Ingeniería*. McGraw-Hill. México 1994.