

Procesamiento Digital de Señales

Dr. Félix Calderón Solorio

17 de marzo de 2017

Índice general

Antecedentes	1
1.1. Definiciones	1
1.2. Señales pares e impares	2
1.2.1. Ejemplos.	2
1.3. Señales básicas de tiempo continuo.	5
1.3.1. Función impulso unitario	5
1.3.2. Función escalón unitario	6
1.3.3. Exponencial compleja y senoidal.	6
1.4. Señales Periodicas	8
1.4.1. Ejemplos	8
Sistemas	11
2.1. Introducción	11
2.1.1. Sistemas inversos.	12
2.1.2. Sistemas causales.	12
2.1.3. Estabilidad.	13
2.2. Invariancia en el tiempo.	13
2.2.1. Linealidad	14
2.3. Ejemplos	14
2.3.1. Ejemplo	14
2.3.2. Ejemplo	15
2.3.3. Ejemplo	15
Convolución	17
3.1. Correlación	17
3.1.1. Propiedades	18
3.1.2. Ejemplo	19
3.1.3. Correlación en dos dimensiones	20
3.2. Representación de señales en términos de impulsos.	20
3.3. Convolución	21

3.3.1.	Propiedades	21
3.3.2.	Sucesión útil	23
3.3.3.	Ejemplos	25
3.4.	Convolución en dos dimensiones	27
3.4.1.	Separabilidad	27
3.5.	Algunos kerneles interesantes	28
3.5.1.	Suavizadores	28
3.5.2.	Derivadas	30
3.6.	Respuesta Sistemas LIT	36
Transformada de Fourier		39
4.1.	Representación de señales periódicas.	39
4.2.	Cálculo de la Transformada Discreta de Fourier	40
4.2.1.	Ejemplo	42
4.2.2.	Ejemplo	43
4.3.	Implementación de la Transformada de Fourier	45
4.4.	Propiedades de la Transformada Discreta de Fourier	47
4.4.1.	Periodicidad	47
4.4.2.	Linealidad	48
4.4.3.	Desplazamiento en tiempo	48
4.4.4.	Desplazamiento en frecuencia	49
4.4.5.	Conjugación	49
4.4.6.	Inversión en Tiempo	50
4.4.7.	Escalamiento en tiempo	50
4.4.8.	Convolución	51
4.4.9.	Multiplicación	52
4.4.10.	Diferenciación en Tiempo	53
4.4.11.	Diferenciación en Frecuencia	53
4.4.12.	Propiedades de Simetría de la transformada de Fourier.	54
4.4.13.	Relación de Parseval.	55
4.4.14.	Resumen de Propiedades	56
4.5.	Transformada de Fourier de algunas funciones interesantes	57
4.5.1.	Exponencial Compleja	57
4.5.2.	Función seno	58
4.5.3.	Función coseno	58
4.5.4.	Función impulso unitario	58
4.5.5.	Constante	59
4.5.6.	Escalón Unitario	59
4.5.7.	Caja	60
4.5.8.	Ejemplos	61
4.6.	Transformada de Discreta de Fourier en Dos dimensiones.	64

4.6.1. Ejemplo	66
4.7. Implementación de la Transformada Discreta de Fourier en 2D	67
4.7.1. Ejemplo	69
4.7.2. Ejemplo	70
4.8. Transformada Rapida de Fourier	71
4.8.1. Ejemplo	72
4.8.2. Complejidad	73
4.8.3. Implementación	74
4.9. Convolución utilizando TF.	75
4.10. Teorema del Muestreo para señales discretas	75
4.10.1. Teorema del muestreo de Nyquist	78
4.10.2. Transformada Discreta de Fourier de un tren de Pulso	79
4.11. Teorema del Muestreo señales continuas	81
4.11.1. Transformada de Fourier de una señal periódica	81
4.11.2. Transformada de Furier de un Tren de Pulsos	83
4.11.3. Señal Muestreada en el Tiempo	85
4.11.4. Integración de la señal continua.	86
4.11.5. Ejemplos	87
Transformadas	91
5.1. Transformada Coseno a partir de la Transformada de Fourier	91
5.1.1. Transformada Inversa	93
5.1.2. Ejemplo	94
5.2. Definición de la DCT	94
5.2.1. Transformadas Coseno Utilizadas	96
Filtros.	97
6.1. Filtros Pasa bajas.	97
6.2. Filtros Pasa Altas.	99
6.3. Filtro pasa bajas Butterworth.	100
6.4. Filtros de pasa banda.	101
6.4.1. Ejemplo	102
6.5. Filtro de Gabor	103
6.5.1. Filtro de Gabor en dos dimensiones	103
6.6. Filtro de Wiener	107
6.7. Como entonar un filtro	109
6.7.1. Ejemplo	109
6.8. Filtro Elimina Banda	110
6.9. Filtro de Mediana	110
6.10. Filtro Binario	111
6.11. Filtro de Membrana	112

6.11.1. Probabilidad de un evento	113
6.11.2. Probabilidad condicional	113
6.11.3. Independencia	114
6.11.4. Regla de Bayes	114
6.11.5. Estimador de maxima verosimilitud.	115
6.11.6. Interpretación del filtro de Membrana en el dominio de la Frecuencia. 116	
Filtro de Kalman	121
7.1. Introducción	121
7.1.1. Ejemplo 1. Filtrado de una señal uni-dimensional	125
7.1.2. Ejemplo 2. Seguimiento de un objeto con aceleración constante . . .	127
7.1.3. Ejemplo 3. Determinación de parámetros.	132
Aplicaciones	137
8.1. Detección de corrimientos utilizando técnicas de restauración	137
8.1.1. Ejemplos	139
8.2. Filtrado de una señal de audio	141
8.2.1. Filtro Pasa Banda	141
8.2.2. Localización de Frecuencias Reales en el Dominio Discreto de Fourier 143	
8.2.3. Ejercicios	145
8.2.4. Código Fuente de los ejercicios	146
Algoritmo EM	151
9.1. Estimación de máxima verosimilitud.	151
9.2. El Algoritmo EM.	152
9.3. Aplicación del EM a mezclas de distribuciones.	153
9.3.1. Mezclas finitas de distribuciones normales.	155
9.4. Imágenes con tonos de gris constantes a pedazos.	156
Tareas	159
10.1. Tarea	159
10.2. Tarea	159

Antecedentes

1.1. Definiciones.

Representaremos a una señal discreta por $x(t)$ la cual puede ser función de una o varias variables. Esta señal discreta resulta de hacer la multiplicación de una señal continua por un tren de pulsos representados por

$$x(t) = f(t) \delta(t - kT)$$

donde $f(t)$ es una función continua y $\delta(t - kT)$ es un tren de pulsos unitarios con periodo T . La función $\delta(t)$ es la delta de Dirac la cual toma el valor unitario cuando su argumento es cero (ver figura 1.1).

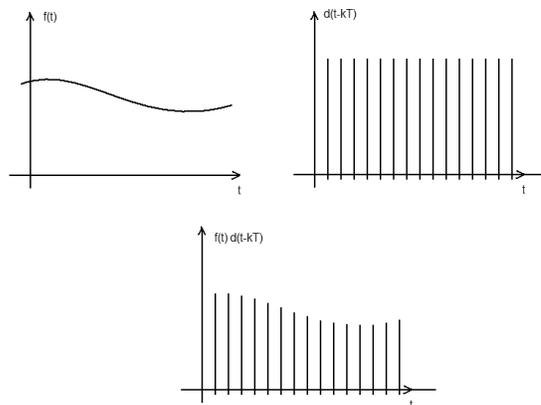


Figura 1.1: Multiplicación de una función por un tren de pulsos

1.2. Señales pares e impares

Podemos decir que una señal es par si es idéntica a su reflexión alrededor del origen, esto es

$$x(-t) = x(t)$$

y que es impar si

$$x(-t) = -x(t)$$

Una característica importante de cualquier señal, es que esta puede ser representada por la suma de una señal par y una señal impar

$$x(t) = \mathcal{P}(x(t)) + \mathcal{I}(x(t))$$

donde la parte par la calculamos

$$\mathcal{P}(x(t)) = \frac{1}{2} [x(t) + x(-t)]$$

y la impar por

$$\mathcal{I}(x(t)) = \frac{1}{2} [x(t) - x(-t)]$$

ver [[Oppenheim, 1997](#)]

1.2.1. Ejemplos.

(a) Demostrar que si $x(n)$ es discreta impar entonces

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

Demostración: si, $x(n)$ es impar, entonces

$$x(n) = -x(-n)$$

es decir

$$x(n) + x(-n) = 0$$

considerando

$$\sum_{-\infty}^{\infty} x(n) = \sum_{n=-\infty}^{-1} x(n) + x(0) + \sum_{n=1}^{\infty} x(n)$$

reordenando

$$\begin{aligned} \sum_{-\infty}^{\infty} x(n) &= x(0) + \sum_{n=-\infty}^{-1} x(n) + \sum_{n=1}^{\infty} x(n) \\ &= x(0) + \sum_{n=1}^{\infty} x(-n) + \sum_{n=1}^{\infty} x(n) \\ &= x(0) - \sum_{n=1}^{\infty} x(n) + x(-n) \end{aligned}$$

con $x(0) = 0$ y sustituyendo la definición de señal par en la ecuación anterior tenemos :

$$\begin{aligned} \sum_{-\infty}^{\infty} x(n) &= 0 + \sum_{n=1}^{\infty} 0 \\ \sum_{-\infty}^{\infty} x(n) &= 0 \end{aligned}$$

(b) Comprobar, si $x_1(n)$ es impar y $x_2(n)$ es par, entonces

$$x_1(n) \cdot x_2(n) \implies \text{impar}$$

Comprobación: Hacemos que

$$z(n) = x_1(n) \cdot x_2(n)$$

y

$$z(-n) = x_1(-n) \cdot x_2(-n)$$

Además, sabemos que

$$x_1(n) = -x_1(-n)$$

y

$$x_2(n) = x_2(-n)$$

sustituyendo las ecuaciones tenemos

$$z(n) = -x_1(-n) \cdot x_2(-n)$$

resulta

$$z(n) = -z(-n)$$

Lo cual implica que $z(n)$ es impar, i.e. $x_1(n) \cdot x_2(n)$ es impar.

(c) Considere que $x(n)$ es una señal con parte par $\mathcal{P}(x(n))$ y parte impar $\mathcal{I}(x(n))$, demostrar que

$$\sum_{n=-\infty}^{\infty} x^2(n) = \sum_{n=-\infty}^{\infty} P^2(x(n)) + \sum_{n=-\infty}^{\infty} I^2(x(n))$$

Demostración: Hacemos

$$\begin{aligned} \mathcal{P}(x(n)) &= x_p(n) \\ \mathcal{I}(x(n)) &= x_i(n) \end{aligned}$$

y

$$\begin{aligned} \sum_{n=-\infty}^{\infty} x^2(n) &= \sum_{n=-\infty}^{\infty} [x_p(n) + x_i(n)]^2 \\ &= \sum_{n=-\infty}^{\infty} (x_p^2(n) + 2x_p(n)x_i(n) + x_i^2(n)) \end{aligned}$$

sustituyendo el término $x_p(n)x_i(n)$ por $z(n)$ y reordenando

$$\sum_{n=-\infty}^{\infty} x^2(n) = \sum_{n=-\infty}^{\infty} (x_p^2(n) + x_i^2(n)) + 2 \sum_{n=-\infty}^{\infty} z(n)$$

Como $z(n)$ es el producto de una señal par y una impar, usamos el resultado del inciso b) y, concluimos que $z(n)$ es impar. Ahora, usando el resultado del inciso a) sabemos que

$$2 \sum_{n=-\infty}^{\infty} z(n) = 0$$

Si sustituimos este resultado en la ecuación tenemos que

$$\begin{aligned}\sum_{n=-\infty}^{\infty} x^2(n) &= \sum_{n=-\infty}^{\infty} (x_p^2(n) + x_i^2(n)) \\ \sum_{n=-\infty}^{\infty} x^2(n) &= \sum_{n=-\infty}^{\infty} x_p^2(n) + \sum_{n=-\infty}^{\infty} x_i^2(n)\end{aligned}$$

Una señal es periódica si tiene la propiedad de tener un valor positivo T para el cual

$$x(t) = x(t + T) \quad \forall t$$

en este caso diremos que la señal es periódica con periodo T

1.3. Señales básicas de tiempo continuo.

1.3.1. Función impulso unitario

La función impulso unitario se define como

$$\delta(t) = \begin{cases} 0, & t < 0 \\ 1, & t = 0 \end{cases}$$

En la figura 1.2 se muestra la función $\delta(t - 2)$

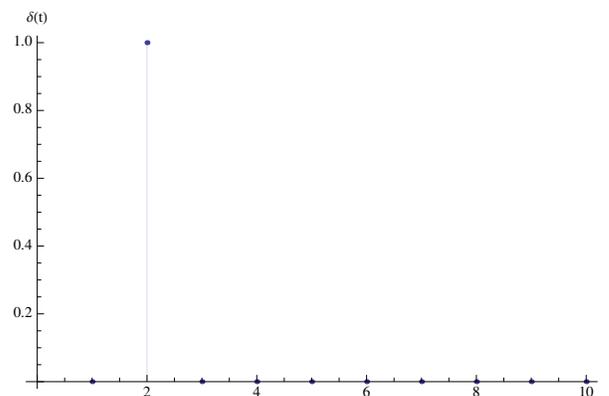


Figura 1.2: Función impulso unitario

1.3.2. Función escalón unitario

Otra señal de interés es la función escalón unitario la cual esta dada por

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t > 0 \end{cases}$$

En la figura 1.3 se muestra la función $u(t - 3)$

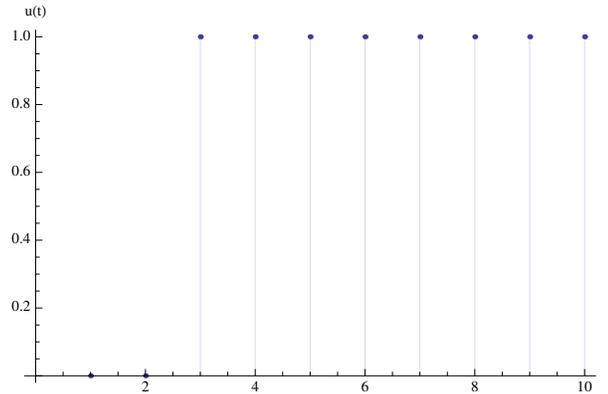


Figura 1.3: Función escalon unitario

Otra señal bastante útil es la función impulso unitario de tiempo continuo la cual está relacionada con el escalón unitario por la ecuación

$$u(t) = \int_{-\infty}^t \delta(\tau) d\tau$$

1.3.3. Exponencial compleja y senoidal.

La señal exponencial de tiempo continuo es de la forma

$$x(t) = Ce^{at}$$

donde C y a son, en general números complejos. Dependiendo de los valores de estos parámetros, la exponencial puede adoptar varias características diferentes. Una clase de exponenciales complejas que nos interesa es

$$x(t) = Ce^{j\omega_0 t}$$

donde $j = \sqrt{-1}$ es la base de los números imaginarios, $w_0 = 2\pi f_0$ es definida como la velocidad angular en radianes por segundo y f_0 es la frecuencia en ciclos por segundo o Hertz. Utilizando la relación de Euler esta señal puede expresarse como

$$x(t) = C \cos(w_0 t) + jC \text{sen}(w_0 t)$$

En la figura 1.4 se muestra una exponencial compleja con frecuencia angular $w = 1$

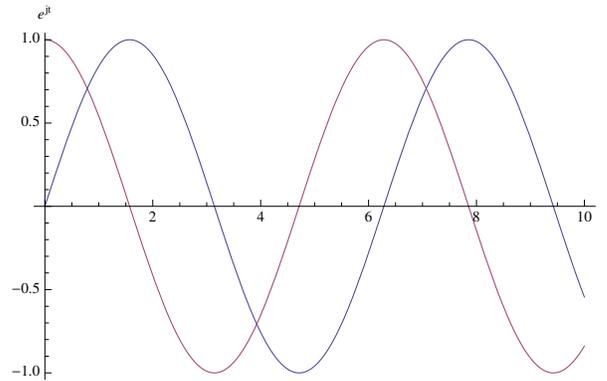


Figura 1.4: Exponencial Compleja

Una propiedad importante de esta señal, es su periodicidad. Una señal es periódica si cada intervalo de tiempo T , tenemos un repetición de la señal. Para verificar esta propiedad hacemos

$$\begin{aligned} e^{jw_0 t} &= e^{jw_0(t+T)} \\ e^{jw_0 t} &= e^{jw_0 t} e^{jw_0 T} \end{aligned}$$

para que esta ecuación se cumpla debemos tener que

$$e^{jw_0 T} = 1$$

Existen dos posibilidades para que esta condición se cumpla: la primera cuando $w_0 = 0$, la cual es periódica para cualquier valor de T , pero si $w_0 \neq 0$, entonces tenemos que existe un valor T_0 al cual llamamos periodo fundamental y esta dado por la ecuación

$$T_0 = \frac{2\pi}{w_0}$$

1.4. Señales Periódicas

Cualquier señal $x(t)$ que sea igual a la suma de dos señales periódicas, $x_1(t)$ y $x_2(t)$, con periodos fundamentales T_1 y T_2 respectivamente, será periódica si se cumple la siguiente relación:

$$\frac{T_1}{T_2} = \frac{m}{n}$$

y el periodo se calcula como $T = nT_1 = mT_2$ para n y m enteros, en caso contrario se considera que la señal no es periódica.

1.4.1. Ejemplos

Calcular el periodo de la señal en el caso de que esta es periódica

- a) $x(t) = \cos(25\pi t)$
- b) $x(t) = \sin(5t) + \cos(6t)$
- c) $x(t) = \sin(20t) - \sin(4t)$
- d) $x(t) = \sin^2(2\pi t)$

Inciso a)

Dado que tenemos una señal simple

$$\begin{aligned} x(t) &= x(t + T) \\ \cos(25\pi t) &= \cos(25\pi(t + T)) \\ \cos(25\pi t) &= \cos(25\pi t)\cos(25\pi T) - \sin(25\pi t)\sin(25\pi T) \end{aligned}$$

por lo tanto $\cos(25\pi T) = 1$ y el valor que hace esto es $25\pi T = 2\pi$ por lo tanto $T = \frac{2}{25}$

Otra manera de resolver el problema es haciendo $w = 25\pi$ y de la definición del periodo tenemos que

$$\begin{aligned} T &= \frac{2\pi}{w} \\ T &= \frac{2\pi}{25\pi} = \frac{2}{25} \end{aligned}$$

Inciso b)

Los periodos fundamentales de cada una de las señales es $T_1 = \frac{2\pi}{5}$ y $T_2 = \frac{2\pi}{6}$, de acuerdo con la formula tenemos

$$\frac{T_1}{T_2} = \frac{m}{n} = \frac{\frac{2\pi}{5}}{\frac{2\pi}{6}} = \frac{6}{5}$$

Podemos notar que tenemos la razón de dos números enteros por lo tanto la señal es periódica y tendrá periodo $T = nT_1 = mT_2$ lo cual nos da

$$T = nT_1 = 5 \frac{2\pi}{5} = 2\pi$$

Inciso c)

Los periodos fundamentales de cada una de las señales es $T_1 = \frac{2\pi}{20}$ y $T_2 = \frac{2\pi}{4}$, de acuerdo con la formula tenemos

$$\frac{T_1}{T_2} = \frac{m}{n} = \frac{\frac{2\pi}{20}}{\frac{2\pi}{4}} = \frac{4}{20} = \frac{1}{5}$$

Podemos notar que tenemos la razón de dos números enteros por lo tanto la señal es periódica y tendrá periodo $T = nT_1 = mT_2$ lo cual nos da

$$T = nT_1 = 5 \frac{2\pi}{20} = \frac{\pi}{2}$$

$$T = mT_2 = 1 \frac{2\pi}{4} = \frac{\pi}{2}$$

Inciso d)

Aplicado la identidad $x(t) = \text{sen}^2(2\pi t) = \frac{1 - \cos(4\pi t)}{2}$ tenemos que la frecuencia fundamental es $w = 4\pi$ por lo tanto el periodo es $T = \frac{2\pi}{4\pi} = 0.5$

Sistemas

2.1. Introducción

Un sistema se puede ver como cualquier proceso que produce una transformación de señales. Un sistema tiene una señal de entrada y una señal de salida la cual está relacionada con la entrada a través de la transformación del sistema.

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

donde $x(t)$ es la señal de entrada, $y(t)$ es la señal de salida y $T[\]$ es la transformación del sistema.

Entre los sistemas podemos tener interconexiones serie y paralelo dadas como los mostrados en la figura 2.5.

Sistemas con y sin memoria.

Si la salida de un sistema para cada valor de la variable independiente depende solo de la entrada en ese mismo instante de tiempo se dice que el sistema no tiene memoria.

$$y(t) = Rx(t)$$

Un ejemplo de un sistema con memoria es

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

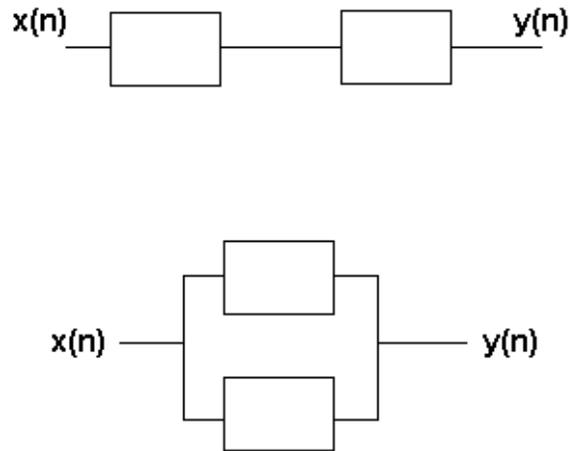


Figura 2.5: Sistemas serie y paralelo

2.1.1. Sistemas inversos.

Decimos que un sistema es invertible si dada una transformación T podemos encontrar la transformación T^{-1} tal que:

$$\begin{aligned} y(t) &= T[x(t)] \\ x(t) &= T^{-1}[y(t)] \end{aligned}$$

Un ejemplo de un sistema que no es invertible es $y(t) = 0$.

2.1.2. Sistemas causales.

Un sistema es causal si su salida en cualquier instante de tiempo depende sólo de los valores en el tiempo presente y en el pasado. Estos sistemas también son llamados no anticipativo.

$$y(t) = x(t-1)$$

$$y(t) = \int_{-\infty}^t x(\tau) d\tau$$

2.1.3. Estabilidad.

Un sistema es llamado estable si se produce una salida acotada para una entrada acotada.

Una señal es acotada si

$$|x(k)| \leq M < \infty$$

y la señal de salida $y(k) = T[x(k)]$, es acotada si

$$|y(k)| \leq N < \infty$$

Ejemplo. Consideremos la secuencia de los números de Fibonacci donde $x(k) = \{1, 2, 3, 5, 8, 13, \dots\}$ ó $x(k) = x(k-1) + x(k-2)$. En este caso tenemos un sistema no acotado ya que la sucesión crece indefinidamente para valores de k .

Un ejemplo de señal estable lo tenemos en la sucesión

$$x(n) = \sum_{k=0}^n \frac{1}{n^2}$$

para esta sucesión podemos ver que

$$x(n) - x(n-1) = \frac{1}{n^2}$$

si tomamos el límite podemos verificar que la sucesión converge en un valor estable

$$\lim_{n \rightarrow \infty} [x(n) - x(n-1)] = \lim_{n \rightarrow \infty} \frac{1}{n^2} = 0$$

2.2. Invariancia en el tiempo.

Para que un sistema sea invariante en el tiempo se debe cumplir que para un desplazamiento en la señal de entrada se produzca el mismo desplazamiento en la señal de salida.

$$T[D^n[x(t)]] = D^n[T[x(t)]]$$

Ejemplo. Considere la señal $y(t) = \text{sen}[x(t)]$

$$\begin{aligned} D^k [x(t)] &= x(t - k) \\ &= \text{sen}[x(t - k)] \\ D^k y(t) &= \text{sen}[x(t - k)] \end{aligned}$$

2.2.1. Linealidad

La característica principal de los sistemas lineales es

$$T[\alpha x(k) + \beta y(k)] = T[\alpha x(k)] + T[\beta y(k)]$$

esta propiedad es conocida como el principio de superposición.

Ejemplo. Considere el sistema $y(k) = mx(k) + b$. Que valores debe tener el sistema para ser lineal.

Consideremos dos señales $x_1(t)$ y $x_2(t)$

$$\begin{aligned} y_1(k) &= mx_1(k) + b \\ y_2(k) &= mx_2(k) + b \\ y_1(k) + y_2(k) &= m(x_1(k) + x_2(k)) + 2b \end{aligned}$$

Si aplicamos la transformación a la suma de $x_1(t)$ y $x_2(t)$ tendremos

$$y_3(k) = m(x_1(k) + x_2(k)) + b$$

note que $y_3(k)$ es diferente de $y_1(k) + y_2(k)$, la única posibilidad es que la constante b sea igual a cero.

2.3. Ejemplos

2.3.1. Ejemplo

Demuestre que $y(n) = x(n) - x(n - 1)$ es invariante en el tiempo.

Solución. Sea, $v(n) = D^{n_0} \{T[x(n)]\}$ y $\hat{v}(n) = T \{D^{n_0}[x(n)]\}$. Un sistema es invariante si $v(n) = \hat{v}(n)$

Para nuestro ejemplo $v(n) = D^{n_0}[x(n) - x(n-1)] = x(n - n_0) - x(n - n_0 - 1)$ por otro lado $\hat{v}(n) = T[x(n - n_0)] = x(n - n_0) - x(n - n_0 - 1)$, como $v(n) = \hat{v}(n)$ el sistema es invariante en el tiempo.

Nota: En este ejemplo, hacemos una aproximación de la derivada utilizando diferencias finitas $y(n) \simeq \frac{dx(n)}{dn} = x(n) - x(n-1)$.

2.3.2. Ejemplo

Demuestre que $y(n) = \sum_{k=k_0}^n x(k)$ es invariante en el tiempo.

Solución. En éste caso, $v(t) = D^{n_0} \left[\sum_{k=k_0}^{k_f} x(k) \right] = \sum_{k=k_0}^{k_f} x(k-n_0)$ y $\hat{v}(k) = T[D^{n_0} [x(k)]] = \sum_{k=k_0}^{k_f} x(k - n_0)$. Como $v(k) = \hat{v}(k)$ el sistema es invariante en el tiempo.

2.3.3. Ejemplo

Dada la sucesión

$$y(n) = \frac{1}{4}y(n-1) + x(n) \quad (2.1)$$

1. Probar que es un sistema Lineal invariante en el Tiempo (LIT) y
2. Determinar la salida $y(n)$ si $x(n) = \delta(n-1)$

Comenzaremos por dar una solución de la recurrencia, haciendo sustituciones sucesivas

$$\begin{aligned} y(n) &= x(n) + \frac{1}{4}y(n-1) \\ &= x(n) + \frac{x(n-1)}{4} + \frac{1}{4}y(n-2) \\ &= x(n) + \frac{x(n-1)}{4} + \frac{x(n-2)}{4^2} + \frac{1}{4^2}y(n-3) \\ &= x(n) + \frac{x(n-1)}{4} + \frac{x(n-2)}{4^2} + \frac{x(n-3)}{4^3} + \dots + \frac{x(1)}{4^{n-1}} \end{aligned}$$

donde de manera general la solución de la recursión es

$$y(n) = \sum_{k=0}^{n-1} \frac{x(n-k)}{4^k} \quad (2.2)$$

Linealidad

Por otra parte tenemos que verificar que el sistema dado por 2.1 es Lineal, primero calculamos utilizando 2.2

$$T[\alpha x_1(n)] + T[\beta x_2(n)] = \alpha \sum_{k=0}^{n-1} \frac{x_1(n-k)}{4^k} + \beta \sum_{k=0}^{n-1} \frac{x_2(n-k)}{4^k}$$

y segundo utilizando 2.2 calculamos

$$\begin{aligned} T[\alpha x_1(n) + \beta x_2(n)] &= \sum_{k=0}^{n-1} \frac{\alpha x_1(n-k) + \beta x_2(n-k)}{4^k} \\ &= \alpha \sum_{k=0}^{n-1} \frac{x_1(n-k)}{4^k} + \beta \sum_{k=0}^{n-1} \frac{x_2(n-k)}{4^k} \end{aligned}$$

lo cual demuestra que el sistema es lineal.

Invariancia en el tiempo

En este caso tenemos que mostrar $D^{n_0} [T[x(n)]] = T[D^{n_0}[x(n)]]$, aplicando la solución de la recursión dada por 2.2, tenemos

$$D^{n_0} [T[x(n)]] = D^{n_0} \left[\sum_{k=0}^{n-1} \frac{x(n-k)}{4^k} \right] = \sum_{k=0}^{n-1} \frac{x(n-n_0-k)}{4^k}$$

Si aplicamos la traslación a la señal x tenemos

$$D^{n_0} [x(n)] = x(n-n_0), x(n-1-n_0), x(n-2-n_0), \dots, x(-n_0),$$

y posteriormente calculamos

$$T[D^{n_0}[x(n)]] = \sum_{k=0}^{n-1} \frac{x(n-n_0-k)}{4^k}$$

De lo anterior podemos concluir que el sistema es invariante en el tiempo.

Respuesta al impulso

Para determinar la salida $y(n) = \frac{1}{4}y(n-1) + x(n)$ y suponiendo que $y(0) = 0$, se obtiene la siguiente tabla al evaluar recursivamente:

n	$x(n)$	$y(n)$
0	0	0
1	1	1
2	0	$\frac{1}{4}$
3	0	$\frac{1}{4^2}$

Se observa que para cualquier n la salida será $y(n) = \left(\frac{1}{4}\right)^{n-1}$, note que esta solución puede verificarse utilizando la solución de la recursión.

Convolución

3.1. Correlación

Consideremos dos señales x y y , donde $x = [y_i, y_{i+1}, \dots, y_{i+N}]$ es un subconjunto de los valores de la señal $y = [y_0, y_1, y_2, \dots, y_M]$. ¿En que punto la señal x empatará mejor con la señal y ? Comenzaremos por utilizar como medida de similitud el producto escalar de vectores, así el ángulo entre estas dos señales lo podemos calcular como

$$\cos(\theta) = \frac{x^T \hat{y}}{|x| |\hat{y}|}$$

donde $\hat{y} = [y_j, y_{j+1}, \dots, y_{j+N}]$ y θ es el ángulo entre los dos vectores. Note que cuando los dos vectores son iguales tendremos que el ángulo es cero, el valor máximo será dado por $|x|^2$ y que el producto $x^T \hat{y}$ tenderá a este máximo.

El producto de $x^T \hat{y}$ lo podemos calcular como

$$r_{yx} = \sum_{k=0}^N x(k) \hat{y}(k)$$

dado que \hat{y} es un subconjunto de y , si aplicamos un desplazamiento podemos calcular el producto $x^T \hat{y}_n$ en las n posiciones posibles de la serie con la siguiente expresión

$$r_{xy}(n) = \sum_{k=0}^N x(k) y(k+n)$$

La cross-correlación de las señales $x(n)$ y $y(n)$ es la secuencia $r_{xy}(n)$ definida por

$$r_{xy}(n) = \sum_{k=-\infty}^{\infty} x(k) y(k+n)$$
$$r_{xy}(t) = \int_{-\infty}^{\infty} x(\tau) y(\tau+t) d\tau$$

La función de correlación suministra una medida de la similitud o inter-dependencia entre las funciones x y y .

3.1.1. Propiedades

Conmutación

Demostrar que la función de correlación y auto-correlación no son conmutativas, es decir

$$\begin{aligned} r_{xy}(n) &= r_{yx}(-n) \\ r_{xx}(n) &= r_{xx}(-n) \end{aligned}$$

para ello hacemos

$$r_{xy}(n) = \sum_{k=-\infty}^{\infty} x(k) y(k+n)$$

y sustituimos $\hat{k} = k + n$

$$r_{yx}(-n) = \sum_{\hat{k}=-\infty}^{\infty} x(\hat{k}-n) y(\hat{k})$$

en el caso de la auto-correlación tenemos

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

y sustituimos $\hat{k} = k + t$

$$r_{xx}(-n) = \sum_{\hat{k}=-\infty}^{\infty} x(\hat{k}-n) x(\hat{k})$$

Lo cual significa que la correlación no es conmutativa.

Invarianza a traslación

Es la correlación invariante a traslación. Comenzamos por aplicar el operador de traslación a la señal $D^a[x(n)] = x(n - a)$ y calculamos la correlación

$$r_{xy}(n) = \sum_{k=-\infty}^{\infty} x(n - a) y(k + n)$$

si sustituimos

$$\begin{aligned}\hat{n} &= n - a \\ n &= \hat{n} + a\end{aligned}$$

$$\sum_{k=-\infty}^{\infty} x(\hat{n}) y(\hat{k} + (n + a)) = r_{xy}(n + a)$$

Lo cual es equivalente a $D^{-a}[r_{xy}(n)]$, por lo tanto la correlación no es invariante a traslación.

Linealidad

Pero, ¿será lineal la correlación?

$$\begin{aligned}r_{xy}(n) &= \sum_{k=-\infty}^{\infty} [\alpha x_1(k) + \beta x_2(k)] y(k + n) \\ &= \alpha \sum_{k=-\infty}^{\infty} x_1(k) y(k + n) + \beta \sum_{k=-\infty}^{\infty} x_2(k) y(k + n) \\ &= \alpha r_{x_1 y}(n) + \beta r_{x_2 y}(n)\end{aligned}$$

si es una transformación lineal la correlación.

3.1.2. Ejemplo

Suponga una señal $x(t) = [0, 1, 2, 3, 4, 0, 1, 2, 3, 4, \dots]$ (Fig. 4.16(a)) y una señal $y(t) = [0, 1, 2, 3, 4]$ (Fig. 3.6(b)). La correlación entre ambas señales, se muestra en la Fig. 4.16(b). Note que los máximos se obtienen cuando la señal y empata perfectamente con x .

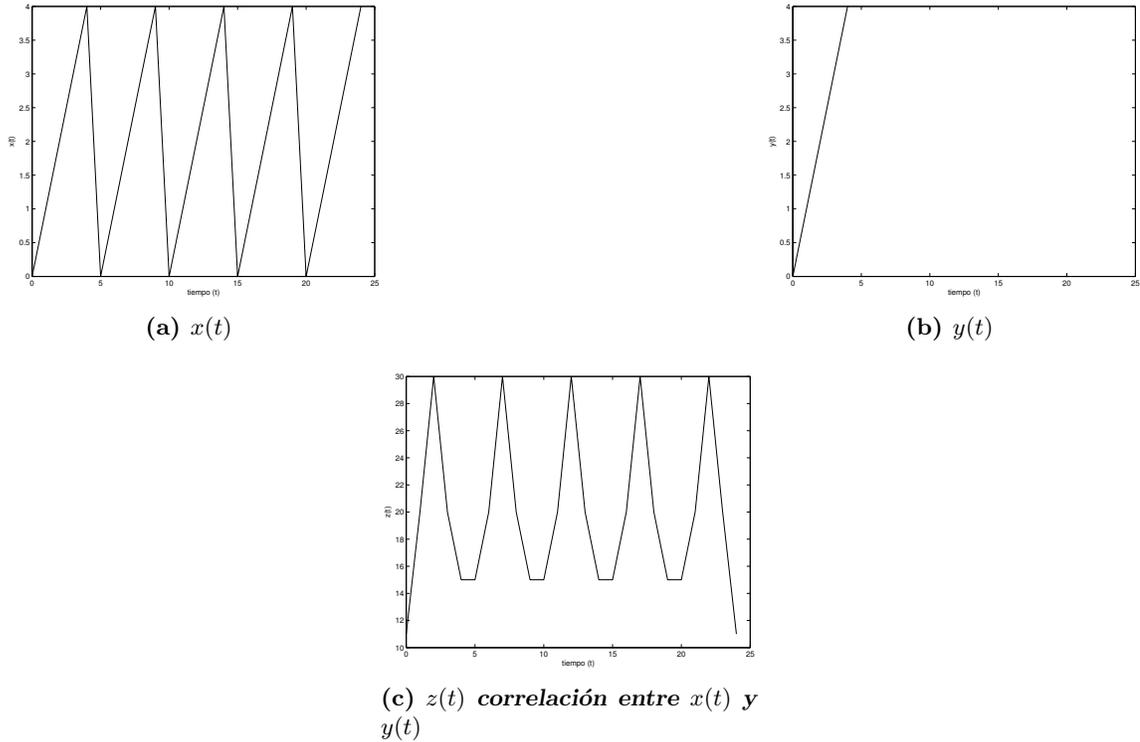


Figura 3.6: Correlación

3.1.3. Correlación en dos dimensiones

Si la señal está en dos dimensiones, como es el caso de imágenes la correlación se representa como

$$r_{xy}(n, m) = \sum_{k=-n_k}^{n_k} \sum_{l=-n_l}^{n_l} x(k, l) * y(k + n, l + m)$$

3.2. Representación de señales en términos de impulsos.

La función impulso unitario, puede utilizarse para construir una clase amplia de señales. Para ilustrar como funciona consideremos que tenemos una señal discreta dada por $x(n)$. Si queremos ver una parte de la señal en cierto instante de tiempo multiplicamos la señal de entrada por un impulso en el instante que deseamos analizar, así por ejemplo:

$$\begin{aligned}
 x(-1)\delta(n+1) &= \begin{cases} x(-1) & \text{si } n=-1 \\ 0 & \text{si no} \end{cases} \\
 x(0)\delta(n+0) &= \begin{cases} x(0) & \text{si } n=0 \\ 0 & \text{si no} \end{cases} \\
 x(1)\delta(n-1) &= \begin{cases} x(1) & \text{si } n=1 \\ 0 & \text{si no} \end{cases}
 \end{aligned}$$

por lo tanto la suma suma de estos términos me da la señal x . Matemáticamente podemos expresar esta suma como

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

esta ecuación es llamada la propiedad de escudriñamiento del impulso unitario.

3.3. Convolución

Podemos extender el concepto sustituyendo la función impulso por cualquier otra función, así obtenemos la expresión de la convolución.

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

note que también es un sistema LIT. La convolución la podemos representar por $y = x * h$

3.3.1. Propiedades

Conmutación

La convolución es conmutativa es decir

$$f_1 * f_2 = f_2 * f_1$$

demostración, para probar la propiedad conmutativa hacemos

$$f_1 * f_2 = \sum_{k=-\infty}^{\infty} f_1(k)f_2(n-k)$$

haciendo el cambio de variable $\widehat{k} = n - k$ tenemos

$$\sum_{\widehat{k}=-\infty}^{\infty} f_1(n - \widehat{k}) f_2(\widehat{k}) = f_2 * f_1$$

Asociativa

La propiedad asociativa consiste en

$$(f_1 * f_2) * f_3 = f_1 * (f_2 * f_3)$$

para demostrar la propiedad asociativa hacemos $f_1 * f_2 = g$ y $f_2 * f_3 = h$

$$g * f_3 = f_1 * h$$

puesto que

$$g(n) = \sum_{k=-\infty}^{\infty} f_1(k) f_2(n - k)$$

se tiene que

$$\begin{aligned} g(n) * f_3(n) &= \sum_{j=-\infty}^{\infty} g(j) f_3(n - j) \\ &= \sum_{j=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} f_1(k) f_2(j - k) \right] f_3(n - j) \end{aligned}$$

sustituyendo $l = j - k$ y cambiando el orden de las sumatorias

$$\begin{aligned} &= \sum_{k=-\infty}^{\infty} f_1(k) \sum_{l=-\infty}^{\infty} f_2(l) f_3(n - (l + k)) \\ &= \sum_{k=-\infty}^{\infty} f_1(k) \sum_{l=-\infty}^{\infty} f_2(l) f_3((n - k) - l) \\ &= \sum_{k=-\infty}^{\infty} f_1(k) h(n - k) \\ &= f_1 * h \end{aligned}$$

Invarianza a traslación

Para probar que la convolución es un sistema invariante a traslación hacemos la convolución de la señal $x(k)$ trasladada un valor k_0

$$y(n) = \sum_{k=-\infty}^{\infty} x(k - k_0) h(n - k)$$

haciendo $\hat{k} = k - k_0$ tenemos:

$$\begin{aligned} \sum_{\hat{k}=-\infty}^{\infty} x(\hat{k}) h(n - (\hat{k} + k_0)) \\ \sum_{\hat{k}=-\infty}^{\infty} x(\hat{k}) h((n - k_0) - \hat{k}) \\ = y(n - k_0) \end{aligned}$$

Linealidad

Para probar que se trata de un sistema lineal, calculamos la convolución de la suma de dos señales x_1 y x_2

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} [\alpha x_1(k) + \beta x_2(k)] h(n - k) \\ &= \alpha \sum_{k=-\infty}^{\infty} x_1(k) h(n - k) + \beta \sum_{k=-\infty}^{\infty} x_2(k) h(n - k) \\ &= \alpha y_1(n) + \beta y_2(n) \end{aligned}$$

3.3.2. Sucesión útil

Una sucesión que nos será especialmente útil para realizar los cálculos de convolución y correlación es la sucesión geométrica

$$s_N = \sum_{n=0}^N a^n = 1 + a + a^2 + \dots + a^N$$

para esta serie podemos ver

$$\begin{aligned} s_N &= 1 + a(1 + a + a^2 + \dots + a^{N-1}) \\ &= 1 + a s_{N-1} \end{aligned}$$

que pasa si multiplicamos s_N por $(1 - a)$

$$\begin{aligned}(1 - a) s_N &= (1 + a + a^2 + \dots + a^N) - (a + a^2 + a^3 \dots + a^{N+1}) \\ (1 - a) s_N &= 1 - a^{N+1}\end{aligned}$$

de lo cual concluimos que

$$s_N = \frac{1 - a^{N+1}}{1 - a}$$

Esta serie será convergente en el caso de que $|a| < 1$ y divergente en el caso de que $|a| > 1$. Lo cual lo podemos verificar haciendo

$$\lim_{N \rightarrow \infty} a^N = 0$$

y el valor de convergencia lo podemos calcular con

$$s_N = 1 + a s_{N-1}$$

en el límite $s_N = s_{N-1} = r$

$$\begin{aligned}r &= 1 + ar \\ r(1 - a) &= 1 \\ r &= \frac{1}{1 - a}\end{aligned}$$

En el caso de que $|r| = 1$, la serie converge al valor de

$$\begin{aligned}s_N &= \sum_{n=0}^N 1^n = 1 + 1 + 1^2 + \dots + 1^N \\ &= N + 1\end{aligned}$$

y en el caso de $|r| > 1$ la serie diverge

$$\lim_{N \rightarrow \infty} a^N \neq 0$$

3.3.3. Ejemplos

Ejemplo 1

Considere una señal $x(n) = \alpha^n$ y un kernel dado por $h(n) = \beta^n$. Calcular la convolución de estas dos señales.

$$\begin{aligned}
 y(n) &= \sum_{k=0}^{N-1} \alpha^k \beta^{(n-k)} \\
 y(n) &= \sum_{k=0}^{N-1} \alpha^k \beta^n \beta^{-k} \\
 y(n) &= \beta^n \sum_{k=0}^{N-1} \left(\frac{\alpha}{\beta}\right)^k \\
 y(n) &= \beta^n \frac{1 - \left(\frac{\alpha}{\beta}\right)^N}{1 - \frac{\alpha}{\beta}}
 \end{aligned}$$

si quisiéramos calcular la correlación hacemos

$$\begin{aligned}
 y(n) &= \sum_{k=0}^{N-1} \alpha^k \beta^{(k-n)} \\
 y(n) &= \sum_{k=0}^{N-1} \alpha^k \beta^{-n} \beta^k \\
 y(n) &= \beta^{-n} \sum_{k=0}^{N-1} (\alpha\beta)^k \\
 y(n) &= \beta^{-n} \frac{1 - (\alpha\beta)^N}{1 - (\alpha\beta)}
 \end{aligned}$$

Ejemplo 2

Mostrar que la convolución de una señal impulso unitario con un kernel cualquiera es el mismo kernel

$$\delta(n) * h(n) = h(n)$$

prueba

Por definición tenemos

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

para nuestro caso sustituimos $x(n)$ por la función impulso $\delta(n)$ dando lugar a

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

recordemos que la función impulso, será igual a 1 solo cuando su argumento es cero, por lo cual

$$y(n) = \cdots + 0 \times h(n-1) + 1 \times h(n) + 0 \times h(n+1) + \cdots$$

$$y(n) = h(n)$$

Por esta propiedad al kernel se le conoce también como respuesta al impulso.

Ejemplo 3

Mostrar que dado $y(n) = x(n) * h(n)$ podemos hacer $y'(n) = x(n) * h'(n)$

Por definición tenemos

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

y que la derivada de una función discretizada puede ser representada por $y'(n) = y(n) - y(n-1)$ tenemos

$$y(n) - y(n-1) = \sum_{k=-\infty}^{\infty} x(k) * h(n-k) - \sum_{k=-\infty}^{\infty} x(k) * h((n-1)-k)$$

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

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

3.4. Convolución en dos dimensiones

Podemos definir la convolución en dos dimensiones para un kernel de tamaño $(2n_k + 1)^2$ como:

$$y(n, m) = \sum_{k=-n_k}^{n_k} \sum_{l=-n_k}^{n_k} h(k, l)x(n - k, m - l)$$

Sin embargo el número de operaciones para calcular la convolución en dos dimensiones para una imagen de tamaño $N \times M$ será $(N \times M) \times (2n_k + 1)^2$, en el caso de una imagen cuadrada y un kernel del tamaño de la imagen el orden de las operaciones será $O(N^4)$. Una forma de reducir el número de operaciones es buscando la separabilidad de los kernels.

3.4.1. Separabilidad

Definimos que un kernel es separable si este puede representarse como

$$h(n, m) = h_1(n) \times h_2(m)$$

Con esta propiedad podemos calcular nuestra convolución como:

$$y(n, m) = \sum_{k=-n_k}^{n_k} \sum_{l=-n_k}^{n_k} h_1(k)h_2(l)x(n - k, m - l)$$

Reorganizando las sumatorias tenemos:

$$y(n, m) = \sum_{k=-n_k}^{n_k} h_1(k) \sum_{l=-n_k}^{n_k} h_2(l)x(n - k, m - l)$$

$$y(n, m) = h_1 * (h_2 * g)(n, m)$$

De la ecuación anterior se desprende que la convolución en dos dimensiones con kernels separable, la podemos hacer como la convolución en una dimensión con un kernel h_1 y posteriormente en la otra dimensión con un kernel h_2 .

Por lo tanto el número de operaciones que tenemos que realizar es $N \times M$ para la primer convolución y $N \times M$ para la segundo dando lugar a que la complejidad algorítmica sea $O(N^2)$ muy por debajo de la convolución bidimensional.

Así que intentaremos hacer uso de kernels separable en lo posible con el propósito de reducir el número de operaciones.

3.5. Algunos kernels interesantes

3.5.1. Suavizadores

Caja

Para implementar este suavizador, en una dimensión, utilizamos una señal dada como $h(n) = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0]$

En dos dimensiones, podemos implementar este kernel de manera separada si hacemos la siguiente operación

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En la figura 3.7(a) y 3.7(d), se presentan gráficamente, los kernels para suavizadores de caja en una y dos dimensiones

Binomial

Este kernel está basado en los coeficientes binomiales. Una manera fácil de calcularlo es utilizar un triángulo de Pascal, de la siguiente forma.

$$\begin{array}{c} 1,1 \\ 1,2,1 \\ 1,3,3,1 \\ 1,4,6,4,1 \\ 1,5,10,10,5,1 \\ 1,6,15,20,15,6,1 \end{array}$$

Dado $b_1 = [1, 1]$ podemos calcular

$$\begin{aligned} b_2 &= b_1 * b_1 \\ b_3 &= b_2 * b_1 \\ b_4 &= b_3 * b_1 \\ &\vdots \\ b_{k+1} &= b_k * b_1 \end{aligned}$$

Para su implementación este kernel es normalizado para que su suma de 1. En dos dimensiones de manera separable se puede implementar como

$$B_3 = b_3 * b_3$$

Note que este kernel es recursivo.

$$\frac{1}{4} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \frac{1}{4} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \frac{1}{16} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Gaussiano

Este kernel es creado utilizando una campana de Gauss. Par calcularla utilizamos la expresión

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

donde: μ es la media de la distribución y σ es la varianza. En nuestro caso la media la consideraremos cero y la varianza la utilizaremos como una estimación del tamaño del kernel.

En dos dimensiones calcularemos nuestro kernel de manera desacoplada haciendo $g(x, y) = g(x) * g(y)$. En la figura 3.7(c) se muestra el kernel Gaussiano en una dimensión y 3.7(f) en dos dimensiones.

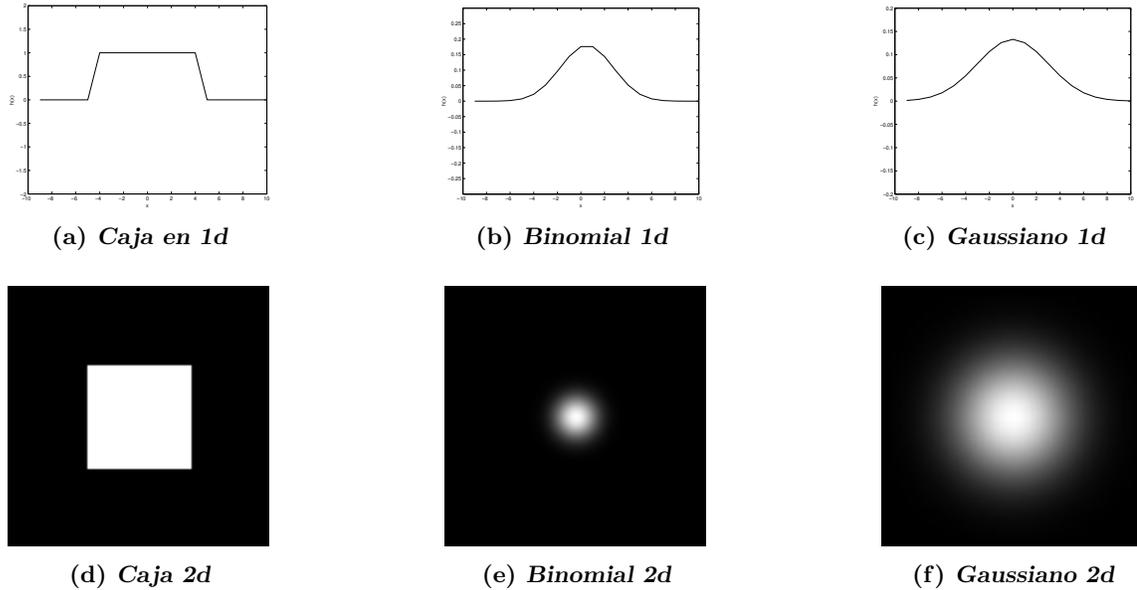


Figura 3.7: Kerneles para suavizado de señales en una y dos dimensiones

3.5.2. Derivadas

Derivada en x y y

La derivada de una función esta definida como

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x) - f(x - h)}{h}$$

considerando un incremento unitario, que para el caso discreto es el más pequeño y que la señal fue capturada con una frecuencia de muestreo adecuada tenemos que la derivada la podemos aproximar por

$$\frac{df(x)}{dx} \approx f(x) - f(x - h)$$

De la expresión anterior podemos ver que el kernel de derivadas será $d(n) = [1 - 1]$, así la derivada la calcularemos como $y(n) * d(n)$.

Para calcular las derivadas de orden N, solamente debemos recordar que se debe aplicar sucesivamente la derivada de orden uno, con ello damos lugar a la siguiente familia de kerneles.

+1,-1
 +1,-2,+1
 +1,-3,+3,-1
 +1,-4,+6,-4,-1

Derivadas de Gaussianas

Un problema que presenta el cálculo de las derivadas utilizando convolución es que son amplificadores de ruido. Esto quiere decir que si la señal presenta ruido, la señal resultante de la convolución, el ruido será más notorio. Una manera de eliminar el ruido de una señal es aplicar un suavizador, el más apropiado es el suavizado Gaussiano [Jahne, 2002]. Así pues, para eliminar el ruido, es deseable primero convolucionamos con un kernel gaussiano y luego aplicamos un kernel de derivadas de la siguiente forma

$$y(n) = (x(n) * h(n)) * d(n) = x(n) * (h(n) * d(n)) = x(n) * \frac{dg(x)}{dx}$$

Resulta que tanto $h(n)$ y $d(n)$ son aproximaciones de una Gaussiana y una derivada respectivamente, y que ambos dan lugar a un nuevo kernel, pero es más inteligente calcular este kernel como la derivada de una función gaussiana. El kernel de derivada Gaussiano es

$$G_x(x) = \frac{dg(x)}{dx} = \frac{-x}{\sigma^2} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{x^2}{2\sigma^2}\right\} \right]$$

$$g_x(x) = \frac{dg(x)}{dx} = \frac{-x}{\sigma^2} g(x)$$

La implementación en Java de este kernel es:

```
public static void GDerKernel(double gk[], double sigma)
{
    int i;
    int m = (int) (3 * sigma + 0.5);
    GKernel(gk, sigma);
    for (i = 0; i <= 2 * m; i++)
        gk[i] *= - (i - m) / (sigma * sigma);
}
```

En dos dimensiones la derivada de la gaussiana la podemos escribir como

$$D_x g(x, y) = -\frac{(x - \mu)}{2\pi\sigma^2\sigma^2} e^{\left(-\frac{(x-\mu)^2 + (y-\eta)^2}{2\sigma^2}\right)}$$

al aplicar la convolución tenemos

$$y(n, m) = \sum_{k=-n_k}^{n_k} \sum_{l=-n_k}^{n_k} \frac{-k}{2\pi\sigma^2\sigma^2} e^{\left(\frac{-k^2 - l^2}{2\sigma^2}\right)} x(n - k, m - l)$$

Reorganizando las sumas tenemos

$$y(n, m) = \sum_{k=-n_k}^{n_k} \frac{-k}{\sigma^2} \frac{1}{\sqrt{2\pi}\sigma} e^{\left(\frac{-k^2}{2\sigma^2}\right)} \sum_{l=-n_k}^{n_k} \frac{1}{\sqrt{2\pi}\sigma^2} e^{\left(\frac{-l^2}{2\sigma^2}\right)} x(n - k, m - l)$$

$$y(n, m) = \sum_{k=-n_k}^{n_k} g_x(k) \sum_{l=-n_k}^{n_k} g(l) x(n - k, m - l)$$

$$y(n, m) = g_x * g * x(n, m)$$

Lo cual es equivalente a convolucionar con un kernel gaussiano en la dirección de y y un kernel de derivadas gaussianas en la dirección de x . El código para esta implementación es

```
public static void DerivadaX(double dat[][], double res[][], double sigma, int nr, int
    int m;
    m = (int) (3.0 * sigma + 0.5);
    double kx[] = new double[2 * m + 1];
    double ky[] = new double[2 * m + 1];
    GDerKernel(kx, sigma);
    GKernel(ky, sigma);
    Conv2DC(dat, res, kx, ky, m, nr, nc);
}
```

En el caso de derivada en y procedemos igual que en el caso anterior

$$y(n, m) = \sum_{k=-n_k}^{n_k} \frac{1}{\sqrt{2\pi}\sigma^2} e^{\left(\frac{-k^2}{2\sigma^2}\right)} \sum_{l=-n_k}^{n_k} \frac{-l}{\sigma^2} \frac{1}{\sqrt{2\pi}\sigma} e^{\left(\frac{-l^2}{2\sigma^2}\right)} x(n - k, m - l)$$

$$y(n, m) = \sum_{k=-n_k}^{n_k} g(k) \sum_{l=-n_k}^{n_k} g_y(l) x(n-k, m-l)$$

$$y(n, m) = g * g_y * x(n, m)$$

Note que en ambos casos $g_x(n) = g_y(n)$

El código Java para esta implementación es:

```
public static void DerivadaY(double dat[][], double res[][], double sigma,
                             int nr, int nc) {
    int m;
    m = (int) (3.0 * sigma + 0.5);
    double kx[] = new double[2 * m + 1];
    double ky[] = new double[2 * m + 1];
    GDerKernel(ky, sigma);
    GKernal(kx, sigma);
    Conv2DC(dat, res, kx, ky, m, nr, nc);
}
```

En las figuras 3.8(a) y 3.8(b), se presentan los kernels de derivadas de Gaussianas en x y y para diferentes valores de desviación estándar σ .

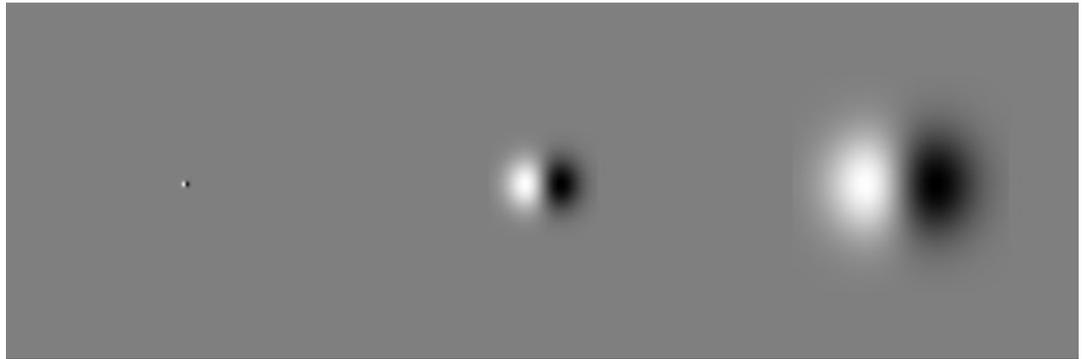
Derivadas de Gaussianas de segundo orden

El kernel de segunda derivada es

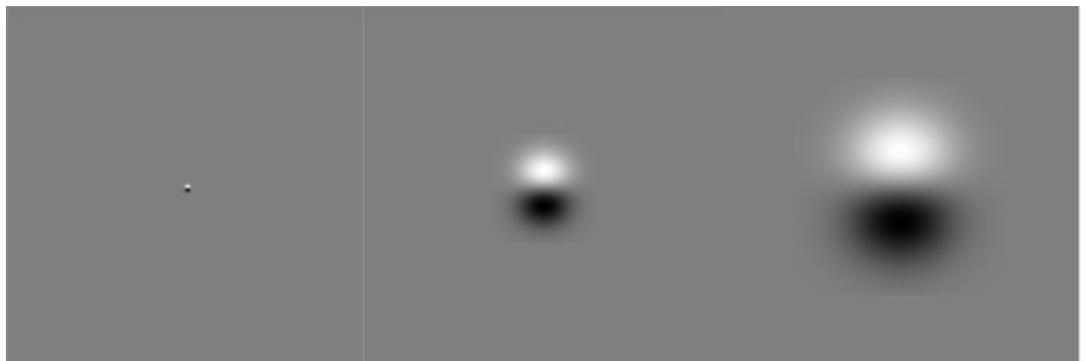
$$\frac{d^2 g(x)}{dx} = \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right] \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{x^2}{2\sigma^2}\right\} \right]$$

En dos dimensiones, la implementación de estos kernels lo haremos de manera separable de la siguiente manera. Así la derivada en la dirección x será

$$\begin{aligned} g(x, y) &= g(x) * g(y) \\ \frac{\partial g(x, y)}{\partial x} &= \frac{\partial g(x)}{\partial x} * g(y) \\ \frac{\partial g(x, y)}{\partial y} &= g(x) * \frac{\partial g(y)}{\partial y} \end{aligned}$$

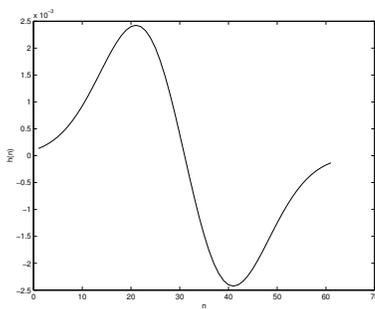


(a) Kernel de Derivada en $x \frac{\partial g(x,y)}{\partial x}$ en 2d

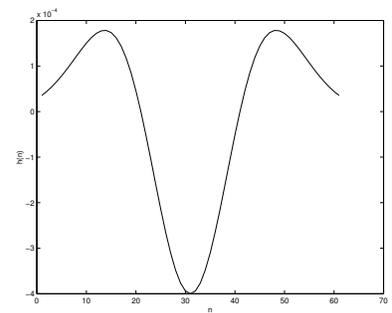


(b) Kernel de derivada en $y \frac{\partial g(x,y)}{\partial y}$ en 2d

Figura 3.8: Kerneles de derivadas de gaussianas con $\sigma = 1, 10$ y 20 en dos dimensiones



(a) Primer derivada 1d



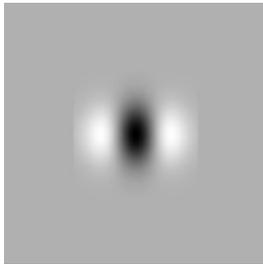
(b) Segunda derivada 1d

Figura 3.9: Kerneles de derivadas de gaussianas con $\sigma = 10$

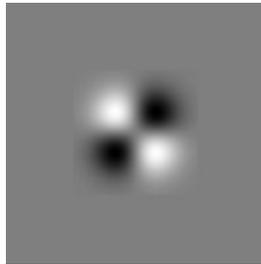
Las segundas derivadas se calcularán como:

$$\begin{aligned}\frac{\partial^2 g(x, y)}{\partial x^2} &= \frac{\partial^2 g(x)}{\partial x^2} * g(y) \\ \frac{\partial^2 g(x, y)}{\partial x \partial y} &= \frac{\partial g(x)}{\partial x} * \frac{\partial g(y)}{\partial y} \\ \frac{\partial^2 g(x, y)}{\partial y^2} &= g(x) * \frac{\partial^2 g(y)}{\partial y^2}\end{aligned}$$

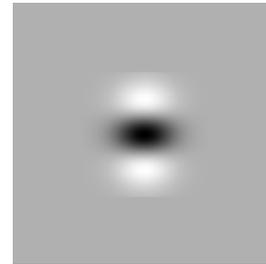
En la figuras 3.10(a), 3.10(b) y 3.10(c), se presentan los kerneles de derivadas de Gaussianas en dos dimensiones.



(a) Kernel $\frac{\partial^2 g(x, y)}{\partial x^2}$ en $2d$



(b) Kernel $\frac{\partial^2 g(x, y)}{\partial x \partial y}$ en $2d$



(c) Kernel $\frac{\partial^2 g(x, y)}{\partial y^2}$ en $2d$

Figura 3.10: Kerneles de derivadas de gaussianas con $\sigma = 10$ en dos dimensiones

Laplaciano

El Laplaciano resulta de sumar las segundas derivadas en x y y de una señal bidimensional y esta dado por la siguiente expresión

$$\Delta = \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2}$$

Este kernel lo podemos construir sumando los kerneles de segundas derivadas de la siguiente forma

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

De manera robusta lo podemos calcular utilizando derivadas de Gaussiana, lo cual da lugar al kernel denominado "Mexican Hat". Este kernel se utiliza para determinar bordes en una imagen y podemos ver este kernel en la figura [3.11](#)

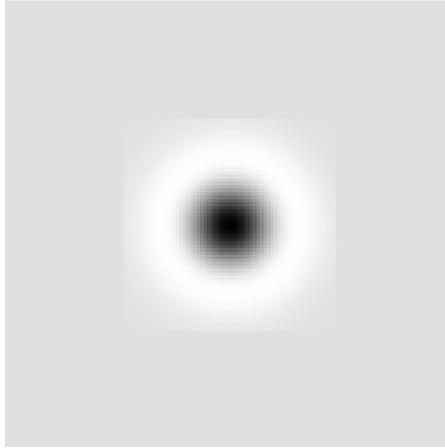


Figura 3.11: Laplaciano. Mexican Hat

3.6. Respuesta de Sistemas lineales invariantes en el tiempo a exponenciales complejas.

La importancia de las exponenciales complejas en el estudio de sistemas LTI proviene del hecho, de que la respuesta de un sistema LTI a una entrada exponencial compleja es la misma exponencial compleja modificada solo en amplitud.

$$T[z^n] = H(z)z^n$$

donde el factor complejo de la amplitud $H(z)$ será en general una función de la variable compleja s . Esto lo podemos mostrar haciendo

$$\begin{aligned}
 y(n) &= \sum_{k=-\infty}^{\infty} h(k) x(n-k) \\
 y(n) &= \sum_{k=-\infty}^{\infty} h(k) z^{(n-k)} \\
 y(n) &= \sum_{k=-\infty}^{\infty} h(k) z^n z^{-k} \\
 y(n) &= z^n \sum_{k=-\infty}^{\infty} h(k) z^{-k} \\
 y(n) &= H(z) z^n
 \end{aligned}$$

donde

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

hemos demostrado que cualquier exponencial compleja es una función característica de un sistema LIT.

Si la entrada de un sistema LIT de tiempo discreto se presenta como una combinación de exponenciales complejas, esto es, si

$$x(n) = \sum_k a_k z_k^n$$

entonces la salida es

$$y(n) = \sum_k a_k H(z_k) z_k^n$$

Transformada de Fourier

4.1. Representación de señales periódicas.

Recordemos que la exponencial compleja $e^{j(2\pi/N)n}$ es periódica con periodo N . A partir de esta podemos crear una familia de exponenciales complejas también con periodo N/k dadas por

$$\begin{aligned}\phi_k(n) &= e^{jk(2\pi/N)n} \\ \phi_k(n) &= \cos\left(\frac{2\pi k}{N}n\right) + j\text{seno}\left(\frac{2\pi k}{N}n\right)\end{aligned}\tag{4.1}$$

donde k es un número entero que representa una frecuencia fundamental. Todas estas exponenciales complejas tienen frecuencias que son múltiplos de la misma frecuencia fundamental $2\pi/N$. Así cuando $k = 0$ llamaremos a este el componente de CD de la señal (constante), cuando $k = 1$ como armónico de frecuencia $2\pi k/N$ fundamental, con $k = 2$ tenemos $4\pi k/N$ segundo armónico y así sucesivamente para todos los valores k .

Con esto queremos llegar a hacer la representación de una señal periódica como la combinación lineal de exponenciales complejas, dadas como (4.2)

$$\begin{aligned}x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)\phi_k(n) \\ x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{jk(2\pi/N)n}\end{aligned}\tag{4.2}$$

donde la ecuación (4.2) es conocida como la Transformada Discreta Inversa de Fourier (TDIF) de $x(n)$ y cuyos coeficientes son $X(k)$.

Para determinar los coeficientes X de la TDIF procedemos

$$\begin{aligned} x(0) &= \frac{1}{N} (X(0)\phi_0(0) + X(1)\phi_1(0) + X(2)\phi_2(0) + \dots + X(N-1)\phi_{N-1}(0)) \\ x(1) &= \frac{1}{N} (X(0)\phi_0(1) + X(1)\phi_1(1) + X(2)\phi_2(1) + \dots + X(N-1)\phi_{N-1}(1)) \\ x(2) &= \frac{1}{N} (X(0)\phi_0(2) + X(1)\phi_1(2) + X(2)\phi_2(2) + \dots + X(N-1)\phi_{N-1}(2)) \\ &\vdots \\ x(N-1) &= \frac{1}{N} (X(0)\phi_0(N-1) + X(1)\phi_1(N-1) + \dots + X(N-1)\phi_{N-1}(N-1)) \end{aligned}$$

En forma matricial

$$\frac{1}{N} \begin{bmatrix} \phi_0(0) & \phi_1(0) & \dots & \phi_{N-1}(0) \\ \phi_0(1) & \phi_1(1) & \dots & \phi_{N-1}(1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(N-1) & \phi_1(N-1) & \dots & \phi_{N-1}(N-1) \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

donde $\phi_k(n)$ se calcula utilizando (4.1), podemos representar en forma compacta esta ecuación como :

$$\frac{1}{N} \Phi X = x \quad (4.3)$$

Para el sistema de ecuaciones ponemos calcular la Transformada Discreta de Fourier X resolviendo 4.3

$$X = N[\Phi]^{-1}x$$

4.2. Cálculo de la Transformada Discreta de Fourier

Si bien, la Transformada Discreta de Fourier (TDF) se puede resolver a partir de un sistema de ecuaciones (4.3) como $X = \Phi^{-1}x$ no es del todo correcto, además de requerir la inversa de un sistema de ecuaciones.

Una alternativa es multiplicar, ambos lados de la ecuación (4.2), por $e^{-jr(2\pi/N)n}$

$$x(n)e^{-jr(2\pi/N)n} = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{jk(2\pi/N)n}e^{-jr(2\pi/N)n}$$

y sumar para todos los N términos de la serie

$$\begin{aligned} \sum_{n=0}^{N-1} x(n)e^{-jr(2\pi/N)n} &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X(k)e^{jk(2\pi/N)n} e^{-jr(2\pi/N)n} \\ \sum_{n=0}^{N-1} x(n)e^{-jr(2\pi/N)n} &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} e^{j(k-r)(2\pi/N)n} \\ \sum_{n=0}^{N-1} x(n)e^{-jr(2\pi/N)n} &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)s(N) \end{aligned}$$

Para la resolver $s(N)$ de la ecuación anterior

$$s(N) = \sum_{n=0}^{N-1} e^{j(k-r)(2\pi/N)n} = \sum_{n=0}^{N-1} a^n$$

con $a = e^{j(k-r)(2\pi/N)}$

La solución de $s(N)$ en el caso de ($k = r$) es:

$$s(N) = \sum_{n=0}^{N-1} e^{j(0)(2\pi/N)n} = \sum_{n=0}^{N-1} 1 = N$$

y en caso de que ($k \neq r$) haremos uso de la sucesión dada por (4.4)

$$s(N) = \sum_{n=0}^{N-1} a^n = \frac{1 - a^N}{1 - a} \quad (4.4)$$

$$s(n) = \frac{1 - e^{j(k-r)(2\pi/N)N}}{1 - e^{j(k-r)(2\pi/N)}} = \frac{1 - e^{j(k-r)(2\pi)}}{1 - e^{j(k-r)(2\pi/N)}} = \frac{1 - 1}{1 - e^{j(k-r)(2\pi/N)}} = 0$$

Por tanto si escogemos valores de r igual que k , tendremos que esta suma es igual a N y cero de lo contrario, por lo que tenemos

$$\begin{aligned} X(r) &= \sum_{n=0}^{N-1} x(n)e^{-jr(2\pi/N)n} \\ X(r) &= \sum_{n=0}^{N-1} x(n)e^{-jr(2\pi/N)n} \\ X(r) &= \sum_{n=0}^{N-1} x(n)\phi_r^*(n) \end{aligned}$$

En lo general representaremos la transformada Discreta de Fourier como (4.5)

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jr(2\pi/N)n} \quad (4.5)$$

En forma matricial lo podemos expresar (4.5) como

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} \phi_0^*(0) & \phi_1^*(0) & \cdots & \phi_{N-1}^*(0) \\ \phi_0^*(1) & \phi_1^*(1) & \cdots & \phi_{N-1}^*(1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0^*(N-1) & \phi_1^*(N-1) & \cdots & \phi_{N-1}^*(N-1) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

$$X = \Phi^* x$$

donde $\phi_k^*(n)$ es el valor conjugado de $\phi_k(n)$ y se calcula utilizando (4.1) cambiando el signo de la parte imaginaria.

4.2.1. Ejemplo

Determinar la TDF $X(k)$ para una señal $x = [1, 4, 1, 1]^T$

De acuerdo con la ecuación (4.5), tenemos que calcular la TDF X , haciendo simplemente la multiplicación:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ -j3 \\ -3 \\ j3 \end{bmatrix}$$

Si nuestra formulación es consistente la transformada Inversa de Fourier de X , nos debe dar el vector original x

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 7 \\ -j3 \\ -3 \\ j3 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 1 \\ 1 \end{bmatrix}$$

4.2.2. Ejemplo

Encontrar los coeficientes de la TDF para la señal mostrada en la Fig. 4.12, cuyos elementos son $x = [1, 1, 1, 0, 0, 0, 0, 1, 1]^T$.

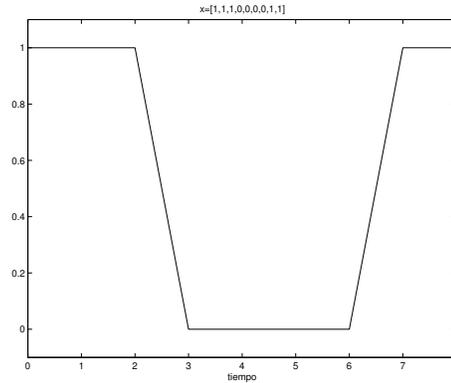


Figura 4.12: Señal a descomponer en Fourier

Para encontrar los coeficientes de Fourier de la señal x , es necesario calcular :

$$X = \Phi^* x$$

donde Φ es la matriz de exponenciales complejas (4.1) y N es la cantidad de muestras de la señal, en este caso es 9. Así la matriz es:

$$\Phi^* = \begin{pmatrix} \phi_0(0) & \phi_1(0) & \phi_2(0) & \phi_3(0) & \phi_4(0) & \phi_5(0) & \phi_6(0) & \phi_7(0) & \phi_8(0) \\ \phi_0(1) & \phi_1(1) & \phi_2(1) & \phi_3(1) & \phi_4(1) & \phi_5(1) & \phi_6(1) & \phi_7(1) & \phi_8(1) \\ \phi_0(2) & \phi_1(2) & \phi_2(2) & \phi_3(2) & \phi_4(2) & \phi_5(2) & \phi_6(2) & \phi_7(2) & \phi_8(2) \\ \phi_0(3) & \phi_1(3) & \phi_2(3) & \phi_3(3) & \phi_4(3) & \phi_5(3) & \phi_6(3) & \phi_7(3) & \phi_8(3) \\ \phi_0(4) & \phi_1(4) & \phi_2(4) & \phi_3(4) & \phi_4(4) & \phi_5(4) & \phi_6(4) & \phi_7(4) & \phi_8(4) \\ \phi_0(5) & \phi_1(5) & \phi_2(5) & \phi_3(5) & \phi_4(5) & \phi_5(5) & \phi_6(5) & \phi_7(5) & \phi_8(5) \\ \phi_0(6) & \phi_1(6) & \phi_2(6) & \phi_3(6) & \phi_4(6) & \phi_5(6) & \phi_6(6) & \phi_7(6) & \phi_8(6) \\ \phi_0(7) & \phi_1(7) & \phi_2(7) & \phi_3(7) & \phi_4(7) & \phi_5(7) & \phi_6(7) & \phi_7(7) & \phi_8(7) \\ \phi_0(8) & \phi_1(8) & \phi_2(8) & \phi_3(8) & \phi_4(8) & \phi_5(8) & \phi_6(8) & \phi_7(8) & \phi_8(8) \end{pmatrix}$$

La parte real de Φ^* es

$$\Phi_r^* = \begin{pmatrix} 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 \\ 1.000 & 0.766 & 0.174 & -0.500 & -0.940 & -0.940 & -0.500 & 0.174 & 0.766 \\ 1.000 & 0.174 & -0.940 & -0.500 & 0.766 & 0.766 & -0.500 & -0.940 & 0.174 \\ 1.000 & -0.500 & -0.500 & 1.000 & -0.500 & -0.500 & 1.000 & -0.500 & -0.500 \\ 1.000 & -0.940 & 0.766 & -0.500 & 0.174 & 0.174 & -0.500 & 0.766 & -0.940 \\ 1.000 & -0.940 & 0.766 & -0.500 & 0.174 & 0.174 & -0.500 & 0.766 & -0.940 \\ 1.000 & -0.500 & -0.500 & 1.000 & -0.500 & -0.500 & 1.000 & -0.500 & -0.500 \\ 1.000 & 0.174 & -0.940 & -0.500 & 0.766 & 0.766 & -0.500 & -0.940 & 0.174 \\ 1.000 & 0.766 & 0.174 & -0.500 & -0.940 & -0.940 & -0.500 & 0.174 & 0.766 \end{pmatrix}$$

y la parte imaginaria de Φ^* es

$$\Phi_i^* = \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.643 & -0.985 & -0.866 & -0.342 & 0.342 & 0.866 & 0.985 & 0.643 \\ 0.000 & -0.985 & -0.342 & 0.866 & 0.643 & -0.643 & -0.866 & 0.342 & 0.985 \\ 0.000 & -0.866 & 0.866 & 0.000 & -0.866 & 0.866 & 0.000 & -0.866 & 0.866 \\ 0.000 & -0.342 & 0.643 & -0.866 & 0.985 & -0.985 & 0.866 & -0.643 & 0.342 \\ 0.000 & 0.342 & -0.643 & 0.866 & -0.985 & 0.985 & -0.866 & 0.643 & -0.342 \\ 0.000 & 0.866 & -0.866 & 0.000 & 0.866 & -0.866 & 0.000 & 0.866 & -0.866 \\ 0.000 & 0.985 & 0.342 & -0.866 & -0.643 & 0.643 & 0.866 & -0.342 & -0.985 \\ 0.000 & 0.643 & 0.985 & 0.866 & 0.342 & -0.342 & -0.866 & -0.985 & -0.643 \end{pmatrix}$$

La solución del producto $(\Phi_r^* + j\Phi_i^*)x$ es:

$$X = \begin{pmatrix} 5.0000 \\ 2.87939 \\ -0.5321 \\ -1.0000 \\ 0.6527 \\ 0.6527 \\ -1.0000 \\ -0.5321 \\ 2.8794 \end{pmatrix}$$

Note que X no contiene elementos imaginario.

A continuación se va a reconstruir la señal original usando los coeficientes de Fourier obtenidos calculando,

$$y = \Phi X$$

Para poder observar la importancia de cada uno de los coeficientes encontrados, primero se va a reconstruir la señal usando un solo coeficiente de manera que $X = [X(0), 0, 0, 0, 0, 0, 0, 0, 0]^T$, después se van a usar dos coeficientes, de modo que $X = [X(0), X(1), 0, 0, 0, 0, 0, 0, 0]^T$, y así sucesivamente hasta utilizar los 9 coeficientes. Los resultados de este ejercicio se muestran en las Figs. 4.13(a)-4.13(i).

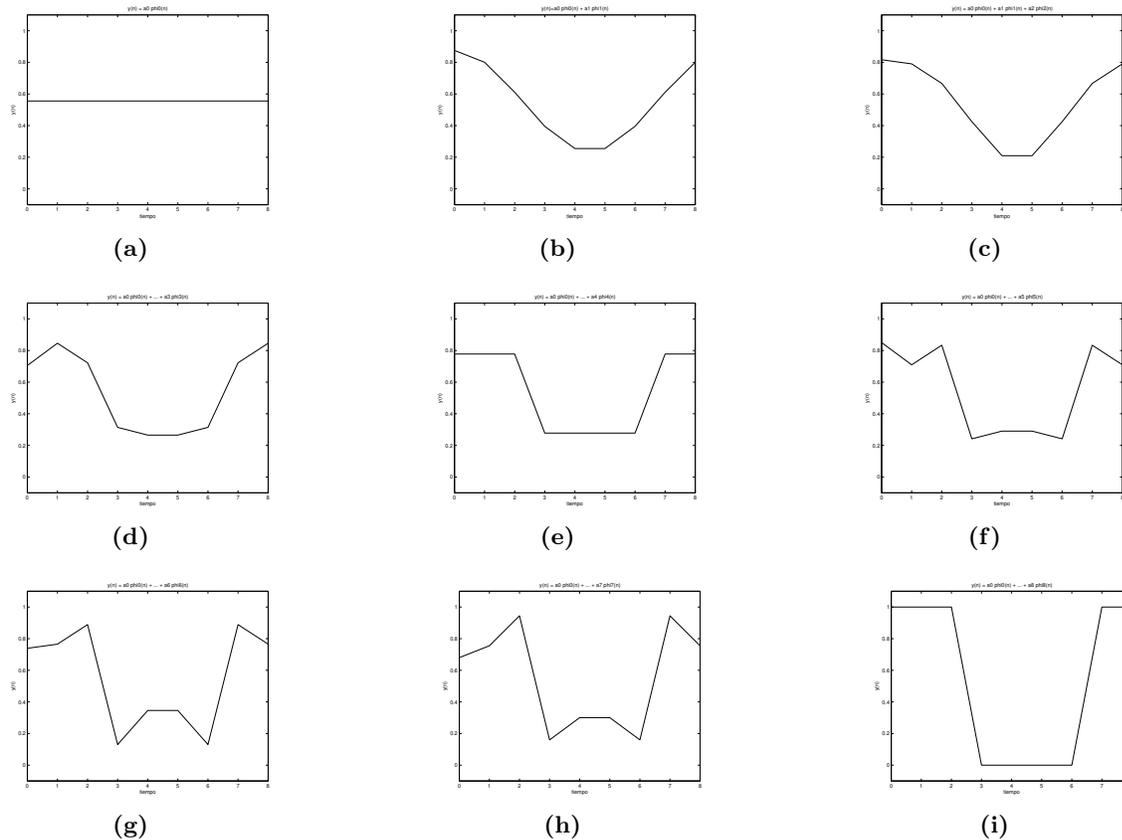


Figura 4.13: Reconstrucción de la señal usando los coeficientes de Fourier

4.3. Implementación de la Transformada de Fourier

A partir de las ecuaciones de la TDF (4.5) y de la TDIF (4.2), es necesario construir una matriz Φ^* y Φ respectivamente. Matricialmente la TDF se calcula con $X = \Phi^*x$ y TDIF como $x = \frac{1}{N}\Phi X$. Note que la diferencia es un signo en la parte imaginaria de las exponenciales complejas y la división entre N . La implementación genérica de la TDF y TDIF se muestra en el siguiente código denominado DFT_1D:

```

private static void DFT_1D(double entrada_r[], double entrada_i[],
    double salida_r[], double salida_i[], int signo){

    int N = entrada_r.length, n, k;
    double Ar[] [] = new double [N] [N];
    double Ai[] [] = new double [N] [N];
    double w = 2.0*Math.PI/(double) N, suma_r, suma_i;

    double factor;

    factor = signo == 1 ? 1.0/(double) N : 1.0 ;

    for(n=0; n<N; n++){
        for(k=n; k<N; k++) {
            Ar[n] [k] = Math.cos(w*n*k);
            Ar[k] [n] = Ar[n] [k];
            Ai[n] [k] = signo*Math.sin(w*n*k);
            Ai[k] [n] = Ai[n] [k];
        }
    }

    for(k=0; k<N; k++) {
        suma_r = 0;
        suma_i = 0;

        for(n=0; n<N; n++){
            suma_r += (Ar[k] [n]*entrada_r[n] - Ai[k] [n]*entrada_i[n]);
            suma_i += (Ar[k] [n]*entrada_i[n] + Ai[k] [n]*entrada_r[n]);
        }

        salida_r[k] = suma_r*factor;
        salida_i[k] = suma_i*factor;
    }
}

```

Para evitar confusiones, de manera particular, para el calculo de la TDF utilizaremos la función

```

public static void Fourier(double entrada_r[], double entrada_i[],
    double salida_r[], double salida_i[]){
    DFT_1D(entrada_r, entrada_i, salida_r, salida_i, -1);
}

```

y para la TDIF utilizamos

```
public static void Inversa_Fourier(double entrada_r[], double entrada_i[],
    double salida_r[], double salida_i[]){
    DFT_1D(entrada_r, entrada_i, salida_r, salida_i, 1);
}
```

Se puede comprobar los resultados de estas funciones en Mathematica haciendo `Fourier[x, FourierParameters-¿1,-1]` y `InverseFourier[X, FourierParameters-¿1,-1]`.

4.4. Propiedades de la Transformada Discreta de Fourier

La transformada de Fourier discreta, presenta algunas propiedades un cuanto diferentes que la transformada continua. A continuación se mencionan algunas y se da la demostración de tales

4.4.1. Periodicidad

Sea una señal $x(n)$, mostrar que la transformada de Fourier, $X(k)$ tiene periodo N .

Si la función es periódica entonces

$$X(k) = X(k + N)$$

La TDF (4.5) de x es

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\left(\frac{2\pi}{N}\right)nk}$$

y la TDF para $x(n + N)$ se calcula a partir de (4.5) como

$$\begin{aligned} X(k + N) &= \sum_{n=0}^{N-1} x(n)e^{-j\left(\frac{2\pi}{N}\right)n(k+N)} \\ X(k + N) &= \sum_{n=0}^{N-1} x(n)e^{-j\left(\frac{2\pi}{N}\right)nk} e^{-j\left(\frac{2\pi}{N}\right)nN} \\ X(k + N) &= \sum_{n=0}^{N-1} x(n)e^{-j\left(\frac{2\pi}{N}\right)nk} = X(k) \end{aligned}$$

Note que

$$X(k + N) = X(k)$$

4.4.2. Linealidad

Si

$$\begin{aligned}x_1(n) &\stackrel{\mathcal{F}}{\leftrightarrow} X_1(k) \\x_2(n) &\stackrel{\mathcal{F}}{\leftrightarrow} X_2(k)\end{aligned}$$

entonces

$$ax_1(n) + bx_2(n) \stackrel{\mathcal{F}}{\leftrightarrow} aX_1(k) + bX_2(k)$$

Prueba: Comenzamos por la TDF (4.5)

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

para la suma de $ax_1(n) + bx_2(n)$ queda

$$\begin{aligned}\hat{X}(k) &= \sum_{n=0}^{N-1} [ax_1(n) + bx_2(n)] e^{-j(\frac{2\pi}{N})nk} \\ \hat{X}(k) &= a \sum_{n=0}^{N-1} x_1(n)e^{-j(\frac{2\pi}{N})nk} + b \sum_{n=0}^{N-1} x_2(n)e^{-j(\frac{2\pi}{N})nk} \\ \hat{X}(k) &= aX_1(k) + bX_2(k)\end{aligned}$$

4.4.3. Desplazamiento en tiempo

Si

$$\begin{aligned}x(n) &\stackrel{\mathcal{F}}{\leftrightarrow} X(k) \\x(n - n_0) &\stackrel{\mathcal{F}}{\leftrightarrow} e^{-j(\frac{2\pi}{N})n_0k} X(k)\end{aligned}$$

Demostración:

La TDF (4.5) de la señal desplazada $x(n - n_0)$ es;

$$X(k) = \sum_{n=0}^N x(n - n_0)e^{j(\frac{2\pi}{N})nk}$$

haciendo en cambio de variable $\hat{n} = n - n_0$ tenemos

$$\begin{aligned} X(k) &= \sum_{\hat{n}=-n_0}^{N-1-n_0} x(\hat{n}) e^{-j\left(\frac{2\pi}{N}\right)(\hat{n}+n_0)k} \\ X(k) &= \sum_{\hat{n}=0}^{N-1} x(\hat{n}) e^{-j\left(\frac{2\pi}{N}\right)\hat{n}k} e^{-j\left(\frac{2\pi}{N}\right)n_0k} \\ X(k) &= e^{-j\left(\frac{2\pi}{N}\right)n_0k} \sum_{\hat{n}=0}^{N-1} x(\hat{n}) e^{-j\left(\frac{2\pi}{N}\right)\hat{n}k} \\ X(k) &= e^{-j\left(\frac{2\pi}{N}\right)n_0k} X(k) \end{aligned}$$

4.4.4. Desplazamiento en frecuencia

Ahora si hacemos un desplazamiento en frecuencia tenemos que

$$e^{j\left(\frac{2\pi}{N}\right)nk_0} x(n) \xleftrightarrow{\mathcal{F}} X(k - k_0)$$

Demostración: Escribimos la formula de la TDIF (4.2)

$$X_2(k) = \frac{1}{N} \sum_{k=0}^{N-1} X(k - k_0) e^{j\left(\frac{2\pi}{N}\right)nk}$$

haciendo el cambio de variable $\hat{k} = k - k_0$ tenemos

$$\begin{aligned} X_2(k) &= \frac{1}{N} \sum_{\hat{k}=-k_0}^{N-1-k_0} X(\hat{k}) e^{j\left(\frac{2\pi}{N}\right)n(\hat{k}+k_0)} \\ X_2(k) &= \frac{1}{N} \sum_{\hat{k}=0}^{N-1} X(\hat{k}) e^{j\left(\frac{2\pi}{N}\right)n\hat{k}} e^{j\left(\frac{2\pi}{N}\right)nk_0} \\ X_2(k) &= \frac{1}{N} e^{j\left(\frac{2\pi}{N}\right)nk_0} \sum_{\hat{k}=0}^{N-1} X(\hat{k}) e^{j\left(\frac{2\pi}{N}\right)n\hat{k}} \\ X_2(k) &= e^{j\left(\frac{2\pi}{N}\right)nk_0} x(n) \end{aligned}$$

4.4.5. Conjugación

La definición del conjugado de un numero complejo es

$$\begin{aligned} x &= R + jI \\ x^* &= R - jI \end{aligned}$$

Mostrar que:

$$x^*(n) \xleftrightarrow{\mathcal{F}} X^*(-k)$$

comenzaremos por

$$\begin{aligned} X_2(k) &= \sum_{n=0}^{N-1} x^*(n) e^{-j\left(\frac{2\pi}{N}\right)nk} \\ X_2(k) &= \left[\sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\pi}{N}\right)n(-k)} \right]^* \\ X_2(k) &= X^*(-k) \end{aligned}$$

4.4.6. Inversión en Tiempo

Mostrar que

$$x(-n) \xleftrightarrow{\mathcal{F}} X(-k)$$

Comenzaremos por calcular la transformada de Fourier de $x(-n)$

$$X_2(k) = \sum_{n=0}^{N-1} x(-n) e^{-j\left(\frac{2\pi}{N}\right)nk}$$

hacemos el cambio de variable $m = -n$

$$\begin{aligned} X_2(k) &= \sum_{m=0}^{-N+1} x(m) e^{-j\left(\frac{2\pi}{N}\right)(-m)k} \\ X_2(k) &= \sum_{m=0}^{N-1} x(m) e^{-j\left(\frac{2\pi}{N}\right)m(-k)} \\ X_2(k) &= X(-k) \end{aligned}$$

4.4.7. Escalamiento en tiempo

Sea $x(an)$ una señal multiplicada por un entero, calcular la transformada de Fourier para este caso

$$X_2(k) = \sum_{n=0}^{N-1} x(an) e^{-j\left(\frac{2\pi}{N}\right)nk}$$

haciendo $m = an$ tenemos

$$\begin{aligned} X_2(k) &= \sum_{n=0}^{N-1} x(m) e^{-j\left(\frac{2\pi}{N}\right)\left(\frac{m}{a}\right)k} \\ X_2(k) &= \sum_{m=0}^{a(N-1)} x(m) e^{-j\left(\frac{2\pi}{N}\right)m\left(\frac{k}{a}\right)} \\ &= X\left(\frac{k}{a}\right) \end{aligned}$$

Note que el rango de la TDF, también se escala.

4.4.8. Convolución

El teorema de la convolución afirma que si $\mathcal{F}[x(n)] = X(k)$ y $\mathcal{F}[y(n)] = Y(k)$ entonces

$$x(n) * y(n) = X(k)Y(k)$$

demostración:

Consideremos que la convolución está $g(n) = x(n) * y(n)$ dada por

$$g(n) = \sum_{m=0}^{N-1} x(m) y(n-m)$$

y la transformada de Fourier de $g(n)$ es

$$\begin{aligned} G(k) &= \sum_{n=0}^{N-1} [g(n)] e^{-j\left(\frac{2\pi}{N}\right)nk} \\ G(k) &= \sum_{n=0}^{N-1} \left[\sum_{m=0}^{N-1} x(m) y(n-m) \right] e^{-j\left(\frac{2\pi}{N}\right)nk} \end{aligned}$$

cambiando el orden de la sumatoria

$$G(k) = \sum_{m=0}^{N-1} x(m) \left[\sum_{n=0}^{N-1} y(n-m) e^{-j\left(\frac{2\pi}{N}\right)nk} \right]$$

haciendo $\hat{n} = n - m$

$$\begin{aligned}
G(k) &= \sum_{m=0}^{N-1} x(m) \left[\sum_{n=0}^{N-1} y(\hat{n}) e^{-j(\frac{2\pi}{N})(\hat{n}+m)k} \right] \\
G(k) &= \sum_{m=0}^{N-1} x(m) e^{-j(\frac{2\pi}{N})mk} \left[\sum_{\hat{n}=0}^{N-1} y(\hat{n}) e^{-j(\frac{2\pi}{N})\hat{n}k} \right] \\
G(k) &= \left[\sum_{m=0}^{N-1} x(m) e^{-j(\frac{2\pi}{N})mk} \right] \left[\sum_{\hat{n}=0}^{N-1} y(\hat{n}) e^{-j(\frac{2\pi}{N})\hat{n}k} \right] \\
G(k) &= X(k)Y(k)
\end{aligned}$$

4.4.9. Multiplicación

De manera inversa podemos ver que si $\mathcal{F}[x(n)] = X(k)$ y $\mathcal{F}[y(n)] = Y(k)$ entonces

$$Nx(n)y(n) = X(k) * Y(k)$$

demostración:

Consideremos que la convolución $G(k) = X(k) * Y(k)$ está dada por

$$G(k) = \sum_{l=0}^{N-1} X(l)Y(k-l)$$

y su transformada Inversa de Fourier (4.2) la podemos calcular como:

$$\begin{aligned}
g(n) &= \frac{1}{N} \sum_{k=0}^{N-1} [G(k)] e^{j(\frac{2\pi}{N})nk} \\
g(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \left[\sum_{l=0}^{N-1} X(l)Y(k-l) \right] e^{j(\frac{2\pi}{N})nk}
\end{aligned}$$

cambiando el orden de la sumatoria

$$g(n) = \frac{1}{N} \sum_{l=0}^{N-1} X(l) \left[\sum_{k=0}^{N-1} Y(k-l) e^{j(\frac{2\pi}{N})nk} \right]$$

haciendo $\widehat{k} = k - l$

$$\begin{aligned} g(n) &= \frac{1}{N} \sum_{l=0}^{N-1} X(l) \left[\sum_{\widehat{k}=-l}^{N-1-l} Y(\widehat{k}) e^{j(\frac{2\pi}{N})n(\widehat{k}+l)} \right] \\ g(n) &= N \left[\frac{1}{N} \sum_{l=0}^{N-1} X(l) e^{j(\frac{2\pi}{N})nl} \right] \left[\frac{1}{N} \sum_{\widehat{k}=0}^{N-1} Y(\widehat{k}) e^{j(\frac{2\pi}{N})n\widehat{k}} \right] \\ g(n) &= Nx(n)y(n) \end{aligned}$$

4.4.10. Diferenciación en Tiempo

Demostrar que

$$x(n) - x(n-1) \xleftrightarrow{\mathcal{F}} \left(1 - e^{-j(\frac{2\pi}{N})k}\right) X(k)$$

tenemos que

$$x(n) \xleftrightarrow{\mathcal{F}} X(k)$$

y

$$x(n-1) \xleftrightarrow{\mathcal{F}} e^{-j(\frac{2\pi}{N})k} X(k)$$

por superposición demostramos la primera.

4.4.11. Diferenciación en Frecuencia

Demostrar que

$$X(k) - X(k-1) \xleftrightarrow{\mathcal{F}} \left(1 - e^{j(\frac{2\pi}{N})k}\right) x(n)$$

tenemos que

$$X(k) \xleftrightarrow{\mathcal{F}} x(n)$$

y

$$X(k-1) \xleftrightarrow{\mathcal{F}} e^{j(\frac{2\pi}{N})k} x(n)$$

por superposición demostramos la primera.

4.4.12. Propiedades de Simetría de la transformada de Fourier.

Para cualquier secuencia $x(n)$ la parte real la podemos calcular haciendo

$$\mathcal{R}[x(n)] = \frac{1}{2} [x(n) + x^*(n)]$$

y la parte imaginaria como

$$\mathcal{I}[x(n)] = \frac{1}{2} [x(n) - x^*(n)]$$

Si $x(n)$ es una secuencia real entonces:

$$\frac{1}{2} [x(n) - x^*(n)] = 0$$

lo que es equivalente a

$$x(n) = x^*(n)$$

Al aplicar la transformada de Fourier tendremos que

$$X(n) = X^*(-n)$$

A partir de esto se observa que la parte real de $X(k)$ es una función par de k y la parte imaginaria de $X(k)$ es una función impar de k . De manera similar, la magnitud de $X(k)$ es una función par y el ángulo de fase es una función impar. Además

$$\mathcal{E}\{x(n)\} \xleftrightarrow{\mathcal{F}} \mathcal{R}\{X(n)\}$$

y

$$\mathcal{O}\{x(n)\} \xleftrightarrow{\mathcal{F}} \mathcal{I}\{X(n)\}$$

demostración:

En el caso de una señal real y par se cumple que $x(n) = x(-n)$ y que $x(n) = x^*(n)$ por lo tanto en el dominio de Fourier

$$\begin{aligned} X(k) &= X(-k) = \mathcal{R}[X(-k)] + j\mathcal{I}[X(-k)] \\ X(k) &= X^*(-k) = \mathcal{R}[X(-k)] - j\mathcal{I}[X(-k)] \end{aligned}$$

la única posibilidad de que esto sea cierto es que la parte imaginaria sea cero. Así la transformada de Fourier se transforma en

$$X(k) = \sum_{n=0}^N x(n) \cos\left(\frac{2\pi}{N}nk\right)$$

a esta ecuación es común que se le conozca como la transformada coseno.

En el caso de una señal real e impar, se cumple que $x(n) = -x(-n)$ $x(n) = x^*(n)$, entonces en el dominio de Fourier

$$\begin{aligned} X(k) &= -X(-k) = -\mathcal{R}[X(-k)] - j\mathcal{I}[X(-k)] \\ X(k) &= X^*(-k) = \mathcal{R}[X(-k)] - j\mathcal{I}[X(-k)] \end{aligned}$$

la única posibilidad de que esto ocurra es que la parte real sea cero. La transformada de Fourier en este caso queda como

$$X(k) = j \sum_{n=0}^N x(n) \operatorname{sen}\left(\frac{2\pi}{N}nk\right)$$

4.4.13. Relación de Parseval.

Si $x(n)$ y $X(k)$ son la señal y su transformada respectivamente, entonces tenemos que:

$$N \sum_{n=0}^{N-1} |x(n)|^2 \stackrel{\mathcal{F}}{\leftrightarrow} \sum_{k=0}^{N-1} |X(k)|^2$$

podemos representar como

$$N \sum_{n=0}^{N-1} x(n)x^*(n) = \sum_{k=0}^{N-1} X(k)X(k)^*$$

Tomando únicamente el segundo termino de la ecuación anterior

$$E = \sum_{k=0}^{N-1} X(k)X(k)^*$$

$$E = \sum_{k=0}^{N-1} X(k) \left[\sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\pi}{n}\right)nk} \right]^*$$

$$E = \sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} x^*(n) e^{j\left(\frac{2\pi}{n}\right)nk}$$

cambiando el orden en que se hacen las sumatorias

$$= N \sum_{n=0}^{N-1} x^*(n) \left[\frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\left(\frac{2\pi}{n}\right)nk} \right]$$

$$= N \sum_{n=0}^{N-1} x^*(n) x(n)$$

$$= N \sum_{n=0}^{N-1} |x(n)|^2$$

4.4.14. Resumen de Propiedades

Dado lo anterior podemos resumir en la siguiente tabla las propiedades de la transformada discreta de Fourier.

Propiedad	Señal en el tiempo	Señal en la frecuencia	Referencia
Periodicidad	$x(n)$	$X(k)$ periodo N	4.4.1
Linealidad	$ax_1(n) + bx_2(n)$	$aX_1(k) + bX_2(k)$	4.4.2
Desp. en tiempo	$x(n - n_0)$	$e^{-j(\frac{2\pi}{N})n_0k} X(k)$	4.4.3
Desp. en frecuencia	$e^{j(\frac{2\pi}{N})nk_0} x(n)$	$X(k - k_0)$	4.4.4
Conjugación	$x^*(n)$	$X^*(-k)$	4.4.5
Inversión en tiempo	$x(-n)$	$X(-k)$	4.4.6
Escalamiento en el Tiempo	$x(an)$	$X(k/a)$	4.4.7
Convolución	$x(n) * y(n)$	$X(k)Y(k)$	4.4.8
Multiplicación	$Nx(n)y(n)$	$X(k) * Y(k)$	4.4.9
Dif. en tiempo	$x(n) - x(n - 1)$	$(1 - e^{-j(\frac{2\pi}{N})k}) X(k)$	4.4.10
Dif. en Frecuencia	$(1 - e^{j(\frac{2\pi}{N})n}) x(n)$	$X(k) - X(k - 1)$	4.4.11
Simetría conjugada	$x(n)$ real	$X(k) = X^*(k)$	4.4.12
Simetría	$x(n)$ real y par	$X(k)$ real y par	4.4.12
Simetría	$x(n)$ real e impar	$X(k)$ imaginaria e impar	4.4.12
Parseval	$N \sum_{n=0}^{N-1} x(n) ^2$	$\sum_{k=0}^{N-1} X(k) ^2$	4.4.13

4.5. Transformada de Fourier de algunas funciones interesantes

4.5.1. Exponencial Compleja

Dado $x(n) = e^{j(\frac{2\pi}{N})k_0n}$ la transformada de Fourier esta dada como

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} x(n)e^{-j(\frac{2\pi}{N})kn} \\
 X(k) &= \sum_{n=0}^{N-1} e^{j(\frac{2\pi}{N})k_0n} e^{-j(\frac{2\pi}{N})kn} \\
 X(k) &= \sum_{n=0}^{N-1} e^{j(\frac{2\pi}{N})(k_0-k)n}
 \end{aligned}$$

La sumatoria tendrá solución N en el caso de que $k = k_0$ dado que $e^0 = 1$ y en caso

contrario aplicamos la formula (4.4)

$$s(N) = \frac{1 - a^N}{1 - a} = \frac{1 - e^{j(\frac{2\pi}{N})(k_0 - k)N}}{1 - e^{j(\frac{2\pi}{N})(k_0 - k)}} = 0$$

Por lo tanto la transformada de Fourier es

$$e^{j(\frac{2\pi}{N})k_0n} \xleftrightarrow{\mathcal{F}} N\delta(k - k_0)$$

4.5.2. Función seno

Dada la función seno, podemos hacer la representación de esta función como

$$\text{sen}\left(\frac{2\pi}{N}k_0n\right) = \frac{1}{2j} \left[e^{j(\frac{2\pi}{N})k_0n} - e^{-j(\frac{2\pi}{N})k_0n} \right]$$

dado que tenemos la suma de dos exponenciales complejas, la transformada de Fourier es

$$\text{sen}\left(\frac{2\pi}{N}k_0n\right) \xleftrightarrow{\mathcal{F}} -j\frac{N}{2}\delta(k - k_0) + j\frac{N}{2}\delta(k + k_0)$$

4.5.3. Función coseno

La representación de la función coseno la podemos dar como

$$\cos\left(\frac{2\pi}{N}k_0n\right) = \frac{1}{2} \left[e^{j(\frac{2\pi}{N})k_0n} + e^{-j(\frac{2\pi}{N})k_0n} \right]$$

de manera similar que en la función seno tenemos

$$\cos\left(\frac{2\pi}{N}k_0n\right) \xleftrightarrow{\mathcal{F}} \frac{N}{2}\delta(k - k_0) + \frac{N}{2}\delta(k + k_0)$$

4.5.4. Función impulso unitario

Recordemos que la función impulso unitario es

$$\delta(n) = \begin{cases} 1 & \text{si } n = 0 \\ 0 & \text{en caso contrario} \end{cases}$$

la transformada de Fourier para esta la calculamos

$$X(k) = \sum_{n=0}^{N-1} \delta(n - n_0) e^{-j(\frac{2\pi}{N})kn}$$

$$X(k) = e^{-j(\frac{2\pi}{N})kn_0}$$

podemos comprobar que $|e^{-j(\frac{2\pi}{N})kn_0}| = 1$ para cualquier valor que tome k por lo que finalmente la magnitud de la transformada de Fourier es

$$\delta(n - n_0) \xrightarrow{\mathcal{F}} 1$$

4.5.5. Constante

Consideremos una función $x(n) = a$ la transformada de Fourier es

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

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

la sumatoria tendrá la solución N para $k = 0$ y 0 para $k \neq 0$ por lo cual

$$a \xrightarrow{\mathcal{F}} Na\delta(k)$$

4.5.6. Escalón Unitario

La función escalón unitario esta definida como

$$u(n_0) = \begin{cases} 1 & \text{si } n \geq n_0 \\ 0 & \text{en caso contrario} \end{cases}$$

la transformada de Fourier estará dada como

$$U(k) = \sum_{n=0}^{N-1} u(n_0) e^{-j(\frac{2\pi}{N})kn}$$

$$U(k) = \sum_{n=n_0}^{N-1} e^{-j(\frac{2\pi}{N})kn}$$

$$U(k) = \sum_{n=0}^{N-1} e^{-j(\frac{2\pi}{N})kn} - \sum_{n=0}^{n_0-1} e^{-j(\frac{2\pi}{N})kn}$$

Lo cual da como resultado

$$U(k) = \begin{cases} N - n_0 & \text{si } k = 0 \\ \widehat{U}(k) & \text{en caso contrario} \end{cases}$$

con

$$\begin{aligned} \widehat{U}(k) &= -\frac{1 - e^{-j(\frac{2\pi}{N})kn_0}}{1 - e^{-j(\frac{2\pi}{N})k}} \\ \widehat{U}(k) &= -\frac{e^{-j(\frac{\pi}{N})kn_0} \left(e^{j(\frac{\pi}{N})kn_0} - e^{-j(\frac{\pi}{N})kn_0} \right)}{e^{-j(\frac{\pi}{N})k} \left(e^{j(\frac{\pi}{N})k} - e^{-j(\frac{\pi}{N})k} \right)} \\ \widehat{U}(k) &= -\frac{\sin(\frac{\pi n_0 k}{N})}{\sin(\frac{\pi k}{N})} e^{-j(\frac{\pi}{N})k(n_0-1)} \end{aligned}$$

4.5.7. Caja

Consideremos una función $x(n)$ dada por la siguiente expresión

$$c(n) = \begin{cases} 1 & \text{si } -d < n < d \\ 0 & \text{si no} \end{cases}$$

Aplicando la formula podemos ver

$$\begin{aligned} C(k) &= \sum_{n=-N/2}^{N/2} c(n) e^{-j(\frac{2\pi}{N})kn} \\ C(k) &= \sum_{n=-d}^d e^{-j(\frac{2\pi}{N})kn} \end{aligned}$$

Podemos organizar los términos de la siguiente forma

$$\begin{aligned} C(k) &= \sum_{n=-d}^{-1} e^{-j(\frac{2\pi}{N})kn} + \sum_{n=0}^d e^{-j(\frac{2\pi}{N})kn} \\ C(k) &= \sum_{n=1}^d e^{-j(\frac{2\pi}{N})k(-n)} + \sum_{n=0}^d e^{-j(\frac{2\pi}{N})kn} \\ C(k) &= \sum_{n=0}^d e^{j(\frac{2\pi}{N})kn} - 1 + \sum_{n=0}^d e^{-j(\frac{2\pi}{N})kn} \end{aligned}$$

Lo cual da como resultado

$$C(k) = \begin{cases} \text{si } k = 0 \\ \dots 2d + 1 \\ \text{si no} \\ \dots \left(\frac{1 - e^{j\frac{2\pi}{N}k(d+1)}}{1 - e^{j\frac{2\pi}{N}k}} + \frac{1 - e^{-j\frac{2\pi}{N}k(d+1)}}{1 - e^{-j\frac{2\pi}{N}k}} - 1 \right) \end{cases}$$

En la figura 4.14 se presenta la parte real de esta función ya que la imaginaria es cero.

La transformada de Fourier continua de esta función, esta dada por la siguiente expresión

$$C(k) = \int_{-\infty}^{\infty} c(t) e^{-j(\frac{2\pi}{N})kt} dt$$

al sustituir nuestra función obtenemos

$$\begin{aligned} C(k) &= \int_{-d}^d e^{-j(\frac{2\pi}{N})kt} dt \\ C(k) &= \frac{1}{[-j(\frac{2\pi}{N})k]} e^{-j(\frac{2\pi}{N})kt} \Big|_{-d}^d \\ C(k) &= \frac{1}{[-j(\frac{2\pi}{N})k]} (-2j) \text{sen} \left[\frac{2\pi}{N} kd \right] \\ C(k) &= N \frac{\text{sen} \left[\frac{2\pi}{N} kd \right]}{\pi k} \end{aligned}$$

la cual luce como 4.15

4.5.8. Ejemplos

En un programa en Java que utilice las clases funciones y gráfica, obtener y explicar brevemente la Transformada de Fourier Discreta (TFD) de las siguientes señales:

1. $x_1(n) = \cos(\frac{2\pi}{N}n) + \sin(\frac{10\pi}{N}n)$
2. $x_2(n) = \delta(n) + \delta(n - 3)$
3. $x_3(n) = 1 + \delta(n + 5) + \cos(\frac{6\pi}{N}n) + \exp^{-j(\frac{2\pi}{N})n}$

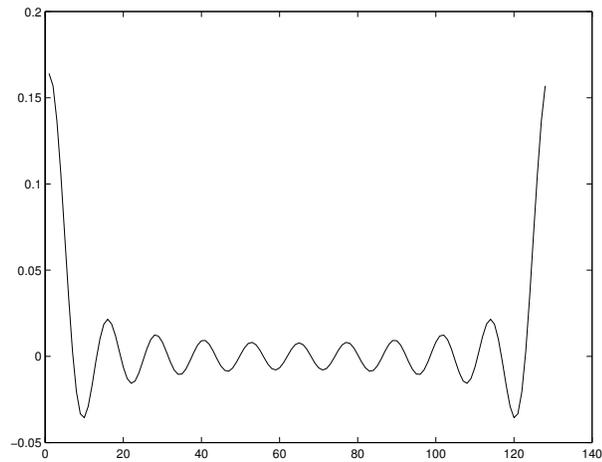


Figura 4.14: Función Sinc con $d = 10$ y $N = 128$.

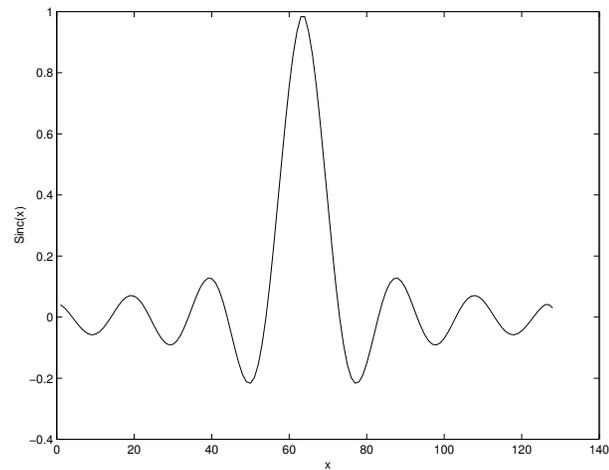


Figura 4.15: Función Sinc con $k = 10$ y $N = 128$.

Ejercicio 1

La figura 4.5.8 muestra las cuatro gráficas que se obtienen al aplicar la TFD en la señal 1. Se puede observar en la figura 4.16(a) la señal que contiene parte real pero no contiene parte imaginaria, además la parte real se forma por un seno y un coseno. El resultado de aplicar la TFD a esta señal equivale a aplicar la TFD a las dos señales que la componen por separado y luego sumarlas. En las figuras 4.16(b) y 4.16(c) se puede observar el resultado de la aplicación de la TFD a la señal 1, cuya solución analítica es:

$$X_1(k) = \frac{N}{2}\delta(k-1) + \frac{N}{2}\delta(k+1) - j\frac{N}{2}\delta(k-5) + j\frac{N}{2}\delta(k+5)$$

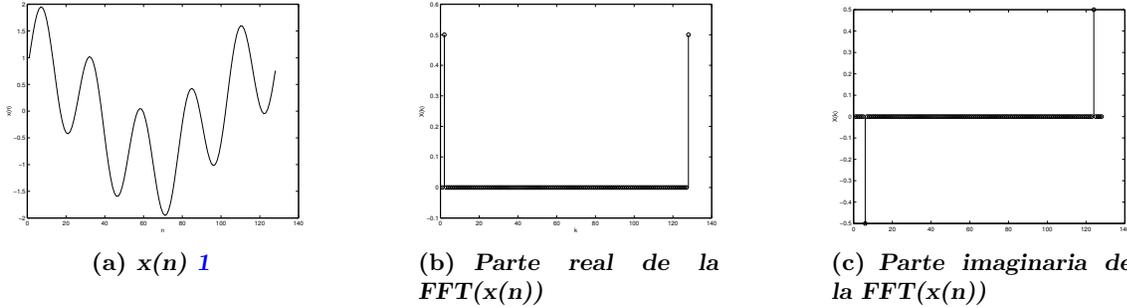


Figura 4.16: Señal 1 y su transformada de Fourier

Con las figuras podemos constatar que el coseno efectivamente contribuye con dos impulsos positivos en $k = 1$ y $k = -1$. Asimismo, el seno contribuye con dos pulsos con diferente signo en la parte imaginaria de la señal, el positivo en $k = 5$ y el negativo en $k = -5$.

Ejercicio 2

La solución analítica de la señal 2 es:

$$X_2(k) = 1 + \cos\left(\frac{6\pi}{N}k\right) - j \sin\left(\frac{6\pi}{N}k\right)$$

Las figura 4.5.8 muestra las gráficas de la señal original su TFD. Observando las gráficas 4.17(b) y 4.17(c) se puede observar que existe un coseno con frecuencia 3 en la parte real y un seno con igual frecuencia en la parte imaginaria. La contribución del impulso en el origen casi no se percibe en las gráficas, ya que es una constante que se suma al resto de la señal.

Ejercicio 3

Las figuras 4.18(a) y 4.18(b) muestra la parte real e imaginaria respectivamente de la señal 3. Al aplicar la TFD de manera analítica a la señal se obtiene:

$$X_3(k) = N\delta(k) + \cos\left(\frac{6\pi}{N}k\right) - j \sin\left(\frac{6\pi}{N}k\right) + N\delta(k+1)$$

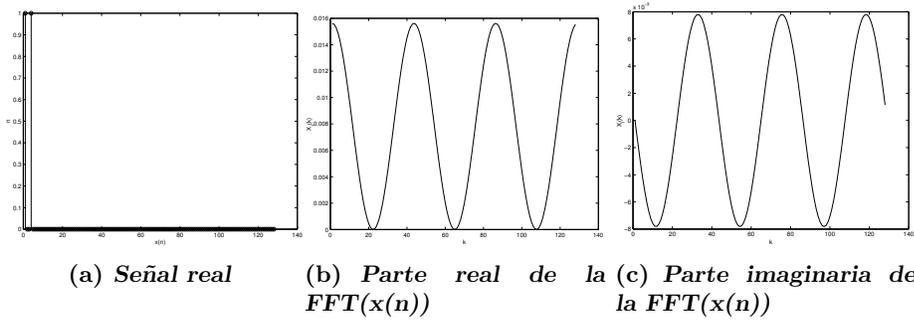


Figura 4.17: Señal 2 y su transformada de Fourier

La contribución de la constante se puede observar en la gráfica de la figura 4.18(c) como un impulso en 1 de la parte real. La contribución del impulso es coseno con frecuencia 5 en la parte real y un seno negativo con frecuencia 5 en la parte imaginaria (figura 4.18(d)). El coseno contribuye con un par de impulsos en la parte real, cuando $k = 3$ y $k = -3$, y finalmente el exponencial complejo contribuye con un pulso en $k = -1$.

4.6. Transformada de Discreta de Fourier en Dos dimensiones.

En dos dimensiones las exponenciales complejas (4.1) las expresaremos como

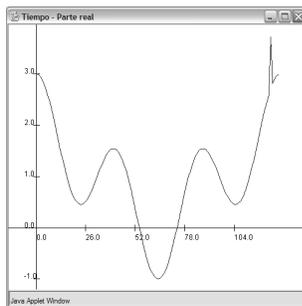
$$\begin{aligned}\phi_{k,l}(n, m) &= e^{j((2\pi nk)/N + (2\pi ml)/M)} \\ \phi_{k,l}(n, m) &= \phi_k(n)\phi_l(m)\end{aligned}\quad (4.6)$$

Definidas las exponenciales complejas (4.6) y de acuerdo con (4.5) podemos escribir la Transformada Discreta de Fourier en 2D (TDF2), como

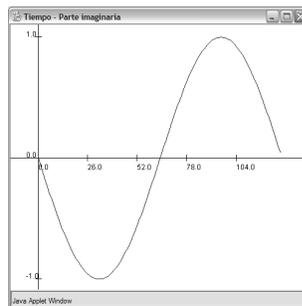
$$X(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(n, m) e^{-j\left(\frac{2k\pi}{N}n + \frac{2l\pi}{M}m\right)} \quad (4.7)$$

reorganizando términos tenemos:

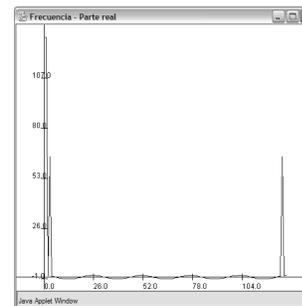
$$X(k, l) = \sum_{m=0}^{M-1} \left[\sum_{n=0}^{N-1} x(n, m) e^{-j\left(\frac{2\pi}{N}\right)nk} \right] e^{-j\left(\frac{2\pi}{M}\right)ml}$$



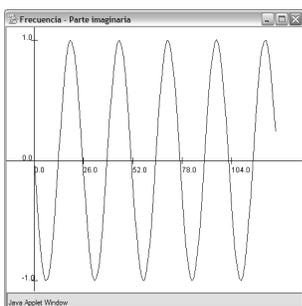
(a) Señal real en el tiempo



(b) Señal imaginaria en el tiempo



(c) Señal real en la frecuencia



(d) Señal imaginaria en la imaginaria

Figura 4.18: Señal 3 y su transformada de Fourier

La formulación de la Transformada Discreta Inversa de Fourier en dos dimensiones (TDIF2) esta dada por (4.8):

$$x(n, m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} X(k, l) e^{j(\frac{2k\pi}{N}n + \frac{2l\pi}{M}m)} \quad (4.8)$$

reorganizando términos

$$x(n, m) = \frac{1}{N} \sum_{k=0}^{N-1} \left[\frac{1}{M} \sum_{l=0}^{M-1} X(k, l) e^{j(\frac{2l\pi}{M}m)} \right] e^{j(\frac{2k\pi}{N}n)}$$

En la siguiente sección se muestra la implementación de la TDF y TDIF en dos dimensiones.

4.6.1. Ejemplo

Calcular la TDF2 para la señal bidimensional

$$x = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

De acuerdo con las formulaciones, podemos hacer la TDF en una dimensión para cada uno de los renglones para luego proceder por columna. La matriz de exponenciales complejas por columnas es

$$\Phi_r^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

y por renglones es

$$\Phi_c^* = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Comenzamos haciendo la Transformada de Fourier por renglones.

Así para el primer renglón calculamos

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 + j0 \\ -2 + j2 \\ -2 + j0 \\ -2 - j2 \end{bmatrix}$$

para el segundo renglón calculamos

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 + j0 \\ 2 - j2 \\ 2 + j0 \\ 2 + j2 \end{bmatrix}$$

Los dos renglones quedan

$$\begin{bmatrix} 10 + j0 & -2 + j2 & -2 + j0 & -2 - j2 \\ 10 + j0 & 2 - j2 & 2 + j0 & 2 + j2 \end{bmatrix}$$

4.7. IMPLEMENTACIÓN DE LA TRANSFORMADA DISCRETA DE FOURIER EN 2D67

Por columnas hacemos

$$X = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 10 + j0 & -2 + j2 & -2 + j0 & -2 - j2 \\ 10 + j0 & 2 - j2 & 2 + j0 & 2 + j2 \end{bmatrix}$$
$$X = \begin{bmatrix} 20 + j0 & 0 + j0 & 0 + j0 & 0 + j0 \\ 0 + j0 & -4 + j4 & -4 + j0 & -4 - j4 \end{bmatrix}$$

4.7. Implementación de la Transformada Discreta de Fourier en 2D

Analizando las expresiones reorganizadas de la TDF2 y TDIF2 podemos ver que son equivalentes a realizar la Transformada en una dirección del arreglo bidimensional y después realizarlo en la otra dirección. Adicionalmente en las formulaciones dadas por (4.7) y (4.8), podemos ver que la diferencia entre ambas es un signo y una división, por lo tanto la implementación genérica de ambas es:

```
private static void DFT_2D(double entrada_r[][], double entrada_i[][],
    double salida_r[][], double salida_i[][], int signo){

    int N = entrada_r.length, M = entrada_r[0].length, n, k;
    double Cr[][] = new double [M][M];
    double Ci[][] = new double [M][M];
    double Rr[][] = new double [N][N];
    double Ri[][] = new double [N][N];
    double w, suma_r, suma_i, factor =1.0;
    int i, j;
    int max = N >M ? N : M;

    double b_r[] = new double[max];
    double b_i[] = new double[max];

    w = 2.0*Math.PI/(double) N;

    for(n=0; n<N; n++){
        for(k=n; k<N; k++) {
            Rr[n][k] = Math.cos(w*n*k);
            Rr[k][n] = Rr[n][k];
            Ri[n][k] = signo*Math.sin(w*n*k);
            Ri[k][n] = Ri[n][k];
```

```

    }
}

w = 2.0*Math.PI/(double) M;

for(n=0; n<M; n++){
    for(k=n; k<M; k++) {
        Cr[n][k] = Math.cos(w*n*k);
        Cr[k][n] = Cr[n][k];
        Ci[n][k] = signo*Math.sin(w*n*k);
        Ci[k][n] = Ci[n][k];
    }
}

factor = signo == 1 ? 1.0/(double) N : 1.0 ;

for(i=0; i<N; i++) {
    for(k=0; k<M; k++) {
        suma_r = 0;
        suma_i = 0;

        for(n=0; n<M; n++){
            suma_r += (Cr[k][n]*entrada_r[i][n] - Ci[k][n]*entrada_i[i][n]);
            suma_i += (Cr[k][n]*entrada_i[i][n] + Ci[k][n]*entrada_r[i][n]);
        }
        salida_r[i][k] = suma_r*factor;
        salida_i[i][k] = suma_i*factor;
    }
}

factor = signo == 1 ? 1.0/(double) M : 1.0 ;

for(j=0; j<M; j++){

    for(k=0; k<N; k++) {
        suma_r = 0;
        suma_i = 0;

        for(n=0; n<N; n++){
            suma_r += (Rr[k][n]*salida_r[n][j] - Ri[k][n]*salida_i[n][j]);
            suma_i += (Rr[k][n]*salida_i[n][j] + Ri[k][n]*salida_r[n][j]);
        }
    }
}

```

4.7. IMPLEMENTACIÓN DE LA TRANSFORMADA DISCRETA DE FOURIER EN 2D69

```
    }
    b_r[k] = suma_r;
    b_i[k] = suma_i;
}

for(i=0; i<N; i++){
    salida_r[i][j] = b_r[i]*factor;
    salida_i[i][j] = b_i[i]*factor;
}
}
```

Para el calculo de la TDF2 y de ls TDIF haremos uso de las funciones

```
public static void Fourier(double entrada_r[][] , double entrada_i[][] ,
    double salida_r[][] , double salida_i[][]) {
    DFT_2D(entrada_r, entrada_i, salida_r, salida_i, -1);
}

public static void Inversa_Fourier(double entrada_r[][] , double entrada_i[][] ,
    double salida_r[][] , double salida_i[][]) {
    DFT_2D(entrada_r, entrada_i, salida_r, salida_i, 1);
}
```

4.7.1. Ejemplo

Transformada de Fourier de una señal bidimensional

Dada la señal $x(n,m) = \cos \left[\frac{2*\pi}{N} (k_0n + l_0m) \right]$ determinar su transformada de Fourier.

Esta señal puede ser representada como la suma de dos exponenciales complejas

$$\begin{aligned} x(n,m) &= \cos \left[\frac{2*\pi}{N} (k_0n + l_0m) \right] \\ &= \frac{1}{2} e^{j\left(\frac{2\pi}{N}\right)(k_0n+l_0m)} + \frac{1}{2} e^{-j\left(\frac{2\pi}{N}\right)(k_0n+l_0m)} \end{aligned}$$

por lo tanto

$$X(k,l) = \frac{N}{2} \delta(k - k_0, l - l_0) + \frac{N}{2} \delta(k + k_0, l + l_0)$$

En la figura 4.19 podemos ver la señal bidimensional correspondiente $x(n,m) = \cos \left[\frac{2*\pi}{N} (10n + 20m) \right]$ y su transformada de Fourier. Note que en la imagen de la transformada de Fourier apare-

cen dos picos en las coordenadas $[10, 20]$ y $[118, 108]$, lo cual, corresponde con la deducción anterior.

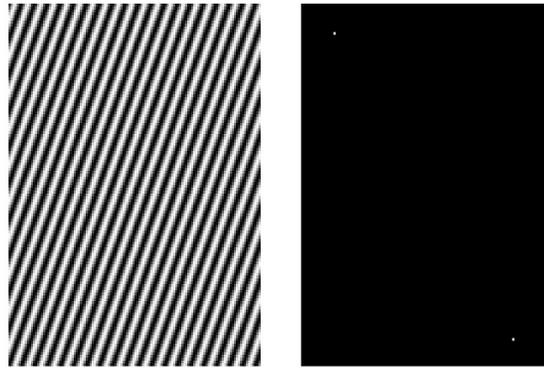


Figura 4.19: Señal coseno en dos dimensiones

4.7.2. Ejemplo

El primer renglón de una imagen es creado utilizando una distribución normal $N(0, 1)$ (ver 4.21) y los renglones subsiguientes se calculan de acuerdo con la siguiente sucesión $f(n, m) = f(n - 1, m - 1)$ tal que se produce un patron como el mostrado en la figura 4.20 (a la izquierda). Explique a que se debe que la magnitud de la transformada de Fourier esta dominado por una linea recta a 45 grados. La transformada de Fourier del ruido gaussiano la podemos ver a la derecha de la figura 4.20 .

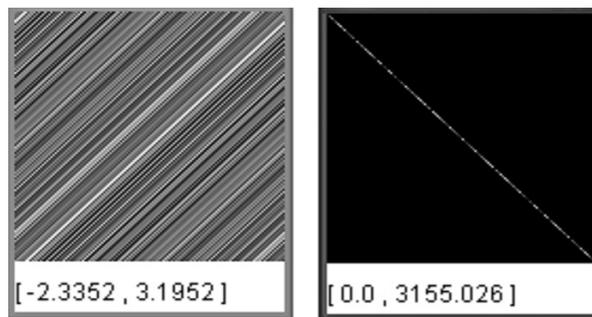


Figura 4.20: A la izquierda Patron regular creado con una señal uni-dimensional trasladada (ruido Gaussiano) y a la derecha Magnitud de su transformada de Fourier

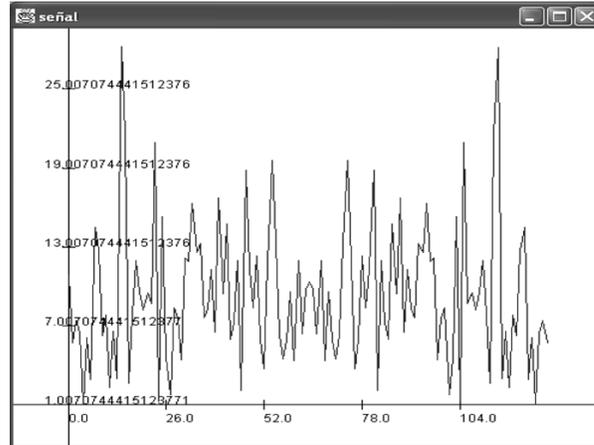


Figura 4.21: Valores aleatorios calculado con una distribución Normal con media cero y varianza unitaria

Demostración:

El primer renglón de imagen fue generado utilizando una distribución normal, y los renglones subsiguientes utilizando la recursión

$$f(n, m) = f(n - 1, m - 1)$$

la transformada de Fourier para $f(n, m)$ es $F(k, l)$ si aplicamos la propiedad de translación de la TF, podemos ver:

$$f(n, m) \xleftrightarrow{\mathcal{F}} F(k, l)$$

$$f(n - n_0, m - m_0) \xleftrightarrow{\mathcal{F}} e^{-j\left(\frac{2\pi}{NM}\right)(n_0k + m_0l)} F(k, l)$$

lo cual nos da como resultado la transformada de Fourier $F(x)$, esta multiplicada por una exponencial compleja $e^{-j\left(\frac{2\pi}{NM}\right)(n_0k + m_0l)}$ cuyo máximo se localiza sobre la recta $n_0k + m_0l = 0$, dado que $n_0 = m_0 = 1$, tendremos que la pendiente de

esta linea recta es 45 grados.

4.8. Transformada Rapida de Fourier

La transformada Discreta de Fourier TDF es $O(N^2)$ y existe la Transformada Rápida de Fourier FFT la cual es $O(N \log N)$, para una señal en una sola dimensión. Para el caso

de señales en dos dimensiones, la TDF es $O((NM)^2)$ y la FFT es $O(NM \log NM)$, razón por la cual se utiliza la FFT. Sin embargo, la FFT requiere que el tamaño de la señales sea potencia de 2, es decir $N = 2^r$ y $M = 2^s$. Todas las propiedades de la transformada de Fourier son válidas para el caso multidimensional y para hacer la demostración solo hay que tener en cuenta que se trata de arreglos bidimensionales.

Danielson y Lanczos mostraron que la transformada Discreta de Fourier para una señal x de longitud N , puede ser descrita como la suma de dos transformadas Discretas de Fourier, cada una de tamaño $N/2$. Esto lo podemos mostrar a partir de la definición de la TDF (4.5), podemos reagrupar la suma en los componentes $N/2$ pares e impares como

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

Agrupando términos tenemos

$$\begin{aligned} X(k) &= \sum_{n=0}^{N/2-1} x(2n)e^{-j(\frac{2\pi k}{N})2n} + e^{-j(\frac{2\pi k}{N})} \sum_{n=0}^{N/2-1} x(2n+1)e^{-j(\frac{2\pi k}{N})(2n)} \\ X(k) &= \sum_{n=0}^{N/2-1} x(2n)e^{-j(\frac{2\pi k}{N/2})n} + W(N) \sum_{n=0}^{N/2-1} x(2n+1)e^{-j(\frac{2\pi k}{N/2})n} \end{aligned}$$

El paso recursivo de la Transformada Rápida de Fourier es

$$TDF(x) = TDF(x_p) + W(n) \times TDF(x_i)$$

y el caso Base será cuando la señal sea de tamaño $N = 1$

$$X(0) = x(0)$$

donde x_p y x_i son vectores de tamaño $N/2$ con los coeficientes pares e impares de x y $W(N) = e^{-j(\frac{2\pi k}{N})}$.

4.8.1. Ejemplo

Calcular la Transformada Rápida de Fourier de la señal $x = [3, 2, 1, 5, 9, 4, 7, 6]^T$

$$\begin{aligned}
TDF(3, 2, 1, 5, 9, 4, 7, 6) &= TDF(3, 1, 9, 7) + W(8)TDF(2, 5, 4, 6) \\
&= TDF(3, 9) + W(4)TDF(1, 7) \\
&\quad + W(8)(TDF(2, 4) + W(4)TDF(5, 6)) \\
&= TDF(3) + W(2)TDF(9) \\
&\quad + W(4)(TDF(1) + W(2)TDF(7)) \\
&\quad + W(8)(TDF(2) + W(2)TDF(4)) \\
&\quad + W(8)W(4)(TDF(5) + W(2)TDF(6))
\end{aligned}$$

Sustituyendo valores tenemos

$$\begin{aligned}
X(k) &= 3 + 9 \times e^{-j(2\pi k/2)} + 1 \times e^{-j(2\pi k/4)} + 7 \times e^{-j(3\pi k/2)} \times e^{-j(2\pi k/4)} \\
&\quad + 2 \times e^{-j(2\pi k/8)} + 4 \times e^{-j(10\pi k/8)} + 5 \times e^{-j(6\pi k/8)} + 6 \times e^{-j(14\pi k/8)} \\
X(k) &= [37.0000, -6.70711 + j8.1213, 4.0000 + j5, -5.2929 - j3.8787, \\
&\quad 3.0000, -5.29289 + j3.8787, 4.0000 - j5.0000, -6.7071 - j8.1213]^T
\end{aligned}$$

4.8.2. Complejidad

Consideremos que el número de elementos es potencia de 2, así por ejemplo $N = 32 = 2^5$. Definimos C_1 el tiempo para calcular la TDF de una señal con un elemento y C_2 el tiempo para sumar la TDF de los elementos pares e impares. Así el tiempo para calcular la TDF de una señal con N elementos lo resolvemos con la recurrencia

$$T(N) = 2T(N/2) + NC_2$$

lo cual significa que el tiempo calcular la TDF de N datos será equivalente al tiempo de la TDF de dos señales la mitad del tamaño más el tiempo de sumar los N elementos

Por ejemplo en el caso de tener $N = 32$ elementos los tiempo los calculamos de la manera siguiente:

<i>Llamado</i>	<i>Regresa</i>
$T(32) = 2T(16) + 32C_2$	$T(32) = 32C_1 + 160C_2$
$T(16) = 2T(8) + 16C_2$	$T(16) = 16C_1 + 64C_2$
$T(8) = 2T(4) + 8C_2$	$T(8) = 8C_1 + 24C_2$
$T(4) = 2T(2) + 4C_2$	$T(4) = 4C_1 + 8C_2$
$T(2) = 2T(1) + 2C_2$	$T(2) = 2C_1 + 2C_2$
$T(1) = C_1$	C_1

De la solución podemos generalizar

$$T(N) = NC_1 + kNC_2$$

donde $k = \log_2(N)$

De acuerdo con esto la complejidad es

$$T(N) = NC_1 + N\log_2(N)C_2 = O(N\log(N))$$

4.8.3. Implementación

Para la implementación de la Transformada Rápida de Fourier, se utilizó el código de Numerical Recipes using C [Press et al., 2012] y para correr la transformada y anti transformada hacemos uso de las funciones:

En una dimensión

```
public static void Fourier_R(double entrada_r[], double entrada_i[],
double salida_r[], double salida_i[])
```

```
public static void Inversa_Fourier_R(double entrada_r[], double entrada_i[],
double salida_r[], double salida_i[])
```

En dos dimensiones

```
public static void Fourier_R(double entra_real[][], double entra_imag[][],
double sale_real[][], double sale_imag[][])
```

```
public static void Inversa_Fourier_R(double entra_real[][], double entra_imag[][],
double sale_real[][], double sale_imag[][])
```

4.9. Convolución utilizando TF.

En una imagen tenemos que la referencia se encuentra en la esquina superior izquierda. Cuando estamos utilizando la transformada de Fourier debemos recordar que está, considera que las señales tienen periodos N en la dirección de x y M en la dirección de y . Así, una imagen estará representada virtualmente, por un conjunto de copias infinitas es un espacio infinito.

Para realizar una translación de la imagen, una manera es convolucionar la imagen con la función delta de Dirac $\delta(n - n_0, m - m_0)$ donde n_0 y m_0 es el desplazamiento, hay que recordar que la convolución es Lineal e invariante a la translación, por lo que, la convolución de dos señales no se verá afectada cuando hacemos la translación del kernel.

Cuando aplicamos el teorema de la convolución

$$x(n, m) * y(n, m) \xleftrightarrow{\mathcal{F}} X(k, l)Y(K, l)$$

debemos tomar en cuenta que si el kernel se desplaza, la transformada de Fourier del kernel estará multiplicada por una exponencial compleja

$$\begin{aligned} x(n - n_0, y - m_0) &\xleftrightarrow{\mathcal{F}} X(k, l)e^{-j\frac{2\pi}{NM}(n_0k+m_0l)} \\ x(n, m) * \delta(n - n_0, m - m_0) &\xleftrightarrow{\mathcal{F}} X(k, l)e^{-j\frac{2\pi}{NM}(n_0k+m_0l)} \end{aligned}$$

Cuando aplicamos el teorema de la convolución tendremos

$$\begin{aligned} x(n, m) * [y(n, m) * \delta(n - n_0, m - m_0)] &\xleftrightarrow{\mathcal{F}} X(k, l) \left[Y(k, l)e^{-j\frac{2\pi}{NM}(n_0k+m_0l)} \right] \\ [x(n, m) * \delta(n - n_0, m - m_0)] * y(n, m) &\xleftrightarrow{\mathcal{F}} \left[X(k, l)e^{-j\frac{2\pi}{NM}(n_0k+m_0l)} \right] Y(k, l) \\ x(n - n_0, m - m_0) * y(n, n) &\xleftrightarrow{\mathcal{F}} \left[X(k, l)e^{-j\frac{2\pi}{NM}(n_0k+m_0l)} \right] Y(k, l) \end{aligned}$$

lo cual significa que $x(n - n_0, m - m_0)$, estará desplazada a las coordenadas $[n_0, m_0]$

4.10. Teorema del Muestreo para señales discretas

Consideramos el caso de una señal continua en el tiempo dada por $x(t) = \cos(15\pi t)$. Es claro que la señal tiene una frecuencia angular $\omega = 15\pi$, una frecuencia $f = 7.5$ y un periodo $T = \frac{1}{7.5}$. Esto significa que si graficamos la señal tendremos siete ciclos y medio por segundo y la gráfica para esta lucirá como la figura 4.22(a). Si esta misma función la submuestreamos con incrementos de $\frac{\pi}{9}$ lucirá como la fig 4.22(b) y esta misma a intervalos

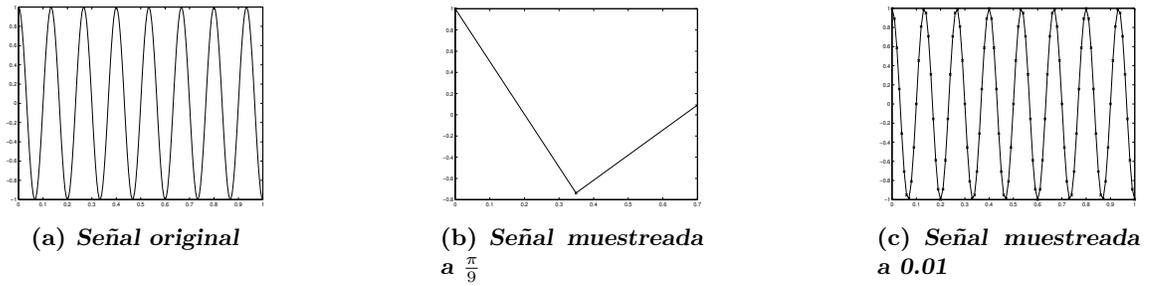


Figura 4.22: Señal $\cos(15\pi t)$ con diferentes muestreos

de 0.01 lucirá como 4.22(c). ¿Cuál es el valor de incremento al que debo discretizar mi señal sin perder información?

Podemos definir una señal muestreada $x_m(n)$ con un periodo T_m como:

$$x_m(n) = x(nT_m) \quad (4.9)$$

Aplicando (4.9) a la señal $x(t) = \cos(15\pi t)$ tenemos una señal muestreada

$$x_m(n) = \cos((15\pi T_m)n)$$

note que la frecuencia de la señal original se ve afectada por el periodo de muestreo. La transformada de Discreta de Fourier de una función coseno con periodo de muestreo T_m es:

$$\cos\left(\frac{2\pi k_0 T_m}{N}n\right) \xleftrightarrow{\mathcal{F}} \frac{N}{2}[\delta(k - T_m k_0) + \delta(k + T_m k_0)]$$

En la figura 4.23, se muestra la TDF de la señal con diferentes periodos de muestreo $T_m = 2/150$, $T_m = 4/150$, $T_m = 6/150$, $T_m = 8/150$, $T_m = 10/150$ y $T_m = 12/150$. Note en esta figura que la los impulso se encuentran en diferentes posiciones de k_0 , debido a que alteramos la frecuencia original de la señal al multiplicarla por el periodo de muestreo T_m . ¿Ahora que relación tienen?

Recordemos que la TDF solamente tiene valores de frecuencia que son múltiplos de $2\pi/N$, a acuerdo con esto, todas las frecuencias ω_0 tendrán una representación dada por:

$$\omega_0 T_m = \frac{2\pi}{N} k_0$$

y el múltiplo k_0 de la frecuencia $2\pi/N$ lo podemos encontrar como (4.10)

$$k_0 = \frac{N\omega T_m}{2\pi} \quad (4.10)$$

De acuerdo con y considerando que la señal tiene $N = 100$ muestras, en la siguiente tabla, se presentan los valores de k_0 para diferentes valores de muestreo T_m

Cuadro 4.1: Múltiplos de la frecuencia $2\pi/100$ para la señal $\cos(15\pi T_m n)$

T_m	k_0
2/150	10
4/150	20
6/150	30
8/150	40
10/150	50
12/150	60

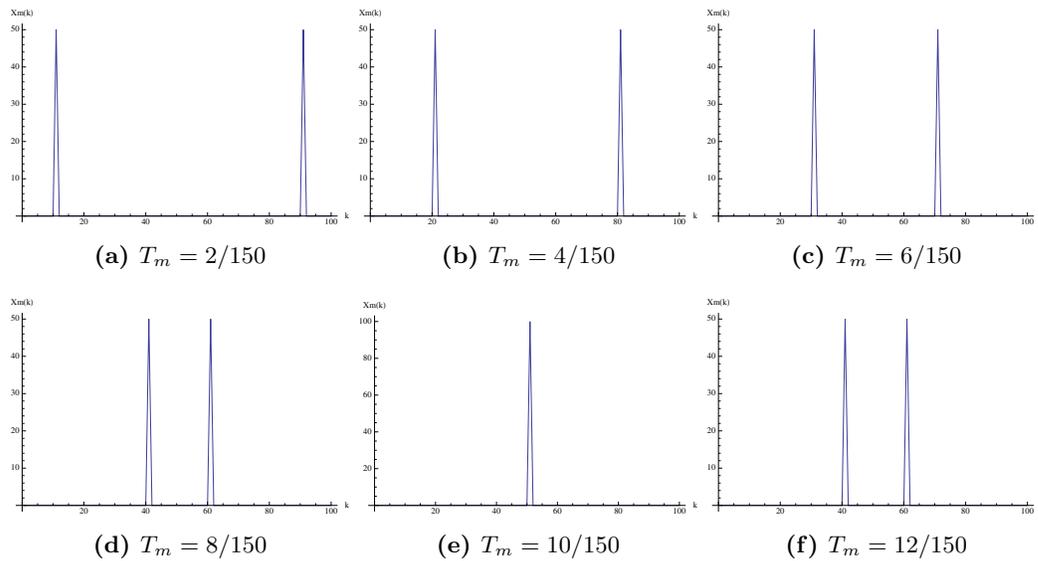


Figura 4.23: TDF de la señal $\cos(15\pi t)$ con diferentes valores de T_m

Note que cuando tenemos un periodo de muestreo $T_m = 12/150$ el múltiplo de frecuencia $k_0 = 60$. Si solamente tenemos múltiplos que van de $-N/2$ hasta $N/2$, tenemos el problema que no podemos representar la señal sin que se traslapen sus transformada.

4.10.1. Teorema del muestreo de Nyquist

De acuerdo con la antes expuesto tenemos una cuota máxima:

$$k_0 < \frac{N}{2}$$

donde k_0 es el múltiplo de nuestra frecuencia que queremos representar y $N/2$ es el múltiplo de frecuencia máxima que podemos representar.

En virtud de esto podemos calcular para una frecuencia ω_0 , muestreada con periodo T_m el valor de k_0 utilizando (4.10) y sustituyendo en la ecuación anterior

$$\begin{aligned} k_0 &= \frac{N\omega_0 T_m}{2\pi} < \frac{N}{2} \\ \omega_0 &< \frac{\pi}{T_m} \\ 2\omega_0 &< 2\frac{\pi}{T_m} \end{aligned}$$

de donde se desprende la formulación del Teorema de Nyquist

$$2\omega_0 < \omega_m \tag{4.11}$$

donde ω_m es la frecuencia de muestreo y esta definida por $\frac{2\pi}{T_m}$. Esta ecuación la podemos poner también como

$$\begin{aligned} 2\frac{2\pi}{T_0} &< \frac{2\pi}{T_m} \\ 2T_m &< T_0 \end{aligned}$$

Con esto podemos concluir que para una señal con ancho de banda ω_0 tenemos que muestrear, si queremos recuperar la señal original, al menos a una frecuencia de dos veces el ancho de banda o que el periodo de muestreo tiene que ser al menos dos veces menor que el periodo de la señal.

En la figura 4.24 se muestra una señal coseno, con un periodo $T_0 = 1333.33$, la cual fue muestreada con un periodo mayor que $T_m = 28$. Para este caso podemos notar que la señal muestreada para nada luce como una señal de la frecuencia original y el Teorema de Nyquist no se cumple ya que $2T_0 > T_m$

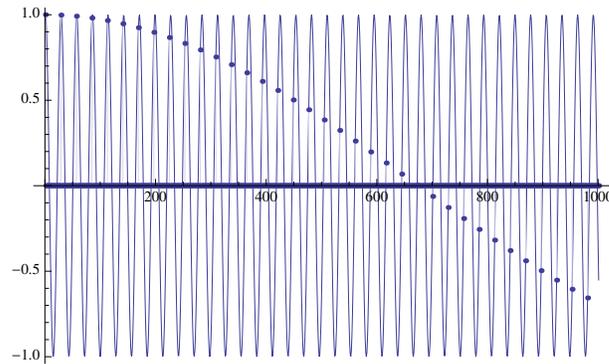


Figura 4.24: Ejemplo de una señal con bajo muestro

4.10.2. Transformada Discreta de Fourier de un tren de Pulso

La transformada de Fourier de una función $\delta(n - n_0)$ es una exponencial compleja dada por

$$X(k) = e^{-\left(\frac{2\pi}{N}n_0\right)k}$$

La transformada de Fourier de un tren de pulsos $p(n)$

$$p(n) = \sum_{m=0}^{\frac{N}{T}-1} \delta(n - mT)$$

por linealidad la podemos calcular utilizando

$$P(k) = \sum_{m=0}^{\frac{N}{T}-1} e^{-\left(\frac{2\pi}{N}mT\right)k}$$

Utilizado la (4.4) podemos ver que

$$X(k) = \frac{1 - e^{-\left(\frac{2\pi}{N}\frac{N}{T}T\right)k}}{1 - e^{-\left(\frac{2\pi}{N}T\right)k}} = 0$$

Sin embargo cuando m es múltiplo de N/T tenemos que $e^{-\left(\frac{2\pi}{N}\frac{N}{T}lT\right)k} = 1$ para $l = 0, 1, 2, \dots$

$$P(k) = \begin{cases} 1 & m = 0 \\ 1 & m = \frac{N}{T} \\ 1 & m = \frac{2N}{T} \\ 1 & m = \frac{3N}{T} \\ \vdots & \vdots \end{cases}$$

de manera compacta lo podemos escribir como

$$P(k) = \sum_{l=0}^{\infty} \delta\left(k - \frac{N}{T}l\right)$$

Podemos concluir que la TDF de un tren de pulso con periodo T es un tren de pulso con un periodo N/T . En la figura 4.25(a) se muestra un tren de pulsos con periodo $T = 6$ y su TDF en 4.25(b). Note que el periodo de la TDF es $N/T = 25$, para una señal con $N = 150$

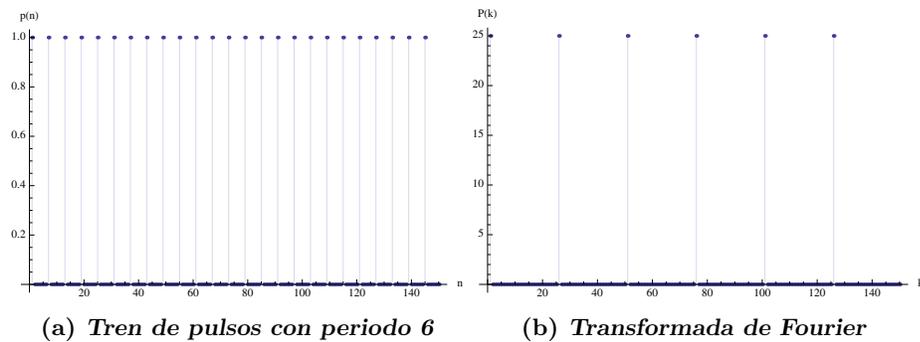


Figura 4.25: Tren de Pulsos y su transformada

Señal muestreada con un tren de pulsos

Una forma de muestrear una señal es multiplicando la señal $x(n)$ por un tren de pulso $p(n)$ cuya TDF esta dada por

$$Nx(n)p(n) \xrightarrow{\mathcal{F}} X(k) * P(k)$$

En la figura 4.26(a) se muestra una Gaussiana $g(n; \mu, \sigma)$ con media $\mu = 0$ y varianza $\sigma = 10$, la cual se muestrea con un tren de pulsos $p(n; T)$, con periodo $T = 6$, como el que se muestra en la figura 4.26(c). La transformada de furier $G(K)$ de la Gaussiana se

muestra en la figura 4.26(b) y en la figura 4.26(d) la TDF del tren de pulsos. Finalmente en la figura 4.26(e) se muestra la TDF de la función $g(n; 0, 10)$ multiplicada por el tren de pulsos $p(n; 6)$

Para el ejemplo de la figura 4.26(e) podemos notar que tenemos una cuota superior que no debemos superar. Si definimos el ancho de banda k_b como la mitad del tamaño de la Gaussiana, tenemos que para que la Gaussiana no se mezcle con los datos de otra Gaussiana, se debe cumplir:

$$2k_b < \frac{N}{T_m}$$

si multiplicamos ambos lados por $2\pi/N$

$$2 \left(\frac{2\pi}{N} \right) k_b < \left(\frac{2\pi}{N} \right) \frac{N}{T_m}$$

tenemos

$$2 \left(\frac{2\pi}{N} \right) k_b < \frac{2\pi}{T_m}$$

$$2\omega_b < \omega_m$$

que coincide con el resultado de la ecuación (4.11) del Teorema de Nyquist.

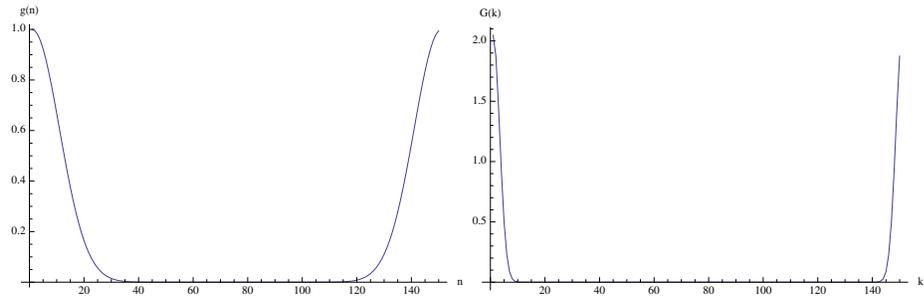
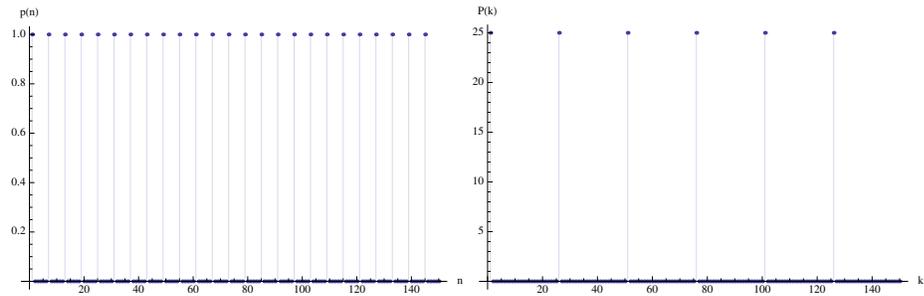
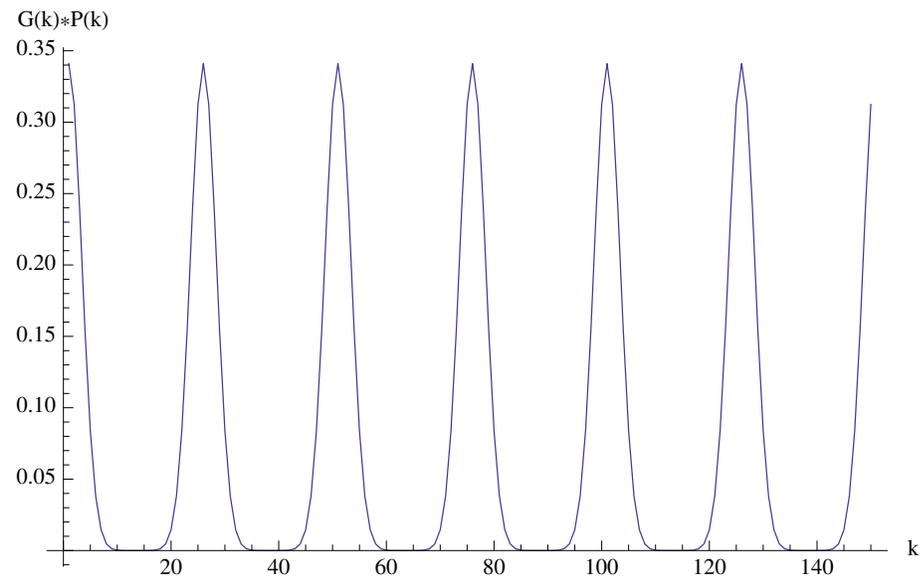
4.11. Teorema del Muestreo señales continuas

En esta sección se presenta todo el análisis con señales continuas a diferencia de la sección anterior donde se presento con señales discretas.

4.11.1. Transformada de Fourier de una señal periódica

Supongamos que una señal con periodo T_0 (con $\omega_0 = \frac{2\pi}{T_0}$), pudiera representarse como la serie

$$x(t) = \sum_{k=-\infty}^{k=\infty} X(k) e^{j\omega_0 k t} \quad (4.12)$$

(a) $g(n; 0, 10)$ (b) $TDF(g(n; 0, 10))$ (c) $p(n; 6)$ con periodo 6(d) $TDF(p(n))$ (e) $TDF(g(n; 0, 10)p(n; 6))$

multiplicando ambos miembros de la ecuación por $e^{-jn\omega_0 t}$ e integrando en el intervalo

$$\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t)e^{-jn\omega_0 t} dt = \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} \sum_{k=-\infty}^{k=\infty} X(k)e^{j\omega_0 kt} e^{-jn\omega_0 t} dt$$

reagrupando los términos tenemos

$$\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t)e^{-jn\omega_0 t} dt = \sum_{k=-\infty}^{k=\infty} X(k) \left[\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} e^{j\omega_0(k-n)t} dt \right]$$

La evaluación de la integral en los corchetes es, considerando que $k \neq n$:

$$\begin{aligned} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} e^{j\omega_0(k-n)t} dt &= \frac{1}{j(k-n)\omega_0} e^{j\omega_0(k-n)t} \Big|_{-\frac{T_0}{2}}^{\frac{T_0}{2}} \\ &= \frac{1}{j(k-n)\omega_0} \left[e^{j(k-n)\pi} - e^{-j(k-n)\pi} \right] \\ &= \frac{1}{j(k-n)\omega_0} [-1 + 1] \\ &= 0 \end{aligned}$$

en el caso de que $k = n$, la integral se transforma en

$$\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} dt = T_0$$

Por lo tanto la transformada de Fourier de una señal periódica ecuación (4.12) es

$$X(n) = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t)e^{-jn\omega_0 t} dt \quad (4.13)$$

4.11.2. Transformada de Fourier de un Tren de Pulsos

Un tren de pulsos lo podemos definir por

$$p(t) = \cdots + \delta(t + 2T) + \delta(t + T) + \delta(t) + \delta(t - T) + \delta(t + 2T) + \cdots$$

la cual podemos representar en forma compacta como

$$p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT)$$

donde $T = \frac{2\pi}{\omega_s}$. Note que esta ecuación es una función periódica y su transformada puede ser calculada utilizando el resultado de la ecuación (4.13)

En este caso hablamos de una transformada de Fourier continua por lo cual haremos la transformada de Fourier utilizando la definición continua

$$P(\omega) = \int_{-\infty}^{+\infty} p(t) e^{-j\omega t} dt$$

sustituyendo tenemos

$$P(\omega) = \sum_{k=-\infty}^{\infty} \int_{-\infty}^{+\infty} \delta(t - kT) e^{-j\omega t} dt \quad (4.14)$$

Aplicando el resultado de la ecuación (4.13) tenemos

$$P(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} \delta(t - kT) e^{-j\omega t} dt$$

de donde se desprende que si $k = 0$

$$\frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} \delta(t - kT) e^{-j\omega t} dt = \frac{1}{T} = \frac{1}{T} \delta(\omega - 0\omega_s)$$

entonces da como resultado un tren de pulsos equiespaciados en la frecuencia con $\omega_s = \frac{2\pi}{T}$.

$$P(\omega) = \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s)$$

4.11.3. Señal Muestreada en el Tiempo

Una señal muestreada se calcula como

$$x_p(t) = x(t)p(t)$$

Esto significa que tenemos que hacer la convolución de la TDF de la señal $X(k)$ con un tren de pulsos $P(k)$. En la figura se muestra una función gaussiana

donde $p(t)$ es un tren de pulsos. Aplicando el teorema de la convolución tenemos que la transformada de Fourier de esta señal es

$$\begin{aligned} X_p(\omega) &= X(\omega) * P(\omega) \\ X_p(\omega) &= X(\omega) * \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_m) \\ &= \sum_{k=-\infty}^{\infty} X(\omega - k\omega_m) \end{aligned}$$

lo cual significa que tendremos copias de nuestra señal $X(\omega)$ repetidas cada $\frac{2\pi}{T}$. ¿Pero que ocurre si el ancho de banda de la señal $X(\omega)$ sobre pasa el valor $\frac{2\pi}{T}$?

Llamemos ω_m la frecuencia de muestreo la cual esta dada como

$$\omega_m = \frac{2\pi}{T_m}$$

y ω_b el ancho de banda de la señal $x(t)$, podemos ver que si

$$\omega_m > 2\omega_b$$

podemos reconstruir exactamente la señal continua $x(t)$ a partir de la señal muestreada $x_p(t)$. La frecuencia de muestreo ω_m también se conoce como la frecuencia de Nyquist.

Podemos notar que

$$x_p(n) = x(nT) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega nT} d\omega$$

dado que la señal en el dominio de la frecuencia es periódica podemos considerar que

$$x_p(n) = \frac{1}{2\pi} \sum_{r=-\infty}^{\infty} \int_{(2r-1)\pi/T}^{(2r+1)\pi/T} X(\omega) e^{j\omega n T} d\omega$$

y cada término de la suma puede ser reducido a una integral sobre el rango $\frac{\pi}{T}$ a $\frac{\pi}{T}$ haciendo el cambio de variable $\omega = \Omega + \frac{2\pi r}{T}$

$$x_p(n) = \frac{1}{2\pi} \sum_{r=-\infty}^{\infty} \int_{-\pi/T}^{\pi/T} X\left(\Omega + \frac{2\pi r}{T}\right) e^{j\Omega n T} e^{j2\pi r n} d\Omega$$

Si intercambiamos el orden de integración y considerando que $e^{j2\pi r n} = 1$ para cualquier valor de n y r , tenemos que:

$$x_p(n) = \frac{1}{2\pi} \int_{-\pi/T}^{\pi/T} \left[\sum_{r=-\infty}^{\infty} X\left(\Omega + \frac{2\pi r}{T}\right) e^{j\Omega n T} \right] d\Omega$$

haciendo un cambio de variable $\Omega = \frac{\omega}{T}$

$$x_p(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\frac{1}{T} \sum_{r=-\infty}^{\infty} X\left(\frac{\omega}{T} + \frac{2\pi r}{T}\right) e^{j\omega n} \right] d\omega$$

de esta ecuación tenemos que

$$X_p(\omega) = \frac{1}{T} \sum_{r=-\infty}^{\infty} X\left(\frac{\omega}{T} + \frac{2\pi r}{T}\right)$$

o si hacemos $\hat{\omega} = \frac{\omega}{T}$ tenemos

$$X_p(\hat{\omega}T) = \frac{1}{T} \sum_{r=-\infty}^{\infty} X\left(\hat{\omega} + \frac{2\pi r}{T}\right)$$

4.11.4. Integración de la señal continua.

Ahora que hacemos para recuperar la señal continua a partir de la señal muestreada. Puesto que la señal $X_p(\omega)$ solo esta definida en el rango $[-\frac{\pi}{T}, \frac{\pi}{T}]$, calcularemos la antitransformada en estos límites, además $X_p(\omega T) = \frac{1}{T} X(\omega)$

$$\begin{aligned}
 x(t) &= \frac{1}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} T X_p(\omega T) e^{j\omega t} d\omega \\
 &= \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} \left[\sum_{n=-\infty}^{\infty} x(nT) e^{-j\omega nT} \right] e^{j\omega t} d\omega \\
 &= \sum_{n=-\infty}^{\infty} x(nT) \left[\frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} e^{j\omega(t-nT)} d\omega \right] \\
 &= \sum_{n=-\infty}^{\infty} x(nT) \frac{\text{sen}\left(\frac{\pi}{T}\right)(t-nT)}{\left(\frac{\pi}{T}\right)(t-nT)}
 \end{aligned}$$

El código para realizar esta integración es:

```

public static double Interpola(double x[], double t, double T)
{
    int n, N = x.length, i;
    double val =0;

    for(i=-3*N; i<3*N; i++)
    {
        n = (i\%N);
        if(n<0) n = N + n;
        val += x[n] * Sinc( (t - (double) i * T) * Math.PI / T);
    }
    return val;
}

```

4.11.5. Ejemplos

Ejemplo 1

Aquella frecuencia que de acuerdo con el teorema de muestreo, debe ser excedida por la frecuencia de muestreo se llama razón de Nyquist. Determine la razón de Nyquist para las siguientes señales

a) $x(t) = 1 + \cos(2000\pi t) + \text{sen}(4000\pi t)$.

La transformada de Fourier para esta señal es

$$x(k) = \delta(0) + \frac{1}{2} [\delta(w - 2000\pi) + \delta(w + 2000\pi)] + \frac{j}{2} [\delta(w - 4000\pi) - \delta(w + 4000\pi)]$$

lo cual indica que el espectro de frecuencia tendrá valores en el intervalo $[-4000\pi, 4000\pi]$, o bien que $\omega_b = 4000\pi$. De acuerdo con esto la razón de Nyquist es $\omega_s = 8000\pi$.

b) $x(t) = \frac{\text{sen}(200\pi t)}{\pi t}$. Podemos ver que la antitransformada de Fourier de una caja es

$$\begin{aligned} x(t) &= A \int_{-d}^d e^{j\omega t} d\omega \\ x(t) &= \frac{A}{jt} e^{j\omega t} \Big|_{-d}^d \\ x(t) &= \frac{1}{jt} (2j) \text{sen}[dt] \\ x(t) &= \frac{2A \text{sen}[dt]}{t} \end{aligned}$$

de donde podemos ver que la transformada de Fourier de esta señal es una caja de ancho $d = 200\pi$ y altura $A = \frac{1}{2\pi}$, por lo tanto su ancho de banda será de $\omega_b = 200\pi$ y su razón de Nyquist será $\omega_s = 400\pi$.

Ejemplo 2

Dada la señal continua $x(t) = \text{sen}^2(10\pi t)$, calcular:

- El ancho de banda correspondiente a esta señal
- La transformada discreta de Fourier si la señal es muestreada en con periodo $T_s = \frac{5}{2N}$ segundos, con un número de muestras $N = 128$.
- Cual es la transformada discreta de Fourier si la señal es muestreada con un periodo del doble del anterior.

Para esta señal aplicando identidades trigonométricas tenemos que:

$$x(t) = \text{sen}^2(10\pi t) = \frac{1}{2} (1 - \cos(20\pi t))$$

Respuestas

- La frecuencia del primer termino es $\omega_0 = 0$ y del segundo termino es $\omega_1 = 20\pi$, con lo cual tenemos que el ancho de banda es $\omega_b = 20\pi$

b) En este caso la frecuencia de muestreo es $\omega_m = \frac{2\pi}{\frac{5}{2N}} = \frac{4N\pi}{5}$ considerando 128 muestras la frecuencia de muestreo es $\omega_m = 321.6991$, note que

$$\begin{aligned}\omega_m &>= 2 * \omega_b 3 \\ 321.6921 &>= 40\pi\end{aligned}$$

Nuestra señal cumple con el teorema del muestreo, ahora es necesario calcular k de la frecuencia $\frac{2\pi}{N}$ para ello

$$\begin{aligned}x_p[n] &= x(nT) \\ x_p[n] &= \frac{1}{2} (1 - \cos(20\pi nT)) \\ x_p[n] &= \frac{1}{2} \left(1 - \cos\left(\frac{2\pi}{N}kn\right) \right)\end{aligned}$$

Por comparación podemos ver que $20\pi T = \frac{2\pi}{N}k$ con lo cual tenemos

$$\begin{aligned}\frac{2\pi}{N}k &= 20\pi T \\ k &= \frac{20\pi TN}{2\pi} \\ k &= \frac{20\pi \frac{5}{2N}N}{2\pi} \\ k &= 25\end{aligned}$$

Con esto tenemos que la transformada Discreta de Fourier es

$$X[k] = 64\delta(k) - 32(\delta(k - 25) + \delta(k + 25))$$

c) Si duplicamos el periodo de Muestreo $T_s = \frac{5}{N}$ aplicando los pasos anteriores tenemos

$$\begin{aligned}\frac{2\pi}{N}k &= 20\pi T \\ k &= \frac{20\pi TN}{2\pi} \\ k &= \frac{20\pi \frac{5}{N}N}{2\pi} \\ k &= 50\end{aligned}$$

y la TDF es

$$X[k] = 64\delta(k) - 32(\delta(k - 50) + \delta(k + 50))$$

Note como la frecuencia de muestreo afecta la transformada discreta de Fourier.

Ejemplo 3

Para el ejemplo anterior considerando un periodo de muestreo $T_s = 0.0390625$, calcular utilizando el código dado y los valores muestreados $x_p[n]$, los valores de la función en $t = T$, $t = 1.5T$ y $t = 2.0T$.

Solución

Para llenar el arreglo dada la función $x(t) = \text{sen}^2(10\pi t)$ se hizo

$$x_p[n] = \text{sen}^2(10\pi nT)$$

Al aplicar

	t	calculados	valores reales
T	0.0390625	0.886505	0.886505
1.5 T	0.05859375	0.928864	0.928864
2.0 T	0.078125	0.402455	0.402455

Transformadas Coseno y Wavelet

5.1. Transformada Coseno a partir de la Transformada de Fourier

Partiremos de una señal x de tamaño $2N - 1$ dada por (5.1) la cual es real y par. La transformada de Fourier de esta señal también será real y par de acuerdo con la propiedades de la Transformada de Fourier.

$$x(n) = x(2N - 1 - n) \quad (5.1)$$

de otra forma

$$\begin{aligned} x(0) &= X(2N - 1) \\ x(1) &= X(2N - 2) \\ x(2) &= X(2N - 3) \\ &\vdots \end{aligned}$$

Un ejemplo de esta señal podría ser $x = [1, 2, 3, 4, 4, 3, 2]$. La transformada de Fourier de cualquier señal x la con la forma dada por (5.1) la podemos calcular como:

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

La sumatoria la podemos dividir en dos partes

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

Si en la segunda sumatoria hacemos el cambio de variable $n = 2N - 1 - \hat{n}$ tenemos que $\hat{n} = 2N - 1 - n$ y podemos escribir

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi nk}{2N-1}} + \sum_{\hat{n}=N-1}^1 x(2N - 1 - \hat{n})e^{-j\frac{2\pi(2N-1-\hat{n})k}{2N-1}}$$

Reorganizando términos tenemos

$$X(k) = x(0) + \sum_{n=1}^{N-1} x(n)e^{-j\frac{2\pi nk}{2N-1}} + \sum_{n=1}^{N-1} x(2N - 1 - n)e^{-j\frac{2\pi(-n)k}{2N-1}} e^{-j\frac{2\pi(2N-1)k}{2N-1}} \rightarrow 1$$

Dado que $x(n) = x(2N - 1 - n)$ podemos simplificar

$$X(k) = x(0) + \sum_{n=1}^{N-1} x(n) \left(e^{-j\frac{2\pi nk}{2N-1}} + e^{+j\frac{2\pi nk}{2N-1}} \right)$$

$$X(k) = x(0) + 2 \sum_{n=1}^{N-1} x(n) \cos\left(\frac{2\pi nk}{2N-1}\right) \quad (5.2)$$

La función que resulta ya es una sucesión de cosenos por lo cual podemos decir que es una Transformada Coseno implementada a partir de la transformada de Fourier.

El Código en Java para hacer la implementación de la Transformada Coseno es:

```
static public void MiTCos(float x[], float X[]) {
    int N = x.length;
    int n, k;
    double w;

    w = (2.0*Math.PI/(2.0*N-1.0));

    for(k=0; k<N; k++){
        X[k] = x[0];
        for(n=1; n<N; n++)
            X[k] += x[n]*2.0*Math.cos(w*k*n);
    }
}
```

}
}

5.1.1. Transformada Inversa

Para llevar a cabo la transformada inversa consideraremos los mismos principios que la transformada Coseno. Por definición la transformada Inversa es:

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

La sumatoria la podemos dividir en dos partes

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

Si en la segunda sumatoria hacemos el cambio de variable $k = 2N - 1 - \hat{k}$ tenemos que $\hat{k} = 2N - 1 - k$ y podemos escribir

$$x(n) = \frac{1}{2N-1} \sum_{k=0}^{N-1} X(k) e^{j \frac{2\pi nk}{2N-1}} + \frac{1}{2N-1} \sum_{\hat{k}=N-1}^1 X(2N-1-\hat{k}) e^{j \frac{2\pi(2N-1-\hat{k})n}{2N-1}}$$

Reorganizando términos tenemos

$$x(n) = \frac{1}{2N-1} \left(X(0) + \sum_{k=1}^{N-1} X(k) e^{j \frac{2\pi nk}{2N-1}} + \sum_{k=1}^{N-1} X(2N-1-k) e^{j \frac{2\pi n(-k)}{2N-1}} e^{-j \frac{2\pi(2N-1)n}{2N-1}} \right)$$

Dado que $X(k) = X(2N - 1 - k)$ podemos simplificar

$$x(n) = \frac{1}{2N-1} \left(X(0) + \sum_{k=1}^{N-1} X(k) \left(e^{j \frac{2\pi nk}{2N-1}} + e^{-j \frac{2\pi nk}{2N-1}} \right) \right)$$

Finalmente

$$x(n) = \frac{1}{2N-1} \left(X(0) + 2 \sum_{k=1}^{N-1} X(k) \cos \left(\frac{2\pi nk}{2N-1} \right) \right) \quad (5.3)$$

5.1.2. Ejemplo

Consideremos una señal $x_1 = [1, 2, 3, 4, 4, 3, 2]$ y otra señal $x_2 = [1, 2, 3, 4]$ calcular:

1. Calcular la transformada de Fourier x_1 ,
2. Calcular la transformada Coseno dada por (5.2) para x_2 ,
3. Calcular la transformada Coseno Inversa dada por (5.3) para $X_2(k)$ y
4. Concluir

La Transformada de Fourier de x_1 es

$$X_1(k) = [19, -5.04892, -0.307979, -0.643104, -0.643104, -0.307979, -5.04892]$$

La Transformada Coseno de x_2 es

$$X_2(k) = [19.0, -5.04892, -0.307979, -0.643104]$$

La transformada Coseno inversa para $X_2(k)$ es

$$\hat{x}_2(n) = [1, 2, 3, 4]$$

Note que los términos de $X_2(k)$ corresponden exactamente a los primero 4 términos de $X_1(k)$.

5.2. Definición de la DCT

Para derivar la DCT de una señal real de con N muestras $x' = [x'(0), x'(1), \dots, x'(N-1)]$, primeramente construiremos una señal $x'(n)$ con $2N$ puntos

$$x(n) = \begin{cases} x'(n) & (0 \leq n \leq N-1) \\ x'(-n-1) & (-N \leq n \leq -1) \end{cases} \quad (5.4)$$

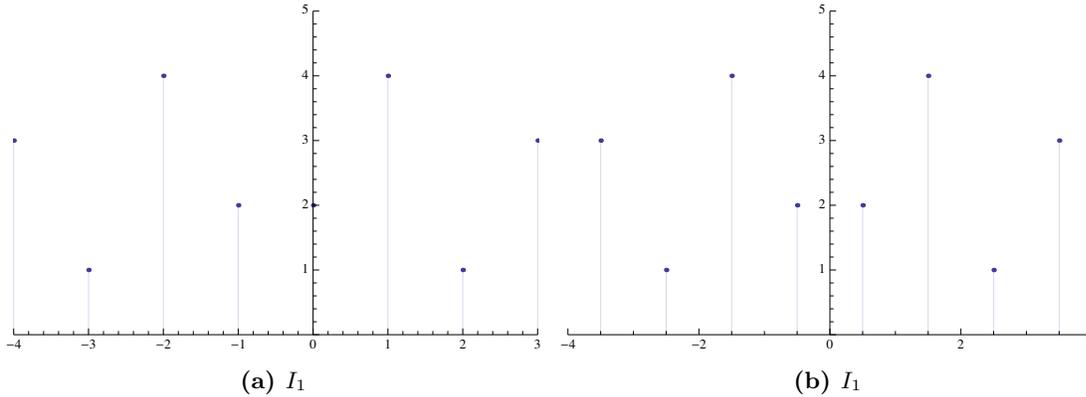


Figura 5.1: Reconstrucción de la señal usando los coeficientes de Fourier

En la Figura 5.1(a) se muestra la imagen correspondiente a la señal generada con (5.4). Para esta señal $x(n)$ asumiremos que tiene periodo $2N$ y es par con respecto al punto $n = -1/2$. Si aplicamos una translación de $1/2$ hacia la derecha de la señal $x(n)$ haciendo el cambio de variable $n' = n + 1/2$, entonces tenemos que $x(n) = x(n' - 1/2)$ y la señal resultante es par con respecto a $n' = 0$ como se muestra en la Figura 5.1(b).

La transformada discreta de Fourier para la señal $x(n' - 1/2)$ la podemos calcular con

$$X(k) = \sum_{n'=-N+1/2}^{N-1/2} x\left(n' - \frac{1}{2}\right) e^{-j2\pi n'k/2N}$$

Podemos hacer la extensión de la exponencial en su forma rectangular y dado que x es real expresar

$$X(k) = \sum_{n'=-N+1/2}^{N-1/2} x\left(n' - \frac{1}{2}\right) \cos(2\pi n'k/2N) + \sum_{n'=-N+1/2}^{N-1/2} x\left(n' - \frac{1}{2}\right) \sin(2\pi n'k/2N)$$

Dado que la señal x es par y que el $\cos(\cdot)$ y el $\sin(\cdot)$ son par e impar respectivamente con respecto a $n' = 0$ entonces la segunda sumatoria será cero.

$$X(k) = \sum_{n'=-N+1/2}^{N-1/2} x\left(n' - \frac{1}{2}\right) \cos(2\pi n'k/2N)$$

Separando la sumatoria en dos términos

$$X(k) = \sum_{n'=-N+1/2}^{-1/2} x\left(n' - \frac{1}{2}\right) \cos(2\pi n'k/2N) + \sum_{n'=1/2}^{N-1/2} x\left(n' - \frac{1}{2}\right) \cos(2\pi n'k/2N)$$

$$X(k) = \sum_{n'=1/2}^{N-1/2} x\left(-n' + \frac{1}{2}\right) \cos(-2\pi n'k/2N) + \sum_{n'=1/2}^{N-1/2} x\left(n' - \frac{1}{2}\right) \cos(2\pi n'k/2N)$$

Dado que la señal es par entonces $x(n' - 1/2) = x(-n' + 1/2)$ podemos simplificar la expresión y obtener

$$X(k) = 2 \sum_{n'=1/2}^{N-1/2} x\left(n' - \frac{1}{2}\right) \cos(2\pi n'k/2N)$$

Finalmente si hacemos el cambio de variable $n = n' - 1/2$ obtenemos

$$X(k) = 2 \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi(n+1/2)k}{N}\right)$$

5.2.1. Transformadas Coseno Utilizadas

Se puede tener un sin número de definiciones de la transformada coseno, sin embargo, centraremos en cuatro principalmente:

tipo	Definición
1	$X_1(k) = \sqrt{\frac{2}{n-1}} \left(\frac{x(0)}{2} + (-1)^s \frac{x(n-1)}{2} \sum_{n=1}^{N-2} x(n) \cos\left(\frac{\pi nk}{N-1}\right) \right)$
2	$X_2(k) = \frac{1}{\sqrt{n}} \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi(n+1/2)k}{N}\right)$
3	
4	

Filtros.

6.1. Filtros Pasa bajas.

La idea de un filtro es permitir el paso de solamente un cierto ancho de banda de un espectro de frecuencias. En el caso de un filtro pasa bajas la frecuencias que este dejará pasar son aquellas que se encuentran cerca de la frecuencia cero o componente de C.D.

El filtro pasa bajas más simple es $h(n) = [\frac{1}{2}, \frac{1}{2}]$ consideremos una señal dada por $x(n) = [1, 1, 1, 2, 2, 2, 2, 1, 1, 1]$ al realizar la convolución de esta señal con el kernel tenemos

$$\begin{aligned} z_1(n) &= x(n) * h(n) \\ &= [1, 1, 1, 2, 2, 2, 2, 1, 1, 1] * [\frac{1}{2}, \frac{1}{2}] \\ &= [1, 1, 1, 1.5, 2, 2, 1.5, 1, 1, 1] \end{aligned}$$

si volvemos a convolucionar con el mismos kernel tenemos

$$z_2(n) = [1, 1, 1, 1.25, 1.75, 1.75, 1.25, 1, 1, 1]$$

el aplicar sucesivamente el filtrado pasa-bajas dará lugar a una señal plana. En lugar de convolucionar varias veces la señal $x(n)$ con el kernel $h(n)$ resulta más practico hacer la convolución del kernel consigo mismo dando lugar a

1,1

1,2,1

1,3,3,1

1,4,6,4,1

Pero veamos la transformada de Fourier del kernel propuesto para ser filtro pasa-bajas.

$$\begin{aligned}
 H(k) &= \sum_{k=0}^N h(n)e^{-j(\frac{2\pi}{N})nk} \\
 &= \left[1 \times e^{-j(\frac{2\pi}{N})0k} + 1 \times e^{-j(\frac{2\pi}{N})1k} \right] \\
 &= \left[1 + e^{-j(\frac{2\pi}{N})k} \right]
 \end{aligned}$$

la magnitud de este filtro es

$$\begin{aligned}
 |H(k)| &= \sqrt{\left(1 + \cos \frac{2\pi k}{N}\right)^2 + \left(\text{sen} \frac{2\pi k}{N}\right)^2} \\
 &= \sqrt{1 + 2 \cos \frac{2\pi k}{N} + \cos^2 \frac{2\pi k}{N} + \text{sen}^2 \frac{2\pi k}{N}} \\
 &= \sqrt{2 + 2 \cos \frac{2\pi k}{N}} \\
 &= 2 \cos \frac{\pi k}{N}
 \end{aligned}$$

note que esta función tiene un máximo en cero y un mínimo en N , razón por la cual, deja pasar las bajas frecuencias y atenúa las altas frecuencias. En la figura podemos ver el comportamiento del filtro pasa bajas en el dominio de la frecuencia (ver figura 6.2).

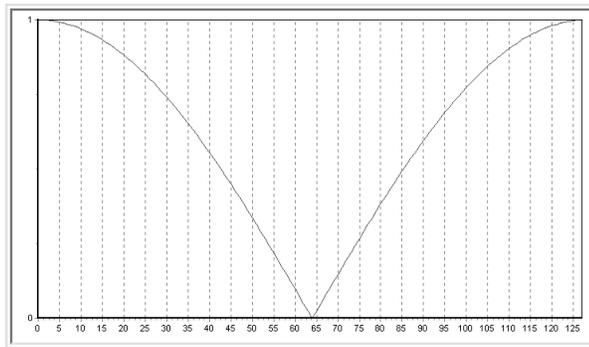


Figura 6.2: Respuesta a la frecuencia del filtro de pasabajas

6.2. Filtros Pasa Altas.

El filtro pasa bajas más simple es $h(n) = [-\frac{1}{2}, \frac{1}{2}]$ consideremos una señal dada por $x(n) = [1, 1, 1, 2, 2, 2, 2, 1, 1, 1]$ al realizar la convolución de esta señal con el kernel tenemos

$$\begin{aligned} z_1(n) &= x(n) * h(n) \\ &= [1, 1, 1, 2, 2, 2, 2, 1, 1, 1] * [-\frac{1}{2}, \frac{1}{2}] \\ &= [1, 0, 0, -0.5, 0, 0, 0.5, 0, 0, 0] \end{aligned}$$

El aplicar sucesivamente el filtrado pasabajas dará lugar a una señal plana. En lugar de convolucionar varias veces la señal $x(n)$ con el kernel $h(n)$ resulta más práctico hacer la convolución del kernel consigo mismo dando lugar a

$$\begin{aligned} &+1,-1 \\ &+1,-2,+1 \\ &+1,-3,+3,-1 \\ &+1,-4,+6,-4,-1 \end{aligned}$$

Pero veamos la transformada de Fourier del kernel propuesto para ser filtro pasa altas.

$$\begin{aligned} H(k) &= \sum_{k=0}^N h(n)e^{-j(\frac{2\pi}{N})nk} \\ &= \left[1 \times e^{-j(\frac{2\pi}{N})0k} - 1 \times e^{-j(\frac{2\pi}{N})1k} \right] \\ &= \left[1 - e^{-j(\frac{2\pi}{N})k} \right] \end{aligned}$$

la magnitud de este filtro es

$$\begin{aligned}
 |H(k)| &= \sqrt{\left(1 - \cos \frac{2\pi k}{N}\right)^2 + \left(\operatorname{sen} \frac{2\pi k}{N}\right)^2} \\
 &= \sqrt{1 - 2 \cos \frac{2\pi k}{N} + \cos^2 \frac{2\pi k}{N} + \operatorname{sen}^2 \frac{2\pi k}{N}} \\
 &= \sqrt{2 - 2 \cos \frac{2\pi k}{N}} \\
 &= 2 \operatorname{sen} \frac{\pi k}{N}
 \end{aligned}$$

note que esta función tiene un máximo en $\frac{N}{2}$ y un mínimo en 0, razón por la cual, deja pasar las altas frecuencias y atenúa las bajas frecuencias. En la figura podemos ver el comportamiento del filtro pasa bajas en el dominio de la frecuencia (ver figura 6.3).

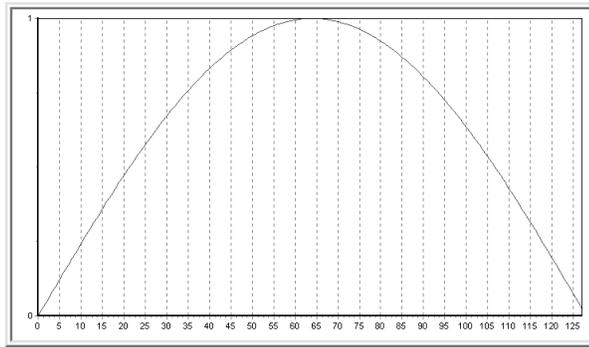


Figura 6.3: Respuesta a la frecuencia del filtro de pasa altas

6.3. Filtro pasa bajas Butterworth.

La magnitud al cuadrado de la respuesta de un filtro Butterworth esta dada por la expresión

$$|H(k)|^2 = \frac{1}{1 + \left(\frac{2\pi k}{N}\right)^{2n}}$$

Note que $|H(0)|^2 = 1$ y que $\left|H\left(\frac{N}{2\pi}\right)\right|^2 = 0.5$, lo cual le da su característica de ser un filtro pasa bajas, pero adicionalmente este filtro tiene otro parámetro de control que es el

exponente n al cual esta elevado la frecuencia. Si cambiamos este valor tendremos que la ventana del filtro se modifica tal como se observa en la figura 6.4.

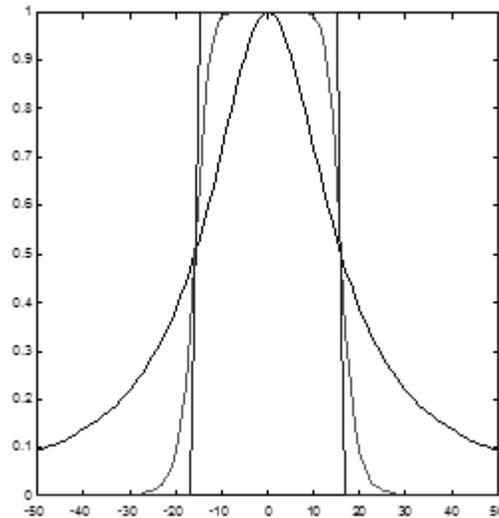


Figura 6.4: Respuesta a la frecuencia del filtro de Butherword

Podemos notar que para valores bajos de n la ventana es suave y para valores grandes tiende a una ventana cuadrada.

6.4. Filtros de pasa banda.

La ecuación de una función Gaussiana g la podemos representar por:

$$g(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

y podemos demostrar que la transformada de Fourier es otra Gaussiana G dada como

$$g(\mu, \sigma_t) \xleftrightarrow{\mathcal{F}} G(\mu, \sigma_\omega)$$

y que $\sigma_t \sigma_\omega \geq K$. Esta expresión es la formulación del principio de incertidumbre y el caso de una Gaussiana se tiene la igualdad. El principio de incertidumbre establece que no podemos tener definición simultáneamente en el dominio de la frecuencia y del tiempo.

Esta función puede ser utilizada como filtro pasa bajas cuando $\mu = 0$, pero con valores diferentes cambiaremos la banda de la señal que queremos filtrar. Así por ejemplo para filtrar una señal consideraremos que la Gaussiana tiene una media cero y que aplicaremos una translación en el tiempo y/o en la frecuencia de la señal dada por μ

$$G(k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{k^2}{2\sigma^2}}$$

si queremos hacer el filtrado en una frecuencia $w_k = \frac{2\pi}{N}k$ simplemente situamos una Gaussiana en $G(k - \mu)$ y otra en $G(k + \mu)$ la transformada de Fourier de estas será

$$\begin{aligned} G(k) &\stackrel{\mathcal{F}}{\longleftrightarrow} g(n) \\ \frac{1}{2}G(k - \mu) &\stackrel{\mathcal{F}}{\longleftrightarrow} \frac{1}{2}g(n)e^{j\left(\frac{2\pi}{N}\right)\mu n} \\ \frac{1}{2}G(k + \mu) &\stackrel{\mathcal{F}}{\longleftrightarrow} \frac{1}{2}g(n)e^{-j\left(\frac{2\pi}{N}\right)\mu n} \end{aligned}$$

el filtro en la frecuencia es la suma de $G(k - \mu)$ y $G(k + \mu)$ por lo que el filtro de convolución es la suma de estos dos

$$\begin{aligned} f_1(n) &= \frac{1}{2}g(n) \left[\cos\left(\frac{2\pi}{N}\right)\mu n + j\text{sen}\left(\frac{2\pi}{N}\right)\mu n \right] \\ f_2(n) &= \frac{1}{2}g(n) \left[\cos\left(\frac{2\pi}{N}\right)\mu n - j\text{sen}\left(\frac{2\pi}{N}\right)\mu n \right] \\ f_1(n) + f_2(n) &= g(n) \cos\left(\frac{2\pi}{N}\right)\mu n \end{aligned}$$

6.4.1. Ejemplo

Una señal $x(t) = \text{seno}(10t) + N(0, 0.3)$, esta corrompida con ruido de Naturaleza Gaussiana con media cero y varianza 0.3. Entonar calcular la media y la varianza de un filtro Gaussiano pasa banda dada una frecuencia de muestreo $T_s = 0.05$ seg y un número total de muestras de $N = 64$

Podemos calcular el múltiplo de la frecuencia $\frac{2\pi}{N}$ recordando que la frecuencia de la señal muestreada es $\omega_s = \omega t$. Con esto tenemos que

$$\begin{aligned}
 k &= \frac{\omega TN}{2\pi} \\
 k &= \frac{10 \times 0.05 \times 64}{2\pi} \\
 k &= 5.0929
 \end{aligned}$$

Haciendo esto tenemos que nuestra Gaussiana la colocaremos en $\mu = 5.0929$ y dado que no tenemos una frecuencia entera utilizaremos una varianza $\sigma = 0.5$. En la figura 6.5(a), se muestra la señal original con ruido y en las figuras 6.5(b) y 6.5(c) la parte real e imaginaria de su transformada Discreta de Fourier. En la figura 6.5(d) se muestra las Gaussiana entonadas para filtrar la señal y en la figura 6.5(e) el resultado de haber aplicado el filtro.

6.5. Filtro de Gabor

Pero que ocurre si en lugar de colocar un filtro en $\frac{1}{2}G(k - \mu)$ y otro en $\frac{1}{2}G(k + \mu)$, pusiéramos uno solo en $G(k - \mu)$

$$f(n) = g(n) \left[\cos\left(\frac{2\pi}{N}\right) \mu n + j \operatorname{sen}\left(\frac{2\pi}{N}\right) \mu n \right]$$

note que la parte real es equivalente a la encontrada anteriormente, pero además tenemos una parte compleja. El hecho de tener solamente un filtro G en el dominio de la frecuencia, nos da más información acerca de la banda que andamos buscando

$$\begin{aligned}
 f_R(n) &= g(n) \cos\left(\frac{2\pi}{N}\right) \mu n \\
 f_I(n) &= g(n) \operatorname{sen}\left(\frac{2\pi}{N}\right) \mu n
 \end{aligned}$$

a este tipo de filtro se le conoce como filtro de Gabor. La magnitud del filtro nos dará los puntos de la señal donde la frecuencia para la cual entonamos el filtro produce esa señal, siendo mas alta en los puntos de dicha frecuencia.

6.5.1. Filtro de Gabor en dos dimensiones

Para el caso de dos dimensiones el Kernel de Gabor que utilizaremos es

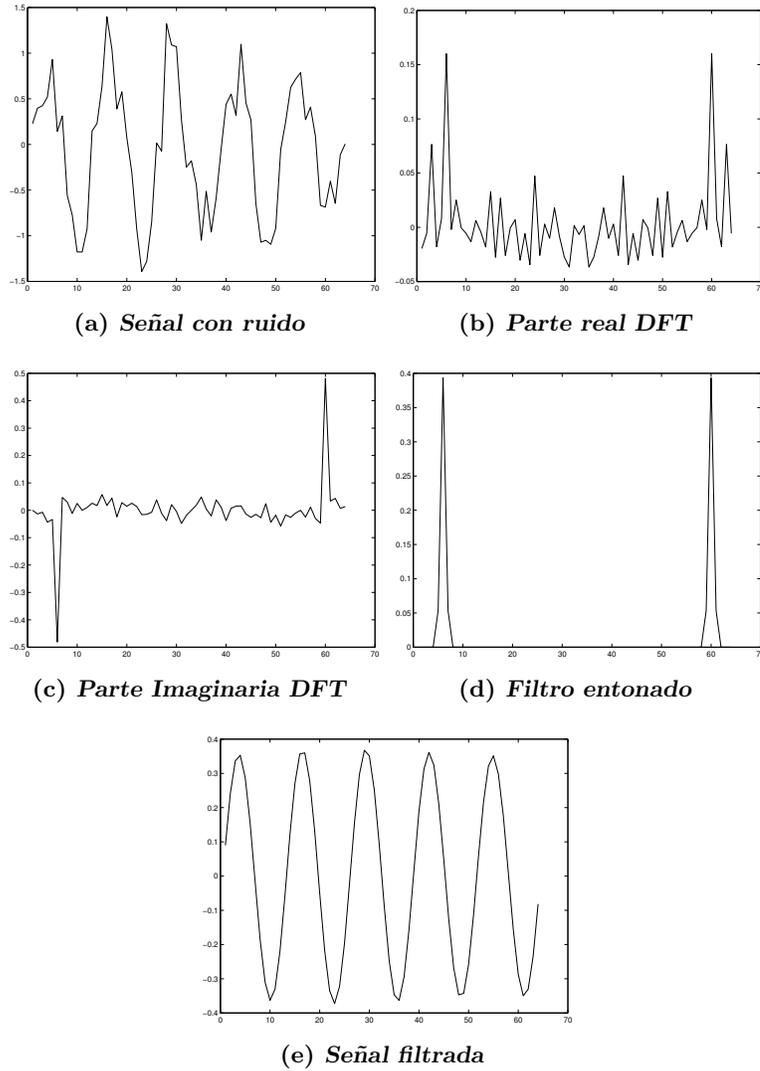


Figura 6.5: Ejemplo de filtrado de una señal con ruido

$$f(n, m) = g(n, m)e^{j(\frac{2\pi}{N}nk_0 + \frac{2\pi}{M}ml_0)}$$

La parte real del filtro es

$$f_R(n, m) = g(n, m) \cos\left(\frac{2\pi}{N}nk_0 + \frac{2\pi}{M}ml_0\right)$$

y la parte imaginaria es

$$f_I(n, m) = g(n, m) \sin\left(\frac{2\pi}{N}nk_0 + \frac{2\pi}{M}ml_0\right)$$

Dado que la convolución requiere de $(NM)^2$ operaciones, resulta mas rápido implementar el kernel de Gabor de manera separable, así el número de operaciones será $N^2M + NM^2$. De manera separable la parte real queda

$$\begin{aligned} f_R(n, m) &= g(n, m) \cos\left(\frac{2\pi}{N}nk_0 + \frac{2\pi}{M}ml_0\right) \\ f_R(n, m) &= g(n) * g(m) \left[\cos\left(\frac{2\pi}{N}nk_0\right) \cos\left(\frac{2\pi}{M}ml_0\right) - \sin\left(\frac{2\pi}{N}nk_0\right) \sin\left(\frac{2\pi}{M}ml_0\right) \right] \\ f_R(n, m) &= \left[g(n) \cos\left(\frac{2\pi}{N}nk_0\right) \right] * \left[g(m) \cos\left(\frac{2\pi}{M}ml_0\right) \right] - \left[g(n) \sin\left(\frac{2\pi}{N}nk_0\right) \right] * \left[g(m) \sin\left(\frac{2\pi}{M}ml_0\right) \right] \end{aligned}$$

Si definimos

$$\begin{aligned} g_1(n) &= g(n) \cos\left(\frac{2\pi}{N}nk_0\right) \\ g_2(n) &= g(n) \sin\left(\frac{2\pi}{N}nk_0\right) \\ g_3(m) &= g(m) \cos\left(\frac{2\pi}{M}ml_0\right) \\ g_4(m) &= g(m) \sin\left(\frac{2\pi}{M}ml_0\right) \end{aligned}$$

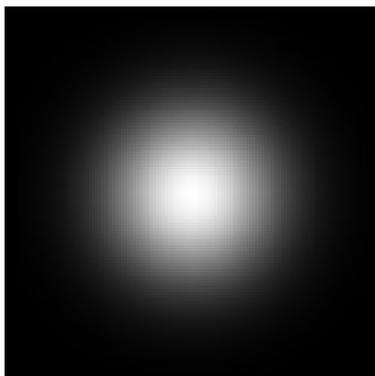
podemos escribir la parte real como

$$f_R(n, m) = g_1(n) * g_3(m) - g_2(n) * g_4(m)$$

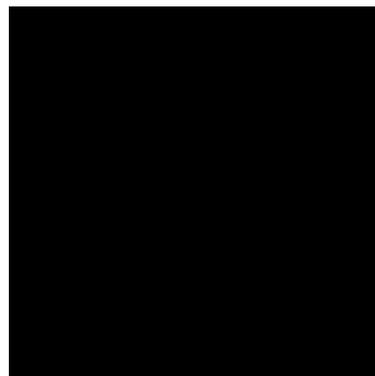
De forma similar podemos calcular la parte imaginaria y llegar al siguiente kernel separable.

$$f_I(n, m) = g_1(n) * g_4(m) + g_2(n) * g_3(m)$$

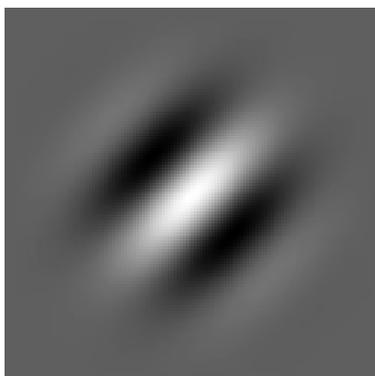
En la figura 6.5.1, podemos ver unos ejemplos del filtro de Gabor para diferentes valores de frecuencia espacial.



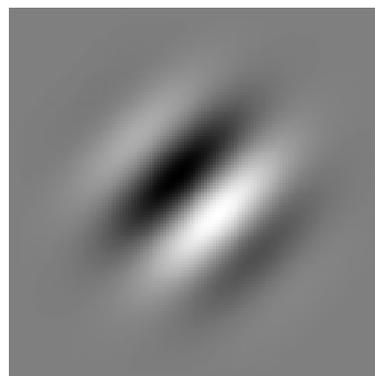
(a) Parte real, entonado en la frecuencia espacial (0,0)



(b) Parte imaginaria, entonado en la frecuencia espacial (0,0)



(c) Parte real, entonado en la frecuencia espacial (2,2)



(d) Parte imaginaria, entonado en la frecuencia espacial (2,2)

Figura 6.6: Filtros de Gabor

Algorithm 1 Filtro de Gabor

- 1: Dadas $k_0, l_0, \sigma_n, \sigma_m$ y f
- 2: **for** $n = -3\sigma_n$ **to** $3\sigma_n$ **do**
- 3: $g[n] = \frac{1}{\sqrt{2\pi\sigma}} \exp^{-n^2/(2\sigma_n^2)}$
- 4: $g_1[n] = g[n] \times \cos(2\pi nk_0/N)$
- 5: $g_2[n] = g[n] \times \sin(2\pi nk_0/N)$
- 6: **end for**
- 7: **for** $m = -3\sigma_m$ **to** $3\sigma_m$ **do**
- 8: $g[m] = \frac{1}{\sqrt{2\pi\sigma}} \exp^{-m^2/(2\sigma_m^2)}$
- 9: $g_3[m] = g[m] \times \cos(2\pi ml_0/N)$
- 10: $g_4[m] = g[m] \times \sin(2\pi ml_0/N)$
- 11: **end for**
- 12: Calcular $f_r^{(1)}[n, m] = f[n, m] * g_1[n] * g_3[m]$
- 13: Calcular $f_r^{(2)}[n, m] = f[n, m] * g_2[n] * g_4[m]$
- 14: Calcular $f_i^{(1)}[n, m] = f[n, m] * g_1[n] * g_4[m]$
- 15: Calcular $f_i^{(2)}[n, m] = f[n, m] * g_2[n] * g_3[m]$
- 16: **for** $n = 0$ **to** $N - 1$ **do**
- 17: **for** $m = 0$ **to** $M - 1$ **do**
- 18: Calcular $f_r[n, m] = f_r^{(1)}[n, m] - f_r^{(2)}[n, m]$
- 19: Calcular $f_i[n, m] = f_i^{(1)}[n, m] + f_i^{(2)}[n, m]$
- 20: **end for**

6.6. Filtro de Wiener

Dadas dos imágenes f' y f y un kernel $h[n]$ podemos decir:

$$f'[n] = f * h[n]$$

lo cual indica que si realizamos convolución con un kernel $h[n]$ de $f[n]$, tendremos una imagen $f'[n]$. Aplicando transformada de Fourier podemos intentar calcular el valor del kernel $h(n)$ haciendo:

$$\begin{aligned} f'[n] &= f * h[n] \\ F'[k] &= F[k]H[k] \end{aligned}$$

de esta expresión podemos despejar el valor de $H[k] = F'[k]/F[k]$, pero si un valor de $F[k] = 0$, no tenemos manera de estimar el valor del mismo.

En su lugar utilizaremos técnicas de Regularización, para ello, minimizamos la función

$$\begin{aligned} &|f' - f * g|^2 \\ \text{sugeto a } &|l * g|^2 \end{aligned}$$

donde $f'[n]$ es la imagen filtrada, $f[n]$ es la imagen origen, $l[n]$ es un operador lineal que escogeremos de la mejor manera para facilitar la solución y $g[n]$ es el kernel que intentamos encontrar. Para nuestro calculo procedemos

$$\begin{aligned} U(g) &= \sum_{n=0}^{N-1} [f'[n] - f * g[n]]^2 + \beta \sum_{n=0}^{N-1} |l * g[n]|^2 \\ &= \sum_{n=0}^{N-1} \left[f'[n] - \sum_{i=0}^{N-1} f[n-i]g[i] \right]^2 + \beta \sum_{n=0}^{N-1} \left[\sum_{i=0}^{N-1} l[n-i]g[i] \right]^2 \end{aligned}$$

Para encontrar el valor de g que minimiza esta expresión hacemos

$$\begin{aligned} \frac{\partial U(g)}{\partial g[i]} &= -2 \sum_{n=0}^{N-1} \left[f'[n] - \sum_{i=0}^{N-1} f[n-i]g[i] \right] f[n-i] \\ &\quad + 2\beta \sum_{n=0}^{N-1} \left[\sum_{i=0}^{N-1} l[n-i]g[i] \right] l[n-i] \\ &= 0 \end{aligned}$$

reorganizado términos tenemos

$$\begin{aligned} \sum_{n=0}^{N-1} \left[\sum_{i=0}^{N-1} f[n-i]g[i] \right] f[n-i] + \beta \sum_{n=0}^{N-1} \left[\sum_{i=0}^{N-1} l[n-i]g[i] \right] l[n-i] &= \sum_{n=0}^{N-1} f'[n]f[n-i] \\ &= [f * f^r + \beta l * l^r] * g[n] = f' * f^r[n] \end{aligned}$$

Al aplicar transformada de Fourier tenemos:

$$\begin{aligned} G[k] &= \frac{[F'[k]]^* F[k]}{[F[k]]^* F[k] + \beta [L[k]]^* L[k]} \\ G[k] &= \frac{[F'[k]]^* F[k]}{|F[k]|^2 + \beta |L[k]|^2} \end{aligned}$$

al sacar la antitransformada de Fourier calculamos el kernel $g[n]$ que aproxima al kernel original $h[n]$.

En dos dimensiones el filtro queda

$$G[k, l] = \frac{[F'[k, l]]^* F[k, l]}{|F[k, l]|^2 + \beta |L[k, l]|^2} \quad (6.5)$$

donde $F'[k, l]^*$ es el conjugado de la imagen.

En general podemos considerar que $L[k, l] = cte$ y simplificar la expresión

$$G[k, l] = \frac{[F'[k, l]]^* F[k, l]}{|F[k, l]|^2 + \beta} \quad (6.6)$$

Algorithm 2 Filtro de Wiener

```

1: Dadas  $f$ ,  $f'$  y  $\beta$ 
2: Calcular  $F = TDF[f]$  y  $F' = TDF[f']$ 
3: for  $k = 0$  to  $N - 1$  do
4:   for  $l = 0$  to  $M - 1$  do
5:     Calcular  $G[k, l] = \frac{F'[k, l]^* F[k, l]}{|F[k, l]|^2 + \beta}$ 
6:   end for
7: end for
8: Calcular  $g = TDF^{-1}[G]$ 
9: return  $g$ 

```

6.7. Como entonar un filtro

Resulta más natural al momento de entonar un filtro, hablar de frecuencias cuyas unidades son ciclos por segundo o Hertz. Dado el proceso de discretización, perdemos la noción del tiempo y nuestra frecuencia en el dominio discreto ahora esta dada en ciclos/muestra. La pregunta es como convertir Hertz a ciclos/muestra.

Consideremos una función muestreada con tamaño N , en el dominio de Fourier en la posición $k = N$ tenemos que N representa la frecuencia angular 2π , por lo tanto, podemos representar todas las frecuencia como múltiplos de $\frac{2\pi}{N}$ como $\omega = \frac{2\pi}{N}k$ [rad/muestra] y la frecuencia es $f_k = \frac{k}{N}$ [ciclos/muestra]. Dada una frecuencia de muestreo f_m en [muestras/seg] podemos calcular el tiempo necesario para una muestra, así : $1muestra = (1/f_m) seg$, sustituyendo esto tenemos:

$$f_t = \frac{k}{N} [ciclos / (1/f_m) seg]$$

$$f_t = \frac{f_m k}{N} [ciclos/seg]$$

$$f_t = f_m f_k [ciclos/seg]$$

finalmente $k = \frac{f_t}{f_m} N$.

6.7.1. Ejemplo

Consideremos una señal $x(t) = sen(200\pi t) + ruido$, la cual es muestreada con una $f_s = 800$ Hz, calcular el valor al cual debe ser entonado el filtro pasa bandas para un tamaño de la muestra $N = 128$.

La frecuencia de la señal es $f_t = 200\pi/(2\pi) = 100$ hz, por lo tanto el valor al que debe entonarse es $k = \frac{100}{800}128 = 16$. De otra forma

$$x(t) = \text{sen}(200\pi t) = \text{sen}\left(200\pi \frac{n}{800}\right) = \text{sen}\left(\frac{2\pi}{128}16n\right)$$

6.8. Filtro Elimina Banda

Dada una señal $x(n)$ el filtro pasa banda nos permitirá para la señal dada, solo mostrar la señal para la cual fue entonado el filtro. Pero como hacer si queremos mostrar la señal eliminando una banda dada.

En el dominio de la frecuencia multiplicamos la transformada de Fourier de la señal por una Gaussiana posicionada en k_0 , esto es

$$\hat{X}_{k_0}(k) = X(k) \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(k-k_0)^2}{2\sigma_k^2}}$$

Para eliminar la señal hacemos

$$\hat{X}_{-k_0}(k) = X(k) \frac{1}{\sqrt{2\pi}\sigma_k} \left[1 - e^{-\frac{(k-k_0)^2}{2\sigma_k^2}} \right]$$

y aplicamos la transformada de Fourier

$$\begin{aligned} \hat{x}_{-k_0}(n) &= x(n) * \text{cte} \left[\delta(n) - g(n) e^{\frac{2\pi k_0}{N}n} \right] \\ &= \text{cte} \left[x(n) - x(n) * g(n) e^{\frac{2\pi k_0}{N}n} \right] \\ &= x(n) - \hat{x}_{k_0}(n) \end{aligned}$$

De la expresión anterior podemos deducir que el filtro elimina banda consiste en restar a la señal original, la señal después de aplicar el filtro pasa banda.

6.9. Filtro de Mediana

Un filtro de mediana consiste en buscar en una vecindad de tamaño $N \times M$ la mediana de un conjunto de píxeles y reemplazar el valor del píxel centrada por la mediana. Para

calcular la mediana será necesario, al igual que la convolución, hacer un barrido sobre toda la imagen y ordenar los $N \times M$ pixeles de menor a mayor y la mediana será el valor a la mitad del arreglo [Parker, 1997]. El filtro de mediana tiene la característica de preservar los bordes.

En base a esto podemos proponer el filtro de moda, en el cual a diferencia de la mediana de la muestra se calcula la moda. Otras alternativas podrían ser reemplazar la media o moda por los valores min o max de la ventana de convolución.

6.10. Filtro Binario

En el caso de tener una imagen binaria aplicar un filtro de convolución dará como resultado una imagen con números reales. En este caso la convolución la reemplazaremos por

$$y(n, m) = \bigvee_{k=-\infty}^{\infty} \bigvee_{l=-\infty}^{\infty} x(k, l) h(n - k, m - l) = x \vee h$$

donde \bigvee representa la operación or y esta operación es conocida como la or convolución o dilatación. La implementación en OpenCV se muestra a continuación.

```
void Dilacion(CvMat *ent, CvMat *sal, CvMat *h) {
    int i, j, k, l;
    int nr, nc, w, s;

    nr = ent->rows;
    nc = sal->cols;
    w = h->rows;

    for(i=0; i<nr-w; i++)
        for(j=0; j<nc-w; j++){
            s = 0;
            for(k=0; k<w; k++)
                for(l=0; l<w; l++) {
                    s += punto(ent, i+k, j+l)*punto(h, k, l);
                }
            punto(sal, i, j) = s >= 1 ? 1 : 0;
        }
}
```

Por otro lado podemos calcular la and convolución como

$$y(n, m) = \bigwedge_{k=-\infty}^{\infty} \bigwedge_{l=-\infty}^{\infty} x(k, l) h(n - k, m - l) = x \wedge h$$

La or convolución también es conocida como operación de dilación y la and convolución como contracción o erosión.

```
void Erosion(CvMat *ent, CvMat *sal, CvMat *h) {
    int i, j, k, l;
    int nr, nc, w, s;

    nr = ent->rows;
    nc = sal->cols;
    w = h->rows;

    for(i=0; i<nr-w; i++)
        for(j=0; j<nc-w; j++){
            s = 1;
            for(k=0; k<w; k++)
                for(l=0; l<w; l++) {
                    s *= punto(ent, i+k, j+l)*punto(h, k, l);
                }
            punto(sal, i, j) = s;
        }
}
```

Si hacemos la aplicación de ambos $(x \vee h) \wedge h$, tenemos primero una dilación y posteriormente una contracción, lo cual permite eliminar ruido de la imagen binaria. Se definen cerradura a la operación $(x \vee h) \wedge h$ y la apertura a la operación $(x \wedge h) \vee h$

Para obtener los bordes de la imagen podemos hacer la diferencia de la imagen original con la imagen dilatada haciendo $x - (x \vee h)$.

6.11. Filtro de Membrana

En esta sección describiremos la aplicación de la regularización al cálculo de un filtro de membrana. La regularización es una técnica de adaptable de filtrado de señales que permite estimar componentes de baja frecuencia. Abordaremos el problema desde un punto de vista estadístico, para ello, calcularemos primero el estimador de máxima verosimilitud, agregaremos información a priori y finalmente con regla de Bayes calcularemos el filtro.

6.11.1. Probabilidad de un evento

Si un experimento puede dar como resultado cualquiera N resultado diferentes igualmente probables, y si exactamente n de estos resultado corresponden al evento A , entonces la probabilidad del evento A es

$$P(A) = \frac{n}{N}$$

6.11.2. Probabilidad condicional

La probabilidad de que ocurra un evento B cuando se sabe que ha ocurrido algún otro evento A se denomina probabilidad condicional y se denota como $P(B|A)$.

Consideremos el espacio muestral $S = \{1, 2, 3, 4, 5, 6, 7\}$, donde cada uno de los eventos son igualmente probables y estos eventos se agrupan en dos conjuntos $A = \{1, 2, 3, 4\}$ y $B = \{3, 4, 5, 6, 7\}$, tal que $S = A \cup B$. De acuerdo a la definición de probabilidad de un evento, la probabilidad de A es $P(A) = 4/7$ y la probabilidad del conjunto B es $P(B) = 5/7$. En la figura podemos ver, de manera mas clara, como se agrupan estos conjuntos (ver fig 6.7).

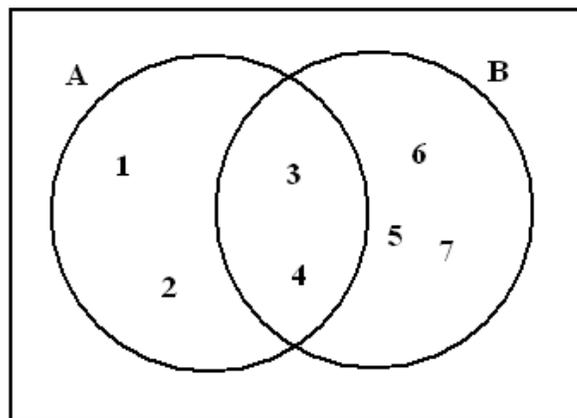


Figura 6.7: Ejemplo de dos conjuntos

Note que existe una intersección entre los conjuntos A y B , en notación de conjuntos este conjunto es $A \cap B$ y la probabilidad de estos elementos es $2/7$.

Ahora queremos determinar la probabilidad de que ocurra un elemento del conjunto A dado que solo tengo solamente elementos del conjunto B , la cual denotaremos por $P(A|B)$. Aplicando nuestra definición podemos ver que

$$P(A|B) = \frac{n\{A \cap B\}}{n\{B\}}$$

si dividimos esta ecuación entre el total de elementos del espacio muestral tenemos

$$\begin{aligned} P(A|B) &= \frac{n\{A \cap B\}/N}{n\{B\}/N} \\ &= \frac{P(A \cap B)}{P(B)} \end{aligned}$$

Para nuestros datos tenemos

$$P(A|B) = \frac{2}{5}$$

Por otro lado la probabilidad de $P(B|A) = 1/2$. Note que estos valores siguen cumpliendo nuestra definición de probabilidad muestral.

Dado lo anterior, podemos definir la probabilidad condicional de B dado A , que se denota $P(B|A)$ con

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

6.11.3. Independencia

Dos eventos A y B son independientes si y solo si $P(B|A) = P(B)$ y $P(A|B) = P(A)$, también se cumple que $P(A \cap B) = P(A)P(B)$.

6.11.4. Regla de Bayes

Si los eventos B_1, B_2, \dots, B_k constituyen una partición del espacio muestral S , donde $P(B_i) \neq 0$, entonces, para cualquier evento A de S

$$P(B_r|A) = \frac{P(B_r)P(A|B_r)}{\sum_{i=1}^k P(B_i)P(A|B_i)}$$

ver [[Walpole and Myers, 1987](#)]

6.11.5. Estimador de máxima verosimilitud.

Para encontrar el estimador de máxima verosimilitud f , dado un conjunto de observaciones g , suponemos que estos tienen una distribución de probabilidad normal con varianza $\sigma^2 = 1/2$, la cual se expresa como:

$$p(g_i|f_i) = \frac{1}{C_1} e^{-(f_i - g_i)^2}$$

considerando que los datos son generados de manera independiente tenemos que la distribución de probabilidades conjunta es

$$p(g|f) = \prod_{i=0}^{N-1} \frac{1}{C_1} e^{-(f_i - g_i)^2}$$

Note que la probabilidad conjunta es máxima cuando $g_i = f_i$. Queremos calcular una señal suave, por lo que, nuestra información a priori será

$$p(f) = \prod_{i=0}^{N-1} \frac{1}{C_2} e^{-\lambda \nabla^2 f_i} = \prod_{i=0}^{N-1} \frac{1}{C_2} e^{-\lambda (f_i - f_{i-1})^2}$$

aplicando la regla de Bayes, encontramos la expresión para la probabilidad a posteriori dada por:

$$p(f|g) = \frac{p(g|f) * p(f)}{p(g)} \tag{6.7}$$

$$p(f|g) = \prod_{i=0}^{N-1} \frac{1}{C} e^{-[(f_i - g_i)^2 + \lambda (f_i - f_{i-1})^2]}$$

$$p(f|g) = \prod_{i=0}^{N-1} \frac{1}{C} e^{-U(f)}$$

La ecuación 6.7 es la distribución de probabilidad a posteriori, el mínimo de esta función, lo encontramos cuando la función de energía $U(f)$ es mínima. Nuestro problema lo traducimos en calcular el mínimo de la siguiente función:

$$U(f) = \sum_{i=0}^{N-1} \left[(f[i] - g[i])^2 + \lambda (f[i] - f[i-1])^2 \right]$$

$$F[k] + \lambda \left(F[k] - F[k]e^{-\left(\frac{2\pi}{N}\right)k} \right) - \lambda \left(F[k]e^{\left(\frac{2\pi}{N}\right)k} - F[k] \right) = G[k]$$

agrupando términos obtenemos

$$\begin{aligned} \left[1 + 2\lambda - \lambda e^{-\left(\frac{2\pi}{N}\right)k} - \lambda e^{\left(\frac{2\pi}{N}\right)k} \right] F[k] &= G[k] \\ \left[1 + 2\lambda - 2\lambda \cos \left(\frac{2\pi}{N} k \right) \right] F[k] &= G[k] \end{aligned}$$

finalmente tenemos que

$$\begin{aligned} F[k] &= \frac{1}{\left[1 + 2\lambda - 2\lambda \cos \left(\frac{2\pi}{N} k \right) \right]} G[k] = H[k]G[k] \\ H[k] &= \frac{1}{\left[1 + 2\lambda - 2\lambda \cos \left(\frac{2\pi}{N} k \right) \right]} \end{aligned}$$

donde la función $H[k]$ es un filtro pasa bajas equivalente al filtro de membrana. Note que cuando el valor de λ aumenta tenemos que el ancho de banda se disminuye, en las figuras 6.9 y 6.10 podemos ver la respuesta a la frecuencia de este filtro para valores de λ 1 y 1000 respectivamente.

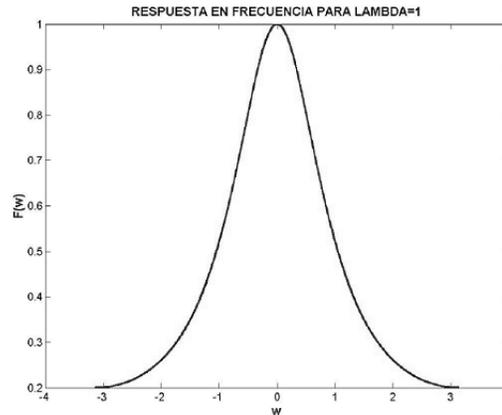


Figura 6.9: Respuesta a la frecuencia para el filtro de membrana con $\lambda = 1$

En dos dimensiones tenemos

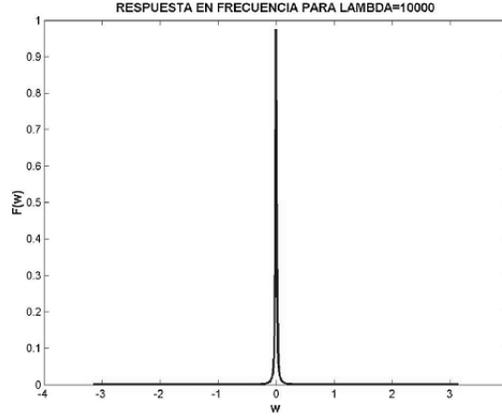


Figura 6.10: Respuesta a la frecuencia para el filtro de membrana con $\lambda = 1000$

$$U(f) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left[(f[n, m] - g[n, m])^2 + \lambda (f[n, m] - f[n-1, m])^2 + \lambda (f[n, m] - f[n, m-1])^2 \right]$$

La derivada es

$$\begin{aligned} \frac{\partial U(f)}{\partial f[n, m]} &= 2(f[n, m] - g[n, m]) + \\ &2\lambda(f[n, m] - f[n-1, m]) - 2\lambda(f[n+1, m] - f[n, m]) \\ &2\lambda(f[n, m] - f[n, m-1]) - 2\lambda(f[n, m] - f[n, m+1]) = 0 \end{aligned}$$

La Transformada Discreta de Fourier queda

$$\begin{aligned} \left[1 + 4\lambda - \lambda e^{(\frac{2\pi}{N})k} - \lambda e^{-(\frac{2\pi}{M})l} - \lambda e^{(\frac{2\pi}{M})l} - \lambda e^{-(\frac{2\pi}{N})k} \right] F[k, l] &= G[k, l] \\ \left[1 + 4\lambda - 2\lambda \cos\left(\frac{2\pi}{N}k\right) - 2\lambda \cos\left(\frac{2\pi}{M}l\right) \right] F[k, l] &= G[k, l] \end{aligned}$$

El Filtro en el dominio de Fourier queda

$$H[k, l] = \frac{1}{\left[1 + 4\lambda - 2\lambda \cos\left(\frac{2\pi}{N}k\right) - 2\lambda \cos\left(\frac{2\pi}{M}l\right) \right]}$$

Algorithm 3 Filtro de Membrana

```
1: Dadas  $g$  y  $\lambda$ 
2: Calcular  $G = TDF[g]$ 
3: for  $k = 0$  to  $N - 1$  do
4:   for  $l = 0$  to  $M - 1$  do
5:     Calcular  $F[k, l] = \frac{G[k, l]}{1 + 4\lambda - 2\lambda \cos(2\pi k/N) - 2\lambda \cos(2\pi l/M)}$ 
6:   end for
7: end for
8: Calcular  $f = TDF^{-1}[F]$ 
9: return  $f$ 
```

Filtro de Kalman

7.1. Introducción

El filtro de Kalman (KF) es un procedimiento óptimo cuando se trata de predecir eventos futuros en sistemas que continuamente están cambiando. El KF da un estado aproximado producido por definiciones recursivas, las cuales describen la estimación del estado actual como una función solo del estado anterior y de la nueva observación.

Supongamos que tenemos una variable $X(t_k)$ cuyo valor deseamos estimar en cierto tiempo t_0, t_1, t_2 , etc. También, supongamos que $X(t_k)$ satisface una ecuación dinámica lineal

$$X(t_k) = AX(t_{k-1}) + Bw_k$$

esta ecuación dinámica representa el conocimiento a priori que necesitamos acerca de la variable, por lo que A es la información a priori para predecir un nuevo estado y w_k es ruido blanco con media cero y varianza 1. Esto es $E[Bw_k] = 0$ y $E[[Bw_k][Bw_k]^T] = BB^T$.

El valor de w_k es independiente de la variable $X(t_k)$ y de su antecesor w_{k-1} , lo cual indica que las variables no están correlacionadas, características principales del ruido blanco $E[X(t_k)w_k] = 0$.

Un filtro de Kalman necesita de una estimación inicial para arrancar. Es como el motor de un automóvil que necesita de un motor de marcha y una vez que arranca no lo necesita más.

Tendremos una estimación inicial para $X(t_k)$ dada por $\bar{X}(t_0)$ y para cualquier instante de tiempo t_k la prediciremos como

$$\begin{aligned}\bar{X}(t_k) &= E[X(t_k)] \\ \bar{X}(t_k) &= E[AX(t_{k-1}) + Bw_k] \\ \bar{X}(t_k) &= AE[X(t_{k-1})] + BE[w_k]\end{aligned}$$

Finalmente obtenemos

$$\bar{X}(t_k) = A\bar{X}(t_{k-1}) \quad (7.9)$$

De manera similar, para la varianza $P(t_k)$ en el tiempo t_k , tendremos un valor inicial dado $P(t_0)$ y en cualquier tiempo la prediciremos de manera recursiva utilizando la ecuación de Riccati (7.10)

$$P(t_k) = E [[X(t_k) - \bar{X}(t_k)][X(t_k) - \bar{X}(t_k)]^T]$$

desarrollando el término $X(t_k) - \bar{X}(t_k)$ podemos encontrar una expresión equivalente

$$\begin{aligned} X(t_k) - \bar{X}(t_k) &= AX(t_{k-1}) + Bw_k - A\bar{X}(t_{k-1}) \\ X(t_k) - \bar{X}(t_k) &= A(X(t_{k-1}) - \bar{X}(t_{k-1})) + Bw_k \end{aligned}$$

$$P(t_k) = E \left[[A(X(t_{k-1}) - \bar{X}(t_{k-1})) + Bw_k] [A(X(t_{k-1}) - \bar{X}(t_{k-1})) + Bw_k]^T \right]$$

$$\begin{aligned} P(t_k) &= E [A(X(t_{k-1}) - \bar{X}(t_{k-1}))(X(t_{k-1}) - \bar{X}(t_{k-1}))^T A^T] + \\ &E [A(X(t_{k-1}) - \bar{X}(t_{k-1}))[Bw_k]^T + Bw_k[A(X(t_{k-1}) - \bar{X}(t_{k-1}))]^T] + \\ &E [Bw_k w_k^T B^T] \end{aligned}$$

$$P(t_k) = AE [(X(t_{k-1}) - \bar{X}(t_{k-1}))(X(t_{k-1}) - \bar{X}(t_{k-1}))^T] A^T + BE [w_k w_k^T] B^T$$

$$P(t_k) = AP(t_{k-1})A^T + BB^T \quad (7.10)$$

Asumamos que hacemos una observación a la que llamaremos Y y consideraremos que esta esta relacionada linealmente con la variable X

$$Y(t_k) = MX(t_k) + Rw_k \quad (7.11)$$

el valor esperado de $Y(t_k)$ es

$$\begin{aligned} \bar{Y}(t_k) &= E [Y(t_k)] \\ \bar{Y}(t_k) &= E [MX(t_k) + Rw_k] \end{aligned}$$

$$\bar{Y}(t_k) = ME [X(t_k)] + RE [w_k]$$

$$\bar{Y}(t_k) = M\bar{X}(t_k) \quad (7.12)$$

Nuestra mejor estimación no será $\bar{X}(t_k)$ ya que esta estimación esta calculada a partir de modelo a priori y no tiene información alguna de la observación. Así que $\bar{X}^+(t_k)$ será una mejor estimación y esta dada por la ecuación

$$\begin{aligned} \bar{X}^+(t_k) &= \bar{X}(t_k) + K_k (Y(t_k) - \bar{Y}(t_k)) \\ \bar{X}^+(t_k) &= \bar{X}(t_k) + K_k (Y(t_k) - M\bar{X}(t_k)) \end{aligned} \quad (7.13)$$

donde K_k es un número que llamaremos ganancia de Kalman.

Note que $Y(t_k) - \bar{Y}(t_k)$, en la ecuación 7.13 es exactamente el error al estimar $Y(t_k)$, el cual en parte se debe al ruido w_k y en parte a la predicción o estimación de la señal (modelo a priori), por lo que poner un valor unitario de K_k sería un error. Entonces ¿Que valor debemos usar?. Antes de decidir calculemos la varianza de $X(t_k)$ respecto a nuestra nueva estimación $\bar{X}^+(t_k)$

$$P^+(t_k) = E \left[[X(t_k) - \bar{X}^+(t_k)] [X(t_k) - \bar{X}^+(t_k)]^T \right]$$

Para encontrar una expresión equivalente tenemos que desarrollar $X(t_k) - \bar{X}^+(t_k)$

$$X(t_k) - \bar{X}^+(t_k) = X(t_k) - \bar{X}(t_k) - K_k(Y(t_k) - M\bar{X}(t_k))$$

$$X(t_k) - \bar{X}^+(t_k) = X(t_k) - \bar{X}(t_k) - K_k(MX(t_k) + Rw_k - M\bar{X}(t_k))$$

$$P^+(t_k) = E \left[[(1 - MK_k)(X(t_k) - \bar{X}(t_k)) + K_kRw_k] [(1 - MK_k)(X(t_k) - \bar{X}(t_k)) + K_kRw_k]^T \right]$$

los productos cruzados son cero ya que el ruido no esta correlacionada con la señal. Tenemos la formula final para la varianza dada por

$$\begin{aligned} P^+(t_k) &= (1 - MK_k) E \left[[X(t_k) - \bar{X}(t_k)] [X(t_k) - \bar{X}(t_k)]^T \right] (1 - MK_k)^T \\ &\quad + (K_kR) E [w_k w_k^T] (K_kR)^T \end{aligned}$$

Finalmente la formula para calcular la varianza es:

$$P^+(t_k) = (1 - MK_k) P(t_k) (1 - MK_k)^T + (K_k R) (K_k R)^T \quad (7.14)$$

En la formula anterior podemos ver que la varianza del error $P^+(t_k)$, la cual queremos minimizar, es función de la ganancia de Kalman. Así que calculemos una nueva ganancia K_k^* que minimice $P^+(t_k)$

$$\frac{\partial P^+(t_k)}{\partial K_k} = 2(1 - MK_k) P(t_k) (-M^T) + 2RR^T K_k = 0$$

despejando tenemos

$$K_k^* = MP(t_k) [MP(t_k) M^T + RR^T]^{-1}$$

Los pasos que seguiremos para implementar el filtro de Kalman son

Paso 1.

Daremos un modelo a priori de acuerdo con las características de la señal. Este modelo puede ser un proceso auto regresivo de primer orden dado por

$$\bar{X}(t_k) = A\bar{X}(t_{k-1})$$

Paso 2.

Utilizamos la ecuación recursiva de Riccati para hacer una estimación de la varianza del error en términos de la varianza anterior

$$P(t_k) = AP(t_{k-1})A^T + BB^T$$

Paso 3.

Calculamos el valor de la ganancia de Kalman para minimizar la varianza en la estimación.

$$K_k^* = MP(t_k) [MP(t_k) M^T + RR^T]^{-1}$$

Paso 4.

Haremos una corrección en la estimación con el valor K_k^*

$$X^*(t_k) = \bar{X}(t_k) + [K_k^*]^T (Y(t_k) - M\bar{X}(t_k))$$

Paso 5.

Hacemos la estimación final de la varianza

$$P^+(t_k) = P(t_k) - K_k^* M P(t_k)$$

Estos cinco pasos se repetirán mientras exista una señal de entrada, considerando únicamente que nuestra nueva estimación de la varianza es $P(t_k) = P^*(t_k)$. Los dos primeros pasos corresponderán a la predicción de la señal y los siguientes a realizar la asimilación de la información, pesada con la ganancia de Kalman.

En la figura 7.11 podemos observar un diagrama de bloque que describe el funcionamiento del filtro de Kalman.

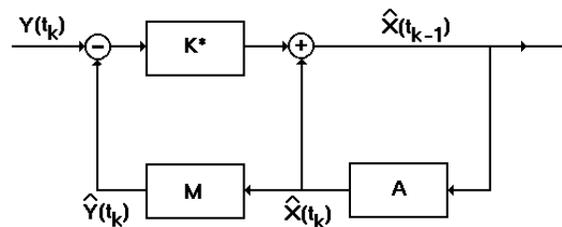


Figura 7.11: Diagrama de aplicación del Filtro de Kalman

7.1.1. Ejemplo 1. Filtrado de una señal uni-dimensional

Consideremos que deseamos filtrar una señal unidimensional dada como $Y(t_k) = 20 + 0.1 \times t_k$ y ruido aditivo de observación el cual está dado como $N(0, 7)$.

Deseamos diseñar un filtro de Kalman que nos permita filtrar la señal. Necesitamos de un modelo a priori, es evidente que desconozco la fórmula para $Y(k)$ pero se que se trata de una señal sin variaciones bruscas, así que el modelo que propondré será

$$\begin{aligned} X(t_k) &= AX(t_{k-1}) + Bw_k \\ Y(t_k) &= MX(t_k) + Rw_k \end{aligned}$$

el valor esperado para esta es

$$\begin{aligned}\bar{X}(t_k) &= A\bar{X}(t_{k-1}) \\ \bar{Y}(t_k) &= M\bar{X}(t_k)\end{aligned}$$

con $A = 1$, $M = 1$, $B^2 = 100$ y $R^2 = 1000$ y condiciones iniciales $\bar{X}(t_0) = 30$ y $P(t_0) = 500$.

Para ver la aplicación de los pasos del filtro de Kalman, calcularé la primer iteración

Paso I.

$$\bar{X}(t_1) = A\bar{X}(t_0) = 1 * 30 = 30$$

Paso II.

$$P(t_1) = A^2P(t_0) + B^2 = 1 * 500 + 100 = 600$$

Paso III.

$$K_1^* = \frac{MP(t_1)}{M^2P(t_1) + R^2} = \frac{1 * 600}{1 * 600 + 1000} = 0.375$$

Paso IV.

$$\begin{aligned}X^*(t_1) &= \bar{X}(t_1) + K_1^*(Y(t_1) - M\bar{X}(t_1)) \\ &= 30 + 0.375(20.931141 - 1 * 30) \\ &= 26.599177875\end{aligned}$$

Paso V

$$\begin{aligned}P^*(t_1) &= (1 - MK_1^*)^2 P(t_1) + (K_1^*R)^2 \\ &= (1 - 1 * 0.375)^2 600 + (0.375)^2 * 1000 \\ &= 375\end{aligned}$$

En la figura 7.12 podemos ver el resultado del filtrado de la señal $Y(k)$. En esta gráfica se muestra la señal $Y(k)$ con ruido, la señal sin el ruido y la señal $X(k)$ filtrada. Es importante hacer notar que si consideramos un valor de varianza B^2 pequeño, en el modelo a priori tendríamos una señal plana dado que el modelo dinámico es $\bar{X}(t_k) = \bar{X}(t_{k-1})$.

Finalmente es importante ver el comportamiento de la varianza de la señal que estamos prediciendo, la cual converge a un valor estable. En la figura 7.13 podemos ver su evolución, para los datos de este ejemplo en particular.

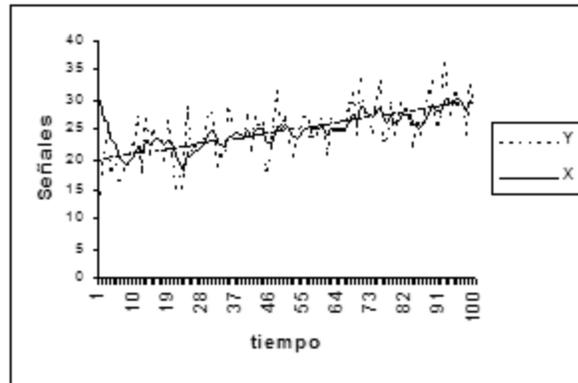


Figura 7.12: Ejemplo de filtrado utilizando KF

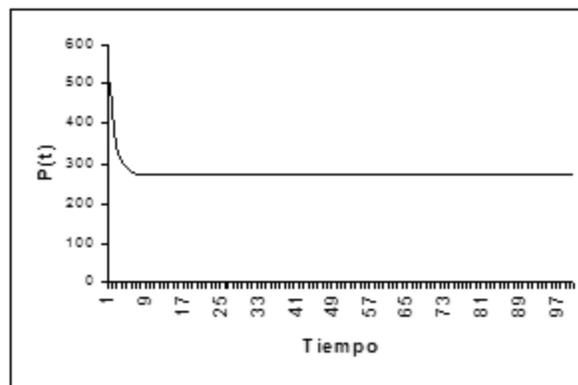


Figura 7.13: Comportamiento de la varianza

7.1.2. Ejemplo 2. Seguimiento de un objeto con aceleración constante

Trataré de explicar la aplicación del filtro de Kalman para realizar el Tracking en tiempo real de un objeto en movimiento. El objeto puede ser, por ejemplo, un avión en pleno vuelo el cual se mueve con aceleración constante.

Definamos un vector

$$X(n) = [X_1(n), X_2(n), X_3(n)]^T \quad (7.15)$$

que representa la posición del objeto en un espacio Cartesiano tridimensional, en el tiempo

nT , donde T es el periodo de muestreo.

Consideremos que $\dot{X}(n)$ es el vector de velocidad y $\ddot{X}(n)$ es el vector de aceleración. Por definición, el vector de velocidad y el vector de aceleración son la primera y segunda derivada respectivamente de la trayectoria de un objeto. Asumimos que la frecuencia de muestreo es lo suficientemente grande para justificar la expresión para el vector de posición del objeto a tiempo $(n+1)T$ como

$$x_i(n+T) = x_i(n) + T \dot{x}_i(n) + \frac{1}{2}T^2 \ddot{x}_i(n) + \eta_p(n)$$

donde $\eta_p(n)$ es un vector de error de truncamiento. Similarmente podemos expresar el vector de velocidad como

$$\dot{x}_i(n+T) = \dot{x}_i(n) + T \ddot{X}(n) + \eta_s(n)$$

donde $\eta_s(n)$ es un vector de error de truncamiento. Similarmente podemos expresar el vector de aceleración como

$$\ddot{x}_i(n+T) = \ddot{x}_i(n) + \eta_a(n)$$

donde $\eta_a(n)$ es un vector de error de truncamiento y asumiremos en general que el periodo de muestreo es $T = 1$

El vector de estado para el objeto lo definimos como

$$X(n) = [x_1(n), \dot{x}_1(n), \ddot{x}_1(n), x_2(n), \dot{x}_2(n), \ddot{x}_2(n), x_3(n), \dot{x}_3(n), \ddot{x}_3(n)]^T$$

el cual podemos escribir de manera simplificada como

$$X(n) = \begin{bmatrix} X_1(n) \\ X_2(n) \\ X_3(n) \end{bmatrix} \quad (7.16)$$

donde

$$X_i(n) = \begin{bmatrix} x_i(n) \\ \dot{x}_i(n) \\ \ddot{x}_i(n) \end{bmatrix} = \begin{bmatrix} x_i(n) \\ v_i(n) \\ a_i(n) \end{bmatrix}$$

De igual forma el vector de error de truncamiento lo definimos como

$$\eta_i(n) = [\eta_{p1}(n), \eta_{s1}(n), \eta_{a1}(n), \eta_{p2}(n), \eta_{s2}(n), \eta_{a2}(n), \eta_{p3}(n), \eta_{s3}(n), \eta_{a3}(n)]^T$$

o de manera simple como

$$\eta_i(n) = [\eta_{pi}(n), \eta_{si}(n), \eta_{ai}(n)]^T \quad (7.17)$$

Podemos usar las ecuaciones 7.15, 7.16, y 7.17 para expresar la ecuación de estado que modela el tracking del objeto

$$\begin{bmatrix} X_1(n+1) \\ X_2(n+1) \\ X_3(n+1) \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} \begin{bmatrix} X_1(n) \\ X_2(n) \\ X_3(n) \end{bmatrix} + \begin{bmatrix} \eta_1(n) \\ \eta_2(n) \\ \eta_3(n) \end{bmatrix}$$

donde

$$A = \begin{bmatrix} 1 & 1 & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

La matriz de transición de estado, es una matriz diagonal en bloque, así que podemos simplificarla como

$$X_i(n+1) = AX_i(n) + \eta_i(n)$$

El vector de observación $Y(n)$ es un vector tridimensional que podemos representar como

$$Y(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{bmatrix} + w(n)$$

donde $w(n)$ es el vector de error de medida. Podemos deducir que la matriz de medidas del modelo de tracking es una matriz dispersa de orden 3×9

$$Y(n) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(n) \\ v_1(n) \\ a_1(n) \\ x_2(n) \\ v_2(n) \\ a_2(n) \\ x_3(n) \\ v_3(n) \\ a_3(n) \end{bmatrix} + w(n)$$

Podemos simplificar la expresión escribiendo

$$Y(n) = \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{bmatrix} \begin{bmatrix} X_1(n) \\ X_2(n) \\ X_3(n) \end{bmatrix} + w(n)$$

donde $M = [1, 0, 0]$. Note que la matriz de medida es una matriz diagonal en bloque, la cual podemos simplificar y escribir

$$Y_i(n) = MX_i(n) + w_i(n)$$

Finalmente nuestro modelo para hacer el tracking esta dado por:

$$\begin{aligned} X_i(n+1) &= AX_i(n) + \eta_i(n) \\ Y_i(n) &= MX_i(n) + w_i(n) \end{aligned}$$

adicionalmente asumimos que

$$\begin{aligned} E [\eta_i(n)\eta_i(n)^T] &= B \\ E [w_i(n)w_i(n)^T] &= R \end{aligned}$$

Solución Numérica

Consideremos el caso de un objeto que se mueve al rededor de una elipse en dos dimensiones dada en forma paramétrica por

$$Y(n) = \begin{bmatrix} y_1(n) \\ y_2(n) \end{bmatrix} = \begin{bmatrix} 200 + 50\cos(n/10) \\ 200 + 40\text{seno}(n/10) \end{bmatrix}$$

El modelo apriori para el vector de parámetros es:

$$\bar{X}(n) = \begin{bmatrix} \bar{x}_1(n) \\ \bar{v}_1(n) \\ \bar{a}_1(n) \\ \bar{x}_2(n) \\ \bar{v}_2(n) \\ \bar{a}_2(n) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0.5 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_1(n-1) \\ \bar{v}_1(n-1) \\ \bar{a}_1(n-1) \\ \bar{x}_2(n-1) \\ \bar{v}_2(n-1) \\ \bar{a}_2(n-1) \end{bmatrix}$$

Nuestra predicción esta dada por

$$\bar{Y}(n) = M\bar{X}(n)$$

$$\bar{Y}(n) = \begin{bmatrix} \bar{x}_1(n) \\ \bar{x}_2(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1(n) \\ \bar{v}_1(n) \\ \bar{a}_1(n) \\ \bar{x}_2(n) \\ \bar{v}_2(n) \\ \bar{a}_2(n) \end{bmatrix}$$

Como valores iniciales consideraremos

$$\bar{X}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$P(0) = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}$$

Los parámetros adicionales los son:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

La implementación en Mathematica

```
Kalman[Y_, Xg_, P_] := Block[{Xn, Xs, Pn, Ps, K, Aux},
  (* Paso 1 *)
  Xn = A . Xg;
  (* Paso 2 *)
  Pn = A . P. Transpose[A] + B . Transpose [B];
  (* Paso 3 *)
  Aux = M. Pn. Transpose [M ] + R. Transpose[R];
  K = (Pn. Transpose[M]).Inverse[Aux];
  (* Paso 4*)
  Xs = Xn + K.(Y - M. Xn);
  (* Paso 5 *)
  Ps = Pn - K. M. Pn;
  {Xs, Ps}
]
```

Para ejecutar podemos hacer y en la figura 7.14 se muestra el resultado de la implementación

```
Table[{X[[k + 1]], P} = Kalman[Y[[k]], X[[k]], P],
  {k, Length[Y] - 1}];
```

7.1.3. Ejemplo 3. Determinación de parámetros.

Consideremos un objeto en movimiento el cual queremos rastrear. Lo único que sabemos de el es que se mueve en un espacio bidimensional en línea recta.



Figura 7.14: Simulación de un movimiento en dos dimensiones sobre una elipse

Comenzaremos por definir el vector de estados, con los parámetros que describen a una línea recta $X(k) = [x_1(k), x_2(k)]^T$ y representaremos las coordenadas de un punto con un vector $[d_x, d_y]$.

Nuestro modelo dinámico será

$$X(k) = AX(k-1) + Bw_k \quad \text{con} \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

El vector de observaciones lo determinamos como

$$d_y(k) = x_1(k)d_x(k) + x_2(k)$$

en forma matricial

$$d_y(k) = M(k)X(k) + Rw_k \quad \text{con} \quad M = [d_x(k), 1]$$

Nuestra predicción estará dada por

$$\begin{aligned} \bar{X}(k) &= \bar{X}(k-1) \\ \bar{d}_y(k) &= M(k)\bar{X}(k) \\ P(k) &= AP(k-1)A^T + BB^T \end{aligned}$$

Con el propósito de probar estas ecuaciones, realice una simulación donde

$$d_y(k) = 1.1d_x(k) + 2.0$$

los parámetros utilizados son

$$\bar{X}(0) = [0, 0]^T, \quad B = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{bmatrix}, \quad P(0) = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad \text{y } R = 16$$

En la figura 7.15 vemos la línea recta que deseamos modelar así como la línea que estamos prediciendo. En la figura 7.16 podemos ver la convergencia de los parámetros a medida que tenemos más muestras.

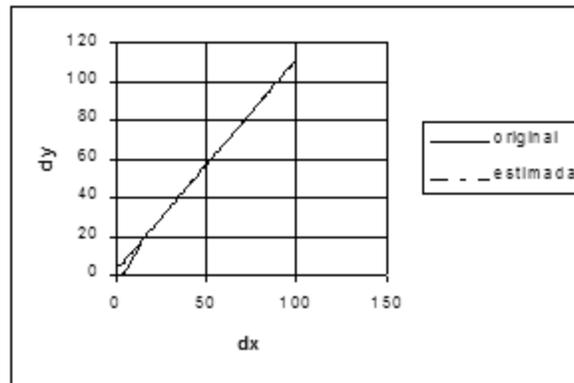


Figura 7.15: Solución del cálculo de parámetros

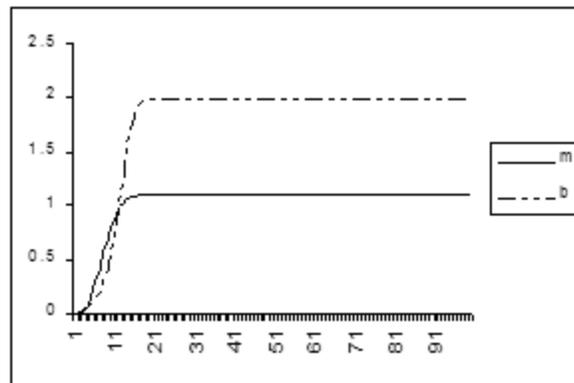


Figura 7.16: Parámetros calculados

En este ejemplo, se tiene una relación lineal entre los parámetros y la observación. De no existir esta linealidad tendríamos que hacer uso de la serie de Taylor para encontrar una expresión lineal, lo que da lugar, al filtro de Kalman extendido.

Aplicaciones

8.1. Detección de corrimientos utilizando técnicas de restauración

Dadas dos imágenes f' y f , la delta de dirac δ y el operador de corrimiento S_d podemos decir:

$$f' = S_d f = S_d f * \delta = f * S_d \delta = f * \delta_d$$

lo cual indica que si realizamos convolución con un impulso desplazado, tendremos una imagen desplazada a un nuevo punto de coordenadas dado por d . Aplicando transformada de Fourier podemos intentar calcular el valor de desplazamiento haciendo

$$\begin{aligned} f'(n) &= f * \delta_d(n) \\ F'(k) &= F(k) e^{j(\frac{2\pi}{N})kd} \end{aligned}$$

de esta expresión podemos despejar el valor de d pero si un valor de $F(k) = 0$, no tenemos manera de estimar el valor del mismo.

En su lugar utilizaremos técnicas de Regularización, para ello, minimizamos la función

$$\begin{aligned} &|f' - f * g|^2 \\ &\text{sugeto a } |l * g|^2 \end{aligned}$$

donde $f'(n)$ es la imagen desplazada, $f(n)$ es la imagen origen, $l(n)$ es un operador lineal que escogeremos de la mejor manera para facilitar la solución y $g(n)$ es la función impulso que intentamos encontrar. Para nuestro calculo procedemos

$$\begin{aligned}
U(g) &= \sum_{n=0}^{N-1} [f'(n) - f * g(n)]^2 + \beta \sum_{n=0}^{N-1} |l * g(n)|^2 \\
&= \sum_{n=0}^{N-1} \left[f'(n) - \sum_{i=0}^{N-1} f(n-i)g(i) \right]^2 + \beta \sum_{n=0}^{N-1} \left[\sum_{i=0}^{N-1} l(n-i)g(i) \right]^2
\end{aligned}$$

Para encontrar el valor de g que minimiza esta expresión hacemos

$$\begin{aligned}
\frac{\partial U(g)}{\partial g(i)} &= -2 \sum_{n=0}^{N-1} \left[f'(n) - \sum_{i=0}^{N-1} f(n-i) * g(i) \right] f(n-i) \\
&\quad + 2\beta \sum_{n=0}^{N-1} \left[\sum_{i=0}^{N-1} l(n-i) * g(i) \right] l(n-i) \\
&= 0
\end{aligned}$$

reorganizado términos tenemos

$$\begin{aligned}
\sum_{n=0}^{N-1} \left[\sum_{i=0}^{N-1} f(n-i) * g(i) \right] f(n-i) + \beta \sum_{n=0}^{N-1} \left[\sum_{i=0}^{N-1} l(n-i) * g(i) \right] l(n-i) &= \sum_{n=0}^{N-1} f'(n) f(n-i) \\
&= [f * f^r + \beta l * l^r] * g(n) = f' * f^r(n)
\end{aligned}$$

Al aplicar transformada de Fourier tenemos:

$$\begin{aligned}
G(k) &= \frac{[F'(k)]^* F(k)}{[F(k)]^* F(k) + \beta [L(k)]^* L(k)} \\
G(k) &= \frac{[F'(k)]^* F(k)}{|F(k)|^2 + \beta |L(k)|^2}
\end{aligned}$$

al sacar la antitrasformada de Fourier calculamos la función impulso $g(n)$.

8.1.1. Ejemplos

Se utilizaron como entrada la imagen f mostrada en 8.17(a), y la imagen f' de la figura 8.17(b). Al parámetro β se le dio un valor de 0.1.

Después de calcular $g(n)$ al computar la TFD^{-1} de $G(k)$, se localizó una magnitud máxima igual a 1926.3965, en las coordenadas (2,179), lo que significa que el segmento de la imagen de f' las coordenadas cuya columna es mayor que 179 no es visible desde f , y que los renglones de f' menores que 2 no son visibles desde f . Una vez ensamblada la imagen panorámica con la información anterior, el resultado se muestra en la Fig. 8.18.



Figura 8.17: Imágenes de entrada para el sistema.



Figura 8.18: Imagen panorámica que se formó con la imagen f y f' .

Pudiera parecer que el ensamble no fue correcto, puesto que las líneas rectas no parecen ajustar perfectamente. Sin embargo, este problema se debe a tres factores principales. El

primero, es que la cámara que se utilizó cuenta con una distorsión considerable, que se puede observar en la curvatura de las líneas que se suponen rectas. En segundo lugar, al tomar la fotografía pudo haber existido una transformación más allá de una traslación horizontal. Finalmente, la iluminación de la escena captada por la cámara puede ser diferente para cada fotografía lo cual perjudicaría el comportamiento del algoritmo A pesar de estos tres factores, se encuentra una aproximación bastante aceptable que puede servir incluso como proveedor de valores iniciales de corrección proyectiva. En el caso ideal, en donde de estos factores no se presentan, el resultado es también ideal. Para ejemplificar esta situación, podemos observar las imágenes iniciales en la Fig. 8.1.1, y el resultado se observa en la Fig. 8.20. En este caso, el empalme es perfecto puesto que no existe otra variante más que traslación, ni cambio de iluminación entre las imágenes, ni distorsión de la cámara. De hecho, este ejemplo se realizó fragmentando una imagen en tres más pequeñas con diferente traslación.

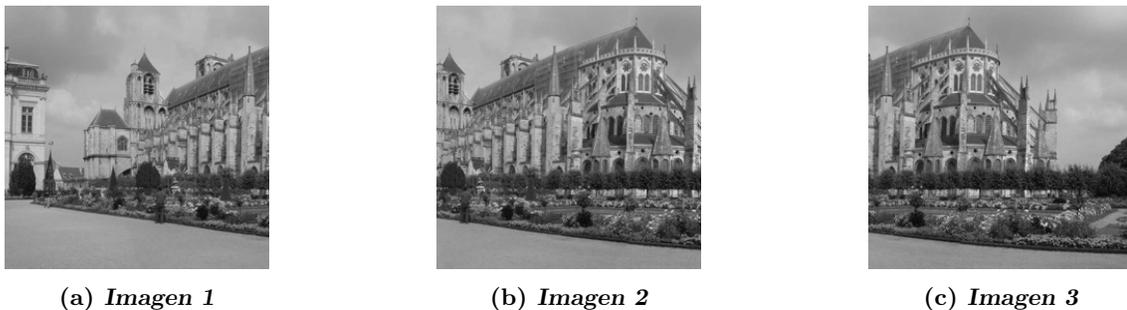


Figura 8.19: Imágenes de entrada para la creación de una panorámica final.



Figura 8.20: Imagen panorámica que se formó con la imagen 1, 2 y 3.

En la Fig. 8.24 se ilustran los resultados obtenidos al aplicar este método. En las figuras 8.21 y 8.22 se muestran 2 fotografías de la misma escena con un desplazamiento horizontal y vertical. En 8.23 puede observarse un máximo cuyas coordenadas (x, y) representan el corrimiento horizontal y vertical respectivamente.

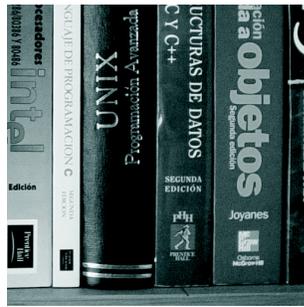


Figura 8.21: Imagen 1



Figura 8.22: Imagen 2

8.2. Filtrado de una señal de audio

En este documento se presentan ejercicios para comprobar lo aprendido acerca de los filtros pasa banda utilizando un archivo de sonido en donde se ejecutan varios instrumentos simultáneamente. Se realizaron ejercicios que consistieron en separar los sonidos de instrumentos diferentes sintonizando el filtro en diferentes posiciones del espectro de frecuencias. Los resultados que se obtuvieron fueron satisfactorios.

8.2.1. Filtro Pasa Banda

Suponga que se ha calculado la Transformada de Fourier Discreta (TFD) de una señal y se quiere filtrar para analizar lo que se tiene en un rango de frecuencias determinado. Como base, tomaremos el filtro Gaussiano que se utilizó en el filtro pasa bajas, el cual se define como,

$$g(x, \mu, \sigma) = \frac{1}{Z} \exp \left(-\frac{(x-\mu)^2}{2\sigma^2} \right) \quad (8.18)$$

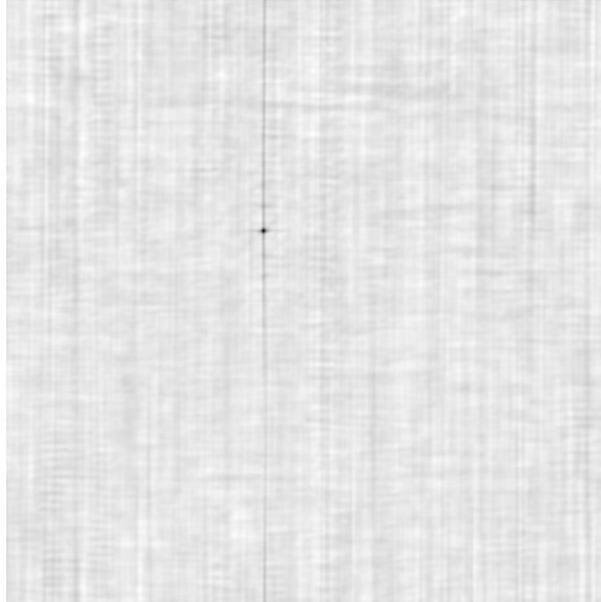


Figura 8.23: Corrimiento calculado



Figura 8.24: Empate de las imágenes 8.21 y 8.22

donde definiremos $Z = \sum_{i=1}^N \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$, cuyo fin es el de normalizar el área de la señal a

1.

La TFD de una señal Gaussiana es también una función Gaussiana,

$$g(\mu, \sigma_t) = G(\mu, \sigma_w) \quad (8.19)$$

Si deseamos filtrar todas las frecuencias fuera de un rango determinado, basta con colocar dos señales Gaussianas¹ $G(\mu, \sigma_w)$ en $k - \mu$ y $k + \mu$, donde k es la media de la banda de frecuencias que se desea preservar.

Por el teorema del desplazamiento tenemos que,

$$\frac{1}{2}G(k - \mu) \xleftrightarrow{\mathcal{F}} \frac{1}{2}g(n) \exp^{j(\frac{2\pi}{N}\mu n)} \quad (8.20)$$

$$\frac{1}{2}G(k + \mu) \xleftrightarrow{\mathcal{F}} \frac{1}{2}g(n) \exp^{-j(\frac{2\pi}{N}\mu n)} \quad (8.21)$$

Como se definió anteriormente, en la frecuencia, el filtro es la suma de $G(k - \mu)$ y $G(k + \mu)$ multiplicado por el espectro de frecuencia de la señal. Por lo tanto, el filtro pasa banda en el tiempo corresponde a la convolución de la señal con el filtro dado por,

$$\begin{aligned} f_1(n) &= \frac{1}{2}g(n) \cos(\frac{2\pi}{N}\mu n) + j \sin(\frac{2\pi}{N}\mu n) \\ f_2(n) &= \frac{1}{2}g(n) \cos(\frac{2\pi}{N}\mu n) - j \sin(\frac{2\pi}{N}\mu n) \\ f_1(n) + f_2(n) &= g(n) \cos(\frac{2\pi}{N}\mu n) \end{aligned} \quad (8.22)$$

8.2.2. Localización de Frecuencias Reales en el Dominio Discreto de Fourier

Si conociéramos la banda que se desea filtrar tenemos que encontrar en dónde se localiza esa banda en nuestra señal discreta. Las frecuencias en el continuo, que denotaremos como f , se miden en ciclos/seg (Hertz), sin embargo, cuando una señal se discretiza, existen

¹Ambas señales cuentan con la misma velocidad angular, y por lo tanto, con frecuencias iguales.

frecuencias en el rango de $[0, 2\pi]$ en N muestras (ver Fig. 8.25). La velocidad angular de cualquier señal discreta es,

$$w = \frac{2\pi}{N}k \text{ radianes/seg} \quad (8.23)$$

cuyo valor máximo se alcanza cuando $k = N$, donde w se hace igual a 2π . Por esta razón, la frecuencia de cualquier señal discreta es $f_D = \frac{2\pi}{2\pi} = 1$.

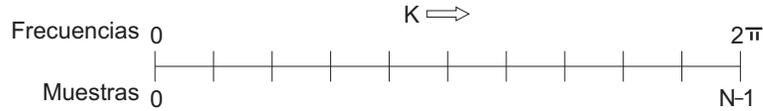


Figura 8.25: Frecuencia de una señal discreta.

Si tenemos una frecuencia de muestreo,

$$f_m = N_m \text{ muestras/seg} \quad (8.24)$$

por lo tanto tenemos un periodo $T_m = \frac{1}{N_m} \text{ seg/muestras}$. Esto quiere decir que cada $\frac{1}{N_m}$ segundos obtenemos una muestra de nuestra señal,

$$\text{muestra} = \frac{1}{N_m} \text{ seg} \quad (8.25)$$

El problema de encontrar la k en donde se represente una frecuencia dada en Hertz, se resuelve sustituyendo la ec. (8.25) en (8.23), con lo cual tenemos que,

$$w = \frac{2\pi}{N}k \text{ radianes/} \left(\frac{1}{N_m} \text{ seg} \right)$$

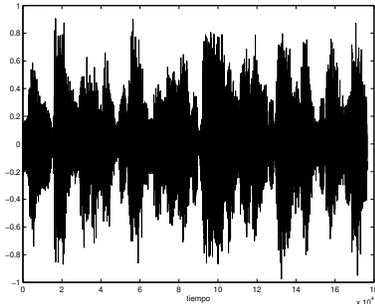
$$w = 2\pi \left(\frac{N_m}{N} \right) k \text{ radianes/seg} \quad (8.26)$$

Observando la ecuación anterior, se puede observar entonces que $\frac{f}{f_m} = \frac{k}{N}$, por lo tanto, podemos obtener el valor de k para cualquier frecuencia en el continuo dado en Hertz como,

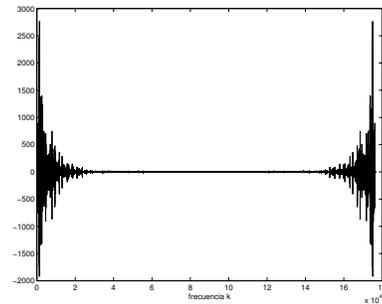
$$k = N \frac{f}{f_m} \quad (8.27)$$

8.2.3. Ejercicios

Dado un archivo de sonido, vamos a tratar de filtrar un determinado rango de frecuencias. El archivo de sonido que se va a utilizar es una fragmento de 2 *seg.* que fue muestreado a 4100 KHz. Es un archivo de música que contiene diversos instrumentos tocando de manera simultanea, entre ellos un piano que ejecuta notas graves, un acordeón que ejecuta una melodía mucho más aguda, entre otros.



(a) Señal de sonido en el tiempo



(b) Señal de sonido en la frecuencia

Figura 8.26: Señal de sonido de prueba utilizado.

Filtrado del Sonido del Piano

Como primer ejercicio, se trató de sintonizar el filtro de manera que se pueda escuchar con mayor claridad la ejecución del piano. De forma experimental se observó que los sonidos del piano en el archivo de prueba se localizan al rededor de 350 Hz. Por lo tanto, aplicando la ec. (8.27), se obtuvo un valor de $k = 1400$, y se utilizó una señal Gaussiana con $\sigma = 60$. La Figura 8.2.3 muestra el kernel en el tiempo (Fig. 8.27(a)) y en el dominio de la frecuencia (8.27(b)). En esta última figura, se puede observar claramente que el kernel en la frecuencia contiene dos Gaussianas localizadas en k y $-k$.

Al multiplicar la TFD de la señal de sonido con la TFD del kernel, se obtiene la señal que se muestra en la Fig. 8.28. Los resultados fueron satisfactorios y se logró separar efectivamente el sonido del piano.

Filtrado del Sonido del Acordeón

Como segundo ejercicio, se trató de separar el sonido del acordeón de nuestro sonido de prueba, cuyas frecuencias se encontraron de manera experimental al rededor de 600 Hz.



Figura 8.27: Kernel de convolución con $\sigma = 60$ en el tiempo.

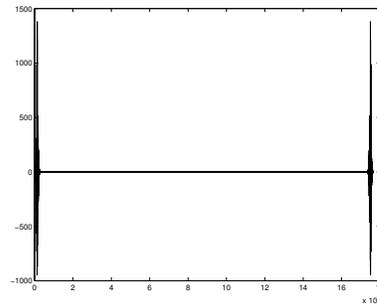


Figura 8.28: Señal de sonido filtrada.

Aplicando la ec. (8.27) se obtuvo un valor de $k = 2400$. Se utilizó un kernel con $\sigma = 60$ que se puede observar en la Fig. 8.2.3, en el tiempo y en la frecuencia. Se puede observar que efectivamente, la TFD del kernel está formada por dos Gaussianas localizadas en k y $-k$.

La señal filtrada se muestra en la Fig. 8.30, la cual logró separar satisfactoriamente el sonido del acordeón, aunque estuvo mayormente contaminada por otros sonidos como algunos violines que comparten parte del espectro con este instrumento.

8.2.4. Código Fuente de los ejercicios

Para los dos ejercicios se utilizó el mismo código, basta con modificar convenientemente el valor de ciertas variables, específicamente sigma y $k0$. El programa fue implementado en matlab 6.5 Rel. 13.

```
clear all;
%se abre el archivo de sonido
%en Y se almacena la se\UNICODE{0xf1}al
```

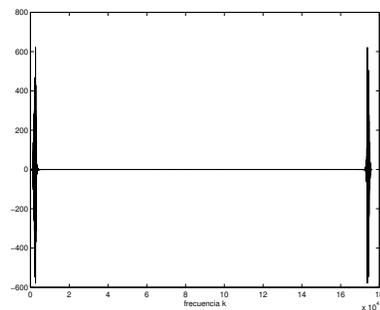
Figura 8.29: Kernel de convolución con $\sigma = 60$ en el tiempo.

Figura 8.30: Señal de sonido filtrada.

```

%en FS se almacena la frecuencia de muestreo
%en NBITS se almacena la cantidad de bits por segundo
[Y,FS,NBITS]=wavread('yannshort.wav'); N=length(Y); serie=[0:N-1];
%se imprime la se\UNICODE{0xf1}al de sonido en el tiempo
f=figure; set(f,'Name','Se\UNICODE{0xf1}al de sonido en el tiempo(original)') plot(serie,Y);
%KERNEL Gaussiano
%creamos el kernel Gaussiano
sigma=60;
%tamkernel=41; %al menos sigma*6+1
tamkernel=6*sigma+1;
kernel=zeros(1,tamkernel); x=-(tamkernel-1)/2;
for i=1:tamkernel,
kernel(i) = exp(-x^2 / (2*sigma^2)); x=x+1; end
%normalizamos el kernel
kernel=kernel./sum(kernel);
%multiplicamos por el coseno para el filtro pasa bandas
%piano400
%acordeon 600
k0=N*600/FS; sc=[-(tamkernel-1)/2:(tamkernel-1)/2];
c=cos(2*pi/N*sc*k0); kernel=kernel.*c;
%lo pasamos a un vector del tama\UNICODE{0xf1}o N
%para hacer la convolucion en la frecuencia
kernelF = zeros(N,1); kernelF(1:(tamkernel+1)/2) =
kernel((tamkernel+1)/2:tamkernel);
kernelF((N-(tamkernel-1)/2)+1:N)=kernel(1:(tamkernel-1)/2);
%se imprime el kernel en el tiempo
serieK = [-(tamkernel-1)/2:(tamkernel-1)/2]; f=figure;
set(f,'Name','Kernel en el tiempo') plot(serieK,kernel,'b');
%se obtiene la TFD del sonido
as=fft(Y);
%se obtiene la TFD del kernel
ak=fft(kernelF);
%se imprime el sonido en la frecuencia
f=figure; set(f,'Name','Se\UNICODE{0xf1}al de sonido en la
frecuencia')

```

```
plot(serie,real(as),'b', serie, imag(as),'r');
%se imprime el kernel en la frecuencia
f=figure; set(f,'Name','Kernel en la frecuencia')
plot(serie,real(ak),'b', serie, imag(ak),'r');
%se hace el filtrado con la multiplicacion
%de la TFD del sonido con la TFD del kernel
filtrada = as.*ak;
%se imprime la se\UNICODE{0xf1}al filtrada
f=figure; set(f,'Name','Se\UNICODE{0xf1}al de sonido en la
frecuencia
(filtrada)') plot(serie,real(filtrada),'b', serie,
imag(filtrada),'r');
%se obtiene la TFD inversa del sonido
final = ifft(filtrada);
%se graba en un archivo de salida
wavwrite(final,FS,NBITS,'salida_gauss.wav');
```

`!TEX root = ../senales.tex`

Algoritmo EM

9.1. Estimación de máxima verosimilitud.

Deseamos encontrar para un conjunto de observaciones x_i , los parámetros de la distribución que las genero. Considerando el caso de una distribución normal, los parámetros θ a calcular serán la media μ y la varianza σ^2

$$g(x_i, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x_i - \mu)^2}{2\sigma^2}\right) \quad (9.28)$$

La probabilidad conjunta de las observaciones, asumiendo independencia, dados los parámetros de la distribución se pueden calcular como

$$P(x|\mu, \sigma) = \prod_{i=1}^N g(x_i, \mu, \sigma)$$

y aplicando regla de Bayes podemos calcular la verosimilitud como

$$P(\mu, \sigma|x) = \frac{P(x|\mu, \sigma)P(\mu)P(\sigma)}{P(x)} = P(x|\mu, \sigma)$$

considerando $P(\mu)$, $P(\sigma)$ y $P(x)$ constantes

La logverosimilitud para las N observaciones es:

$$\begin{aligned}
 P(\mu, \sigma | x) &= \prod_{i=1}^N g(x_i, \mu, \sigma) \\
 \mathcal{L}(\mu, \sigma) &= \ln P(\mu, \sigma | x) = \sum_{i=1}^N \left[-\ln \sqrt{2\pi} - \ln \sigma - \frac{(x_i - \mu)^2}{2\sigma^2} \right]
 \end{aligned} \tag{9.29}$$

El estimador θ^* de máxima verosimilitud lo obtenemos calculando $\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = 0$

$$\begin{aligned}
 \frac{\partial \mathcal{L}(\mu, \sigma)}{\partial \mu} &= \sum_{i=1}^N \left[\frac{x_i - \mu}{\sigma^2} \right] = 0 \\
 \mu^* &= \frac{\sum_{i=1}^N x_i}{N}
 \end{aligned} \tag{9.30}$$

$$\begin{aligned}
 \frac{\partial \mathcal{L}(\mu, \sigma)}{\partial \sigma} &= \sum_{i=1}^N \left[-\frac{1}{\sigma} + \frac{(x_i - \mu)^2}{2\sigma^3} \right] = 0 \\
 (\sigma^2)^* &= \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}
 \end{aligned} \tag{9.31}$$

9.2. El Algoritmo EM.

El término EM fue introducido por Dempster en 1977 para designar un algoritmo muy general para la estimación de máxima verosimilitud en conjuntos de datos incompletos. El nombre se refiere a los pasos que constituyen el algoritmo: un paso E de cálculo de la esperanza, seguido por otro paso M, de maximización de la verosimilitud. Los pasos a seguir para el algoritmo EM son:

- 1.- Se asigna un valor inicial a los parámetros por algún método razonable.
- 2.- Los valores que faltan son reemplazados por sus valores esperados, condicionados a los valores observados y a los valores corrientes de los parámetros.
- 3.- Los parámetros son reestimados usando los valores observados y modificando los valores faltantes.

4.- Los valores perdidos son recalculados asumiendo que los nuevos parámetros son correctos.

5.- Los parámetros son recalculados una vez más en el paso tres y en el paso cuatro hasta cumplir cierto criterio de paro.

9.3. Aplicación del EM a mezclas de distribuciones.

El problema cuando tratamos con mezclas de distribuciones es que uno no sabe que distribución en la mezcla fue la que produjo cada ejemplo de observación. Si tuviéramos esa información disponible, solo resta calcular el estimador de máxima verosimilitud para una distribución en particular. Utilizaremos una variable ε , para etiquetar la i -ésima observación generada por la k -ésima distribución así:

$$\begin{aligned}\varepsilon_{ik} &= 1 \text{ si } x_i \sim g(x_i, \theta_k) \\ \varepsilon_{ik} &= 0 \text{ para los demás}\end{aligned}\tag{9.32}$$

Nuestros datos completos son $X = (x_i, \varepsilon_i)$ donde x_i esta dado y ε_i lo desconocemos. La verosimilitud para los datos completos es $P(\varepsilon, \theta|x) = P(x|\varepsilon, \theta) * P(\varepsilon|\theta) * P(\theta)/P(x)$ donde cada término lo definidos como

$$P(x|\varepsilon, \theta) = \prod_{i=1}^N g(x_i, \theta_k) = \prod_{i=1}^N \prod_{k=1}^C g(x_i, \theta_k)^{\varepsilon_{ik}}\tag{9.33}$$

$$P(\varepsilon|\theta) = \prod_{k=1}^C \alpha_k = \prod_{i=1}^N \prod_{k=1}^C \alpha_k^{\varepsilon_{ik}}\tag{9.34}$$

y $P(\theta)$, $P(x)$ como constantes.

La logverosimilitud $\$(\varepsilon, \theta) = \ln P(\varepsilon, \theta|x)$ queda definida

$$\begin{aligned}\$(\varepsilon, \theta) &= \ln \prod_{i=1}^N \prod_{k=1}^C [\alpha_k g(x_i, \theta_k)]^{\varepsilon_{ik}} \\ &= \sum_{i=1}^N \sum_{k=1}^C \varepsilon_{ik} [\ln \alpha_k + \ln g(x_i, \theta_k)]\end{aligned}\tag{9.35}$$

Tendremos que determinar dadas las observaciones x_i los parámetros θ_k de las C distribuciones que conforman la mezcla, así como el valor de α_k que indica la proporción de observaciones que pertenecen a la k – esima distribución $g(x_i, \theta_k)$. Como desconocemos el valor ε_{ik} utilizaremos el valor esperado $\bar{\varepsilon}_{ik} = E[\varepsilon_{ik}|x_i, \theta]$ (Paso E del algoritmo EM). Analizando los valores de la variable ε_{ik} (ε_i es un vector de $C - 1$ de ceros y únicamente un 1) por lo que el valor esperado $\bar{\varepsilon}_{ik}$ es la probabilidad marginal a posteriori de que $\varepsilon_{ik} = 1$ dadas x_i y θ . Utilizando la regla de Bayes podemos encontrar que:

$$\begin{aligned}\bar{\varepsilon}_{ik}^t &= P(\varepsilon_{ik} = 1|x_i, \theta)^t = \frac{P(x_i|\varepsilon_{ik} = 1, \theta)^t P(\varepsilon_{ik} = 1|\theta)^t}{P(x_i|\theta)^t} \\ \bar{\varepsilon}_{ik}^t &= P(\varepsilon_{ik} = 1|x_i, \theta)^t = \frac{g(x_i, \theta_k^t) * \alpha_k^t}{\sum_{k=1}^C g(x_i, \theta_k^t) * \alpha_k^t}\end{aligned}\quad (9.36)$$

Para calcular (9.36) tenemos que maximizar la log-verosimilitud de las observaciones sujeta a la restricción de que $\sum_k \alpha_k = 1$. Aplicando multiplicadores de Lagrange tenemos:

$$l_{obs} = \sum_{i=1}^N \sum_{k=1}^C \bar{\varepsilon}_{ik} [\ln \alpha_k + \ln g(x_i, \theta_k)] - \lambda \left[\sum_{k=1}^C \alpha_k - 1 \right] \quad (9.37)$$

Para resolver calculamos $\frac{\partial l_{obs}}{\partial \alpha_k} = 0$ para la ecuación (9.37)

$$\frac{\partial l_{obs}}{\partial \alpha_k} = \sum_{i=1}^N \frac{\bar{\varepsilon}_{ik}}{\alpha_k} - \lambda = 0 \quad (9.38)$$

Reorganizando términos y sumando sobre las C distribuciones tendremos

$$\begin{aligned}\lambda \sum_{k=1}^C \alpha_k &= \sum_{i=1}^N \sum_{k=1}^C \bar{\varepsilon}_{ik} \\ \lambda &= N\end{aligned}\quad (9.39)$$

Sustituyendo en la ecuación (9.38) el valor de λ

$$\alpha_k = \frac{\sum_{i=1}^N \bar{\varepsilon}_{ik}}{N} \quad (9.40)$$

9.3.1. Mezclas finitas de distribuciones normales.

Para el desarrollo del algoritmo, será necesario conocer a priori el tipo de distribuciones que forman la mezcla. Un caso sencillo resulta ser cuando utilizamos una distribución normal. El vector de parámetros θ_k es (μ_k, σ_k) y la distribución está dada por:

$$g(x_i, \mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(\frac{-(x_i - \mu_k)^2}{2\sigma_k^2}\right) \quad (9.41)$$

Dada la función de distribución la log-verosimilitud es:

$$\mathcal{L}(\mu, \sigma) = \sum_{i=1}^N \sum_{k=1}^C \bar{\varepsilon}_{ik} \left[\ln \alpha_k - \ln \sqrt{2\pi} - \ln \sigma - \frac{(x_i - \mu)^2}{2\sigma^2} \right] \quad (9.42)$$

Maximizamos $\mathcal{L}(\mu, \sigma)$ con respecto al vector de parámetros (μ, σ)

$$\frac{\partial E}{\partial \mu_k} = \sum_{i=1}^N \bar{\varepsilon}_{ik} \frac{(x_i - \mu_k)}{\sigma_k^2} = 0 \quad (9.43)$$

Resolviendo para μ_k y recordando que es el valor a tiempo $t + 1$.

$$(\mu_k^*)^{t+1} = \frac{\sum_{i=1}^N \bar{\varepsilon}_{ik}^t x_i}{\sum_{i=1}^N \bar{\varepsilon}_{ik}^t} \quad (9.44)$$

Hacemos lo mismo para el caso de σ_k .

$$\frac{\partial E}{\partial \sigma_k} = \sum_{i=1}^N \bar{\varepsilon}_{ik} \left(-\frac{1}{\sigma_k} + \frac{(x_i - \mu_k)^2}{\sigma_k^3} \right) = 0 \quad (9.45)$$

$$(\sigma_k^{*2})^{t+1} = \frac{\sum_{i=1}^N \bar{\varepsilon}_{ik}^t (x_i - \mu_k)^2}{\sum_{i=1}^N \bar{\varepsilon}_{ik}^t} \quad (9.46)$$

9.4. Imágenes con tonos de gris constantes a pedazos.

Para este tipo de imágenes queremos determinar el valor del tono de gris que caracteriza a una region R_k tal que $I(r) = \mu_k + n_r \forall r \in R_k$. A priori sabemos que los tonos de gris son constantes a pedazos y que existe un ruido de observación n_r que es normal $N(0, \sigma^2)$. La verosimilitud de los datos completos esta dada por

$$P(x, \varepsilon | \theta) = P(x | \varepsilon, \theta) * P(\varepsilon | \theta) \quad (9.47)$$

El término $P(x | \varepsilon, \theta)$ es la probabilidad de una observación x dados ε, θ . Consideraremos que la distribución es gaussiana y que el tono de gris a calcular tiene media μ_k y varianza $\sigma_k^2 = 1$.

$$P(x | \varepsilon, \theta) = \prod_{i=1}^N \prod_{k=1}^C g(x_i, \mu_k, 1)^{\varepsilon_{ik}} \quad (9.48)$$

El modelo más popular, usado para restringir un campo a ser constante a pedazos, es el modelo generalizado de Ising, cuyos potenciales están dados por.

$$\begin{aligned} V(\varepsilon_i, \varepsilon_j) &= -1 \text{ si } \varepsilon_i = \varepsilon_j \\ V(\varepsilon_i, \varepsilon_j) &= +1 \text{ si } \varepsilon_i \neq \varepsilon_j \end{aligned} \quad (9.49)$$

La probabilidad conjunta $P(\varepsilon | \theta)$ no podrá ser representada como el producto de las probabilidades marginales, ya que el imponer una restricción de coherencia espacial hace que se pierda la independencia. Supondremos una distribución de probabilidades de Gibbs para $P(\varepsilon | \theta)$ (campo aleatorio Markoviano) dada por.

$$P(\varepsilon | \theta) = e^{-\beta \sum_{\langle i,j \rangle} V(\varepsilon_i, \varepsilon_j)} \quad (9.50)$$

Las expresiones finales para la verosimilitud y la logverosimilitud son:

$$P(x, \varepsilon | \theta) = \prod_{i=1}^N \prod_{k=1}^C g(x_i, \mu_k, 1)^{\varepsilon_{ik}} \times e^{-\beta \sum_{\langle i,j \rangle} V(\varepsilon_{ik}, \varepsilon_{jk})} \quad (9.51)$$

$$\mathcal{L}(\varepsilon, \mu) = - \sum_{i=1}^N \sum_{k=1}^C \frac{(x_i - \mu_k)^2}{2} \varepsilon_{ik} - \beta \sum_{\langle i,j \rangle} V(\varepsilon_{ik}, \varepsilon_{jk}) \quad (9.52)$$

Minimizar la expresión anterior resulta complicada, para ello se utilizan algoritmos como Simulated Annealing, el muestreador Gibbsiano, el algoritmo de Montecarlo, y de manera mas eficaz campos de medidas.

Tareas

10.1. Tarea

Considere que $x(n)$ es una señal con parte par $\mathcal{P}(x(n))$ y parte impar $\mathcal{I}(x(n))$, demostrar que

$$\sum_{n=-\infty}^{\infty} x^2(n) = \sum_{n=-\infty}^{\infty} \mathcal{P}(x(n))^2 + \sum_{n=-\infty}^{\infty} \mathcal{I}(x(n))^2$$

10.2. Tarea

Determinar si cada una de las siguientes señales discretas es periódica o no lo es. Si la señal es periódica, determine su periodo fundamental

1. $x[n] = \sin(\frac{6\pi}{7}n + 1)$
2. $x[n] = \cos(\frac{n}{8} - \pi)$
3. $x[3 - n]$
4. $x[n] = \cos(\frac{\pi}{2}n) \cos(\frac{\pi}{4}n)$
5. $x[n] = 2 \cos(\frac{\pi}{4}n) + \sin(\frac{\pi}{8}n) - 2 \cos(\frac{\pi}{2}n + \frac{\pi}{6})$

Determine para cada uno de los siguientes sistemas discreto cuales que propiedades cumplen y cuales no. Sin memoria, Invariante en el tiempo, Lineal, Causal, Estable. Ofrezca argumentos que justifiquen su respuesta, En cada ejemplo $y[n]$ denota la salida y $x[n]$ la entrada del sistema.

1. $y[n] = x[-n]$
2. $y[n] = x[n - 2] - 2x[n - 8]$

$$3. y[n] = nx[n]$$

$$4. y[n] = \mathcal{I}\{x[n]\}$$

$$5. y[n] = \begin{cases} x[n] & n \geq 1 \\ 0 & n = 0 \\ x[n+1] & n \leq -1 \end{cases}$$

$$6. y[n] = \begin{cases} x[n] & n \geq 1 \\ 0 & n = 0 \\ x[n] & n \leq -1 \end{cases}$$

$$7. y[n] = x[4n+1]$$

Hacer la implementación de las funciones de Correlación y Convolución, mostrar un ejemplo de que funciones.

Bibliografía

- [Jahne, 2002] Jahne, B. (2002). *Digital Image Processing*. Springer, New York, N.Y.
- [Oppenheim, 1997] Oppenheim, A. V. (1997). *Señales y Sistemas*. Pearson Education, MÈxico.
- [Parker, 1997] Parker, J. R. (1997). *Algorithms for Image Processing and Computer Vision*. Wiley, USA.
- [Press et al., 2012] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2012). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, USA.
- [Walpole and Myers, 1987] Walpole, R. E. and Myers, R. H. (1987). *Probabilidad y Estadística para Ingenieros*. Interamericana, MÈxico D.F.