

PROCESAMIENTO DIGITAL DE SEÑALES

BIBLIOGRAFÍA

1. Oppenheim, A.V., and R.W. Schafer. Discrete-Time Signal Processing. Englewood Cliffs, NJ: Prentice Hall, 1989.
2. Parks, T.W., and C.S. Burrus. Digital Filter Design. New York: John Wiley & Sons, 1987.
3. Proakis, J.G., and D.G. Manolakis. Digital Signal Processing: Principles, Algorithms, and Applications. Englewood Cliffs, NJ: Prentice Hall, 1996.
4. Steven W. Smith, Ph.D. The Scientist and Engineer's Guide to Digital Signal Processing
5. Lonnie c. Ludeman. Fundamentals of digital signal processing.
6. Ricardo A. Losada. Digital Filters with MATLAB

MÓDULO 1: TRANSFORMADA DE FOURIER

INTRODUCCIÓN

El Procesamiento digital de Señales usa como único tipo de datos las señales. Estas señales en la mayoría de los casos se originan de sensores: Vibraciones sísmicas, imágenes, sonidos, etc. DSP da las herramientas matemáticas, algoritmos, y técnicas para manipular estas señales después de que han sido convertidos a forma digital. Esto incluye: mejoramiento de imágenes, reconocimiento y generación de voz, compresión de datos para almacenaje y transmisión, etc.

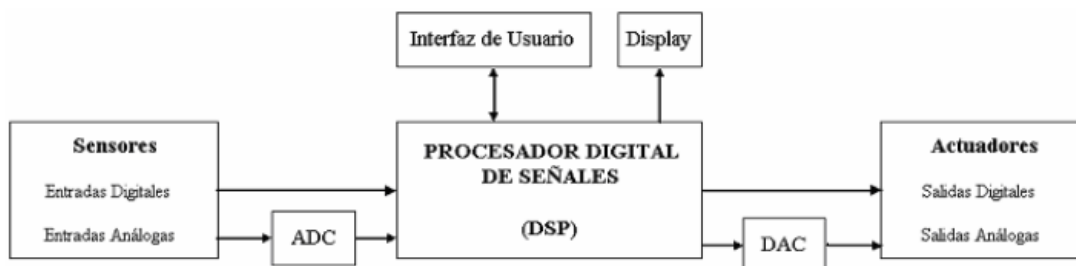
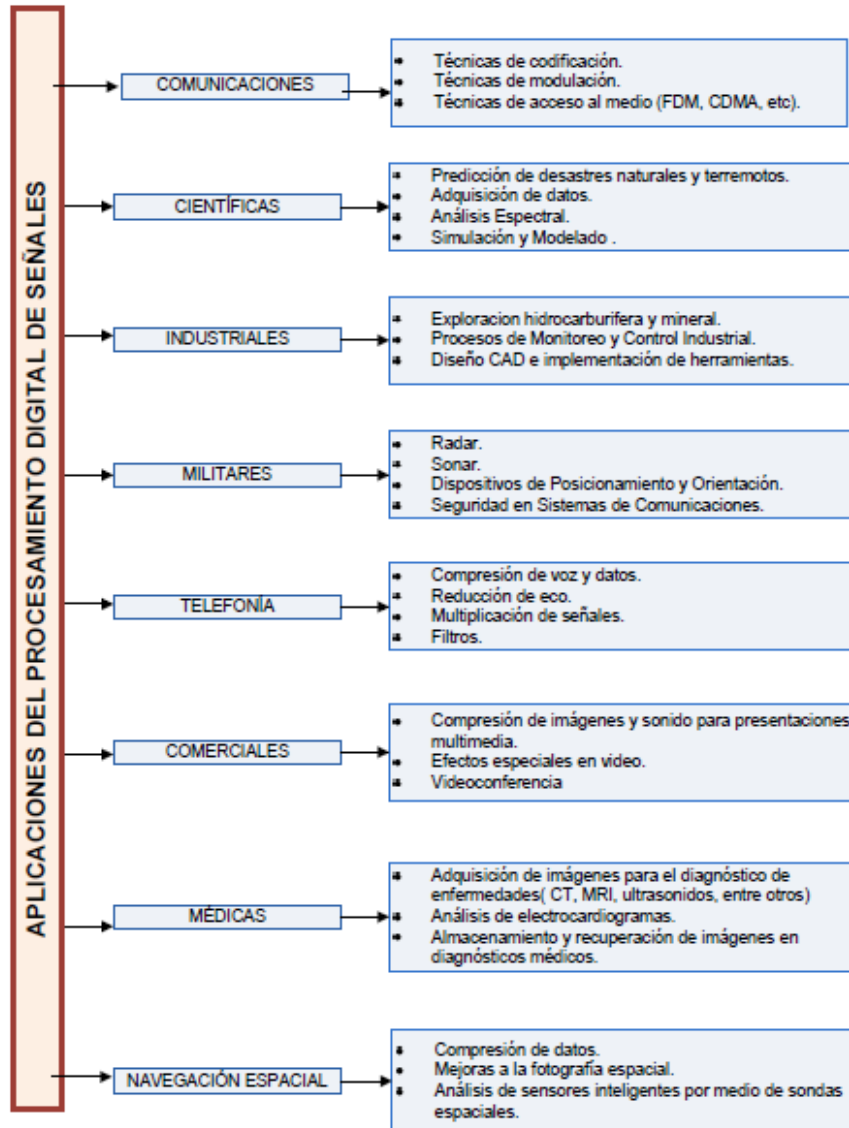
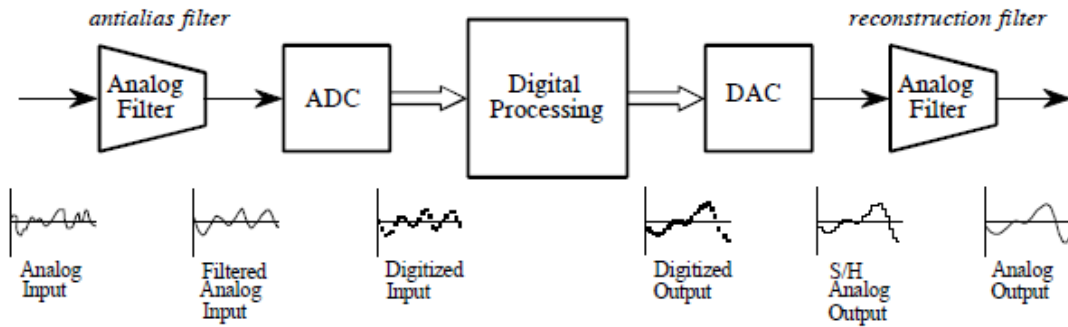


Diagrama de bloques conceptual de un sistema típico
ADC : Conversor Análogo Digital. – DSP : Digital Signal Processor.
DAC : Conversor Digital Análogo.





ESTADÍSTICA

La estadística es la ciencia utilizada en el PDS para interpretar los datos numéricos.

Media (μ)

Es el valor promedio de una señal.

N: Número total de muestras de una señal. Ejemplo N= 512 muestras

$$\mu = \frac{1}{N} \sum_{i=0}^{N-1} x_i$$

Desviación estándar (σ)

Es el promedio de las desviaciones de las señales en función de su potencia.

$$\sigma^2 = \frac{1}{N} \sum_{i=0}^N (x_i - \mu)^2$$

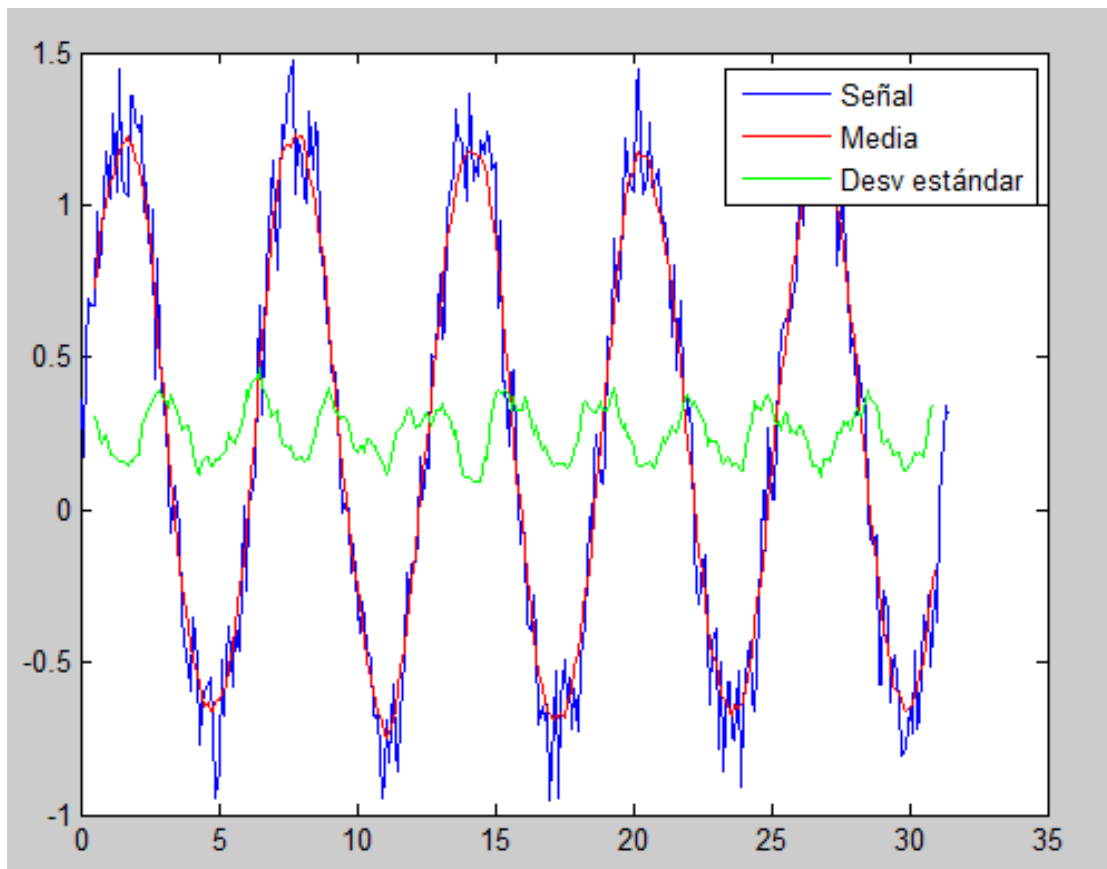
EJEMPLO:

```
clear all
t=0:0.1:10*pi;
noise=0.5*(rand(1,length(t)));
x=sin(t)+noise;      %Señal ruido
a=10;               %tamaño de la ventana
```

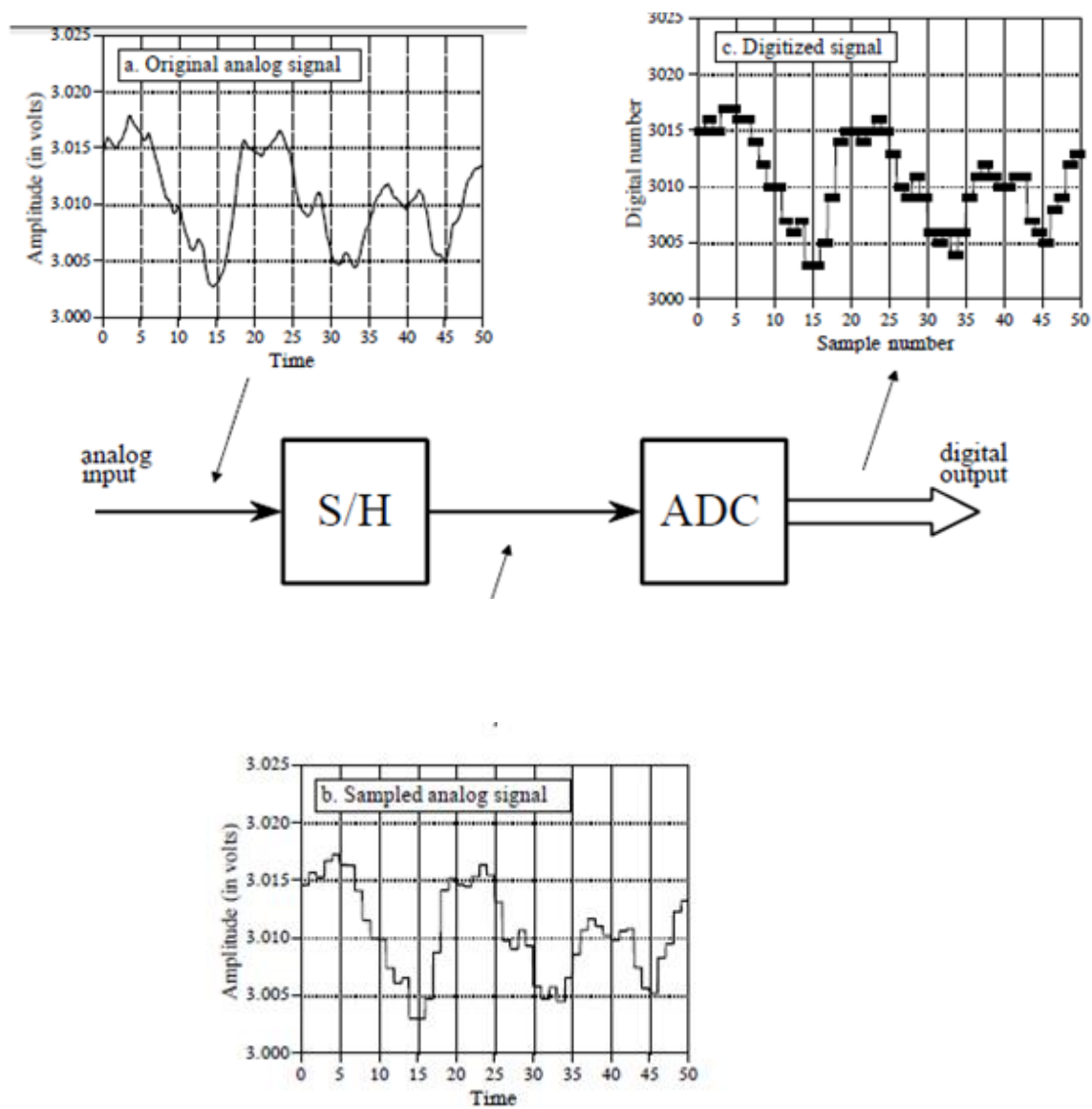
```

mx=zeros(1,length(t)-a);
%media de cada ventana
for n=6:length(t)-a/2
    mx(n-a/2)=mean(x(n-a/2:n+a/2));
    dx(n-a/2)=std(x(n-a/2:n+a/2));
end
mt=t(a/2+1:length(t)-a/2);
plot(t,x,mt,mx,'r',mt,dx,'g')
%hold on
%plot(mt,mx,'r')
legend('Señal','Media','Desv estándar')

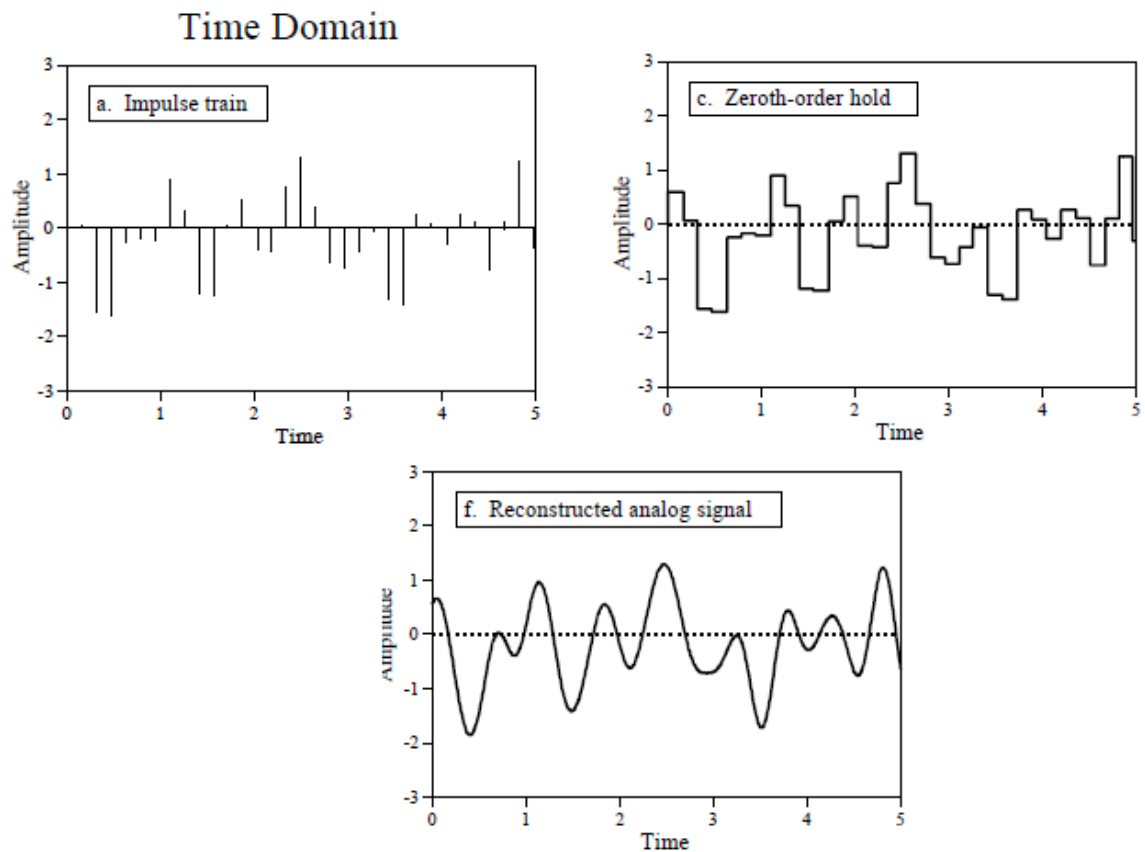
```



CONVERSIÓN: ANÁLOGO – DIGITAL



CONVERSIÓN: DIGITAL – ANÁLOGO



TRANSFORMADA DE FOURIER DISCRETA - DFT

En DSP se utiliza la DFT (Transformada de Fourier Discreta) porque es la que se puede procesar en un microcontrolador por ser discreta y finita.

Type of Transform	Example Signal
Fourier Transform <i>signals that are continious and aperiodic</i>	
Fourier Series <i>signals that are continious and periodic</i>	

Discrete Time Fourier Transform
signals that are discrete and aperiodic

Discrete Fourier Transform
signals that are discrete and periodic



COMPONENTES DE UNA SEÑAL: FUNCIONES BÁSICAS

Para resolver muchos problemas de ingeniería es conveniente descomponer una señal dada en suma de funciones básicas. Una forma razonable es tomar la señal de entrada y representarla con la funciones básicas con exponenciales complejas o señales senoidales. Para señales discretas se usan la Transformada de Fourier Discreta DFT y un cálculo eficiente de ella es la transformada Rápida de Fourier FFT.

Para una señal discreta periódicas:

a) Representación en funciones básicas de *forma exponencial*,

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

si $W_N = e^{-j \frac{2\pi}{N}}$, entonces,

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

$$X(k) = DFT[x(n)], \quad y \quad x(n) = IDFT[X(k)]$$

$$x(n) = \sum_{k=0}^{N/2} \text{Re}X(k) \cos\left(k \frac{2\pi}{N} n\right) + \sum_{k=0}^{N/2} \text{Im}X(k) \text{sen}\left(k \frac{2\pi}{N} n\right)$$

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

$$\text{Im}X(k) = - \sum_{n=0}^{N-1} x(n) \text{sen}\left(k \frac{2\pi}{N} n\right)$$

a) Representación en funciones básicas de *forma trigonométrica*

Para N par:

$$x(n) = A(0) + \sum_{k=1}^{\frac{N}{2}-1} A(k) \cos\left(k \frac{2\pi}{N} n\right) + \sum_{k=1}^{\frac{N}{2}-1} B(k) \text{sen}\left(k \frac{2\pi}{N} n\right) + A\left(\frac{N}{2}\right) \cos(\pi n)$$

$$\text{Donde } A(0) = \frac{1}{N} \sum_{n=0}^{N-1} x(n); \quad A(k) = \frac{2}{N} \sum_{n=0}^{N-1} x(n) \cos\left(k \frac{2\pi}{N} n\right)$$

$$B(k) = \frac{2}{N} \sum_{n=0}^{N-1} x(n) \text{sen}\left(k \frac{2\pi}{N} n\right); \quad A\left(\frac{N}{2}\right) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cos(\pi n)$$

Para N Impar:

$$x(n) = A(0) + \sum_{k=1}^{\frac{N-1}{2}} A(k) \cos\left(k \frac{2\pi}{N} n\right) + \sum_{k=1}^{\frac{N-1}{2}} B(k) \text{sen}\left(k \frac{2\pi}{N} n\right)$$

Los coeficientes son iguales a N par con la excepción de que:

$$A\left(\frac{N}{2}\right) = 0$$

Relaciones entre la forma trigonométrica y la forma exponencial de $x(n)$:

$$A(0) = \frac{X(0)}{N}$$

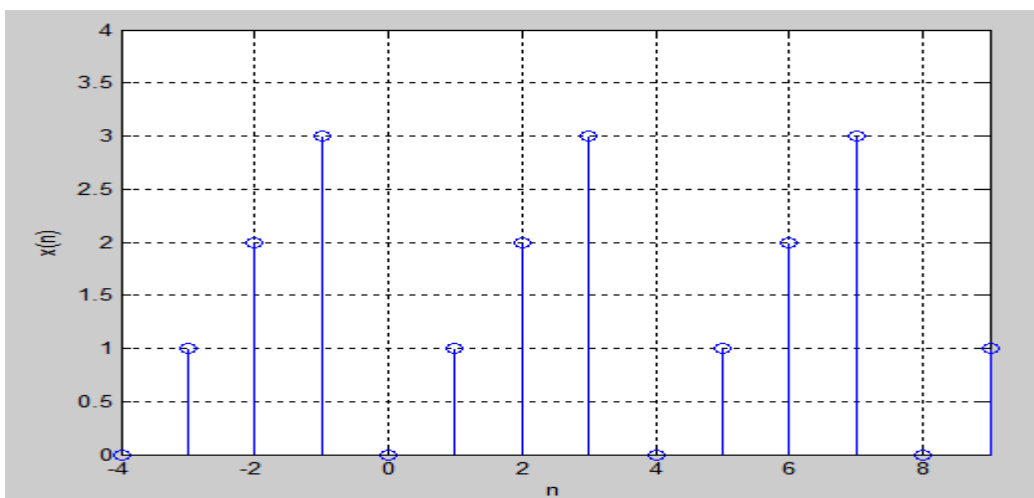
$$A(k) = \frac{[X(k) + X(N - k)]}{N}$$

$$B(k) = \frac{j[X(k) - X(N - k)]}{N}$$

$$A\left(\frac{N}{2}\right) = \frac{X\left(\frac{N}{2}\right)}{N}$$

EJEMPLO 1:

Encontrar la forma exponencial y trigonométrica de la señal discreta $x(n)$:



a) Forma exponencial

Periodo $N = 4$

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

$$\text{donde } W_N = e^{-j\frac{2\pi}{N}}, \quad W_N^k = e^{-j\frac{2\pi}{N}k}$$

$$W_4^1 = e^{-j\frac{2\pi}{4}1} = e^{-j\frac{\pi}{2}} = \cos\frac{\pi}{2} - j\sin\frac{\pi}{2} = -j$$

$$W_4^2 = W_4^1 W_4^1 = (-j)(-j) = -1$$

$$W_4^3 = W_4^2 W_4^1 = (-1)(-j) = j$$

$$W_4^4 = W_4^3 W_4^1 = (j)(-j) = 1$$

$$X(0) = \sum_{n=0}^{4-1} x(n)W_4^{0n} = x(0) + x(1) + x(2) + x(3) = 0 + 1 + 2 + 3 = 6$$

$$\begin{aligned} X(1) &= \sum_{n=0}^{4-1} x(n)W_4^{1n} = x(0)W_4^0 + x(1)W_4^1 + x(2)W_4^2 + x(3)W_4^3 \\ &= 0(1) + 1(-j) + 2(-1) + 3(j) = -2 + 2j \end{aligned}$$

$$\begin{aligned} X(2) &= \sum_{n=0}^{4-1} x(n)W_4^{2n} = x(0)W_4^0 + x(1)W_4^2 + x(2)W_4^4 + x(3)W_4^6 \\ &= 0(1) + 1(-1) + 2(1) + 3(-1) = -2 \end{aligned}$$

$$\begin{aligned} X(3) &= \sum_{n=0}^{4-1} x(n)W_4^{3n} = x(0)W_4^0 + x(1)W_4^3 + x(2)W_4^6 + x(3)W_4^9 \\ &= 0(1) + 1(j) + 2(-1) + 3(-j) = -2 - 2j \end{aligned}$$

$$X(k) = [6, -2 + 2j, -2, -2 - 2j]$$

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

b) Forma trigonométrica

$$A(0) = \frac{X(0)}{N} = \frac{3}{2} = 1.5$$

$$A(1) = \frac{[X(1) + X(4-1)]}{4} = -1$$

$$A(2) = A\left(\frac{N}{2}\right) = \frac{X\left(\frac{4}{2}\right)}{4} = -\frac{1}{2} = -0.5$$

$$B(1) = \frac{j[X(1) - X(4-1)]}{4} = -1$$

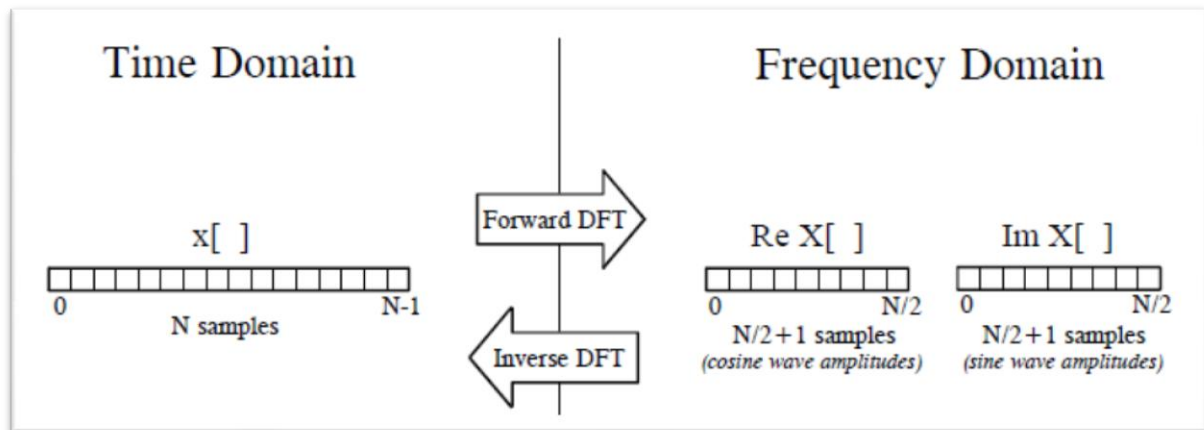
$$x(n) = A(0) + \sum_{k=1}^{\frac{N}{2}-1} A(k) \cos\left(k \frac{2\pi}{N} n\right) + \sum_{k=1}^{\frac{N}{2}-1} B(k) \sen\left(k \frac{2\pi}{N} n\right) + A\left(\frac{N}{2}\right) \cos(\pi n)$$

$$x(n) = A(0) + A(1) \cos\left(\frac{\pi}{2} n\right) + B(1) \sen\left(\frac{\pi}{2} n\right) + A(2) \cos(\pi n)$$

$$x(n) = 1.5 - \cos\left(\frac{\pi}{2} n\right) - \sen\left(\frac{\pi}{2} n\right) - 0.5 \cos(\pi n)$$

Forma matricial

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^0 & W_4^2 \\ W_4^0 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$



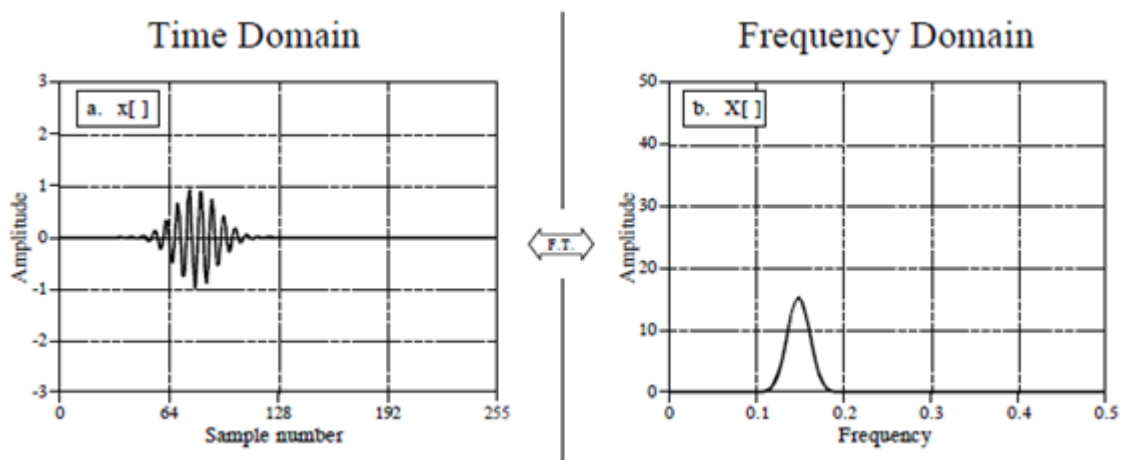
Los valores de $\text{Re } X[k]$ son las amplitudes de las señales coseno y los valores de $\text{Im } X[k]$ son las amplitudes de las señales seno.

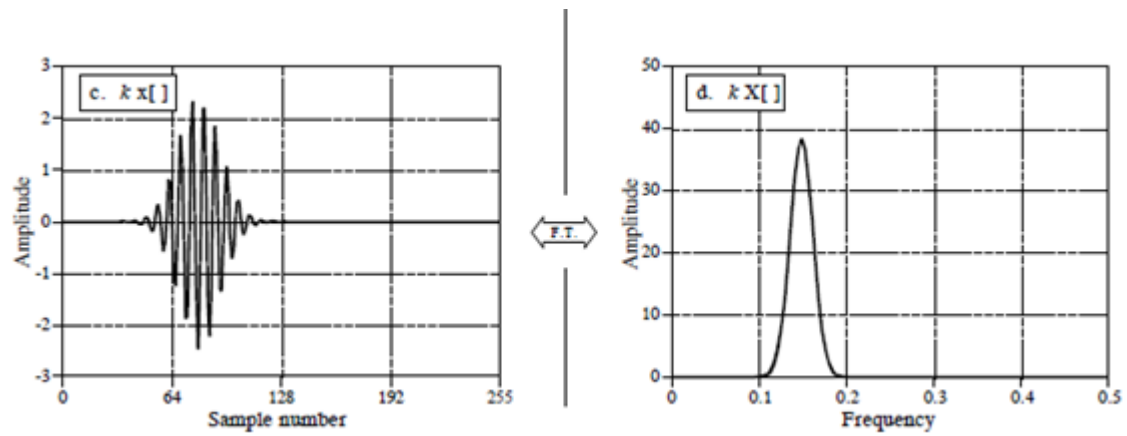
PROPIEDADES DE LA DTF

1. LINEALIDAD

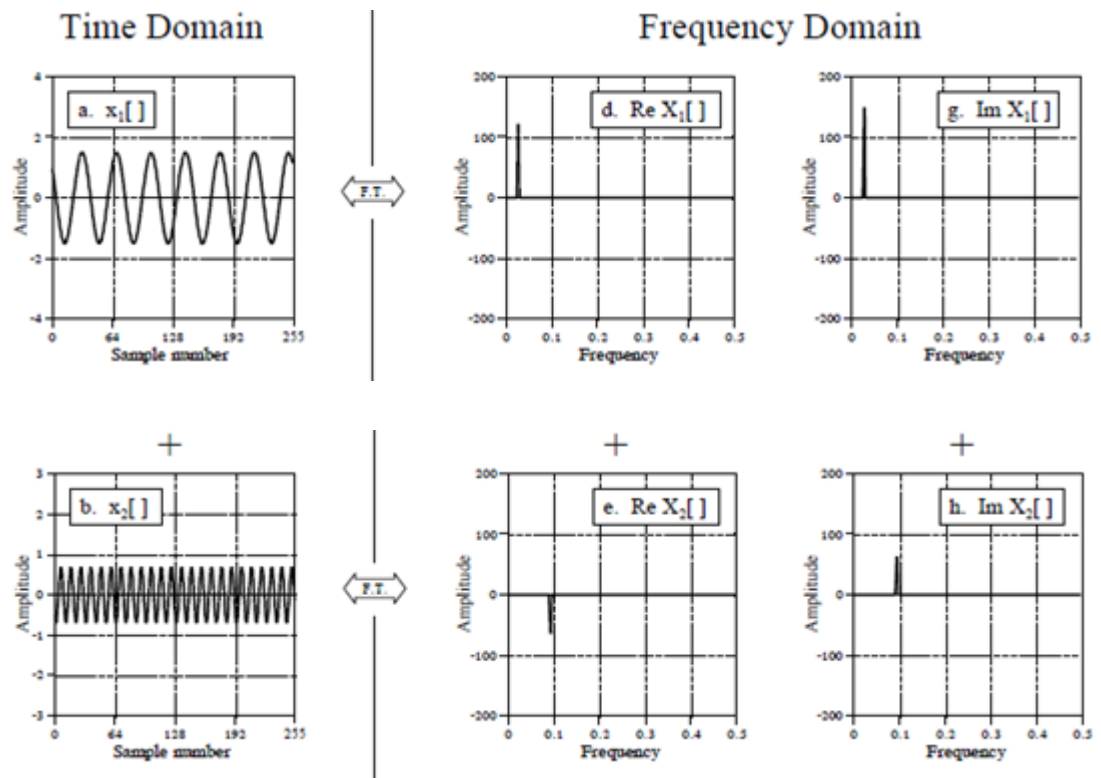
La Transformada de Fourier es lineal, esto es, tiene las propiedades de homogeneidad y aditividad.

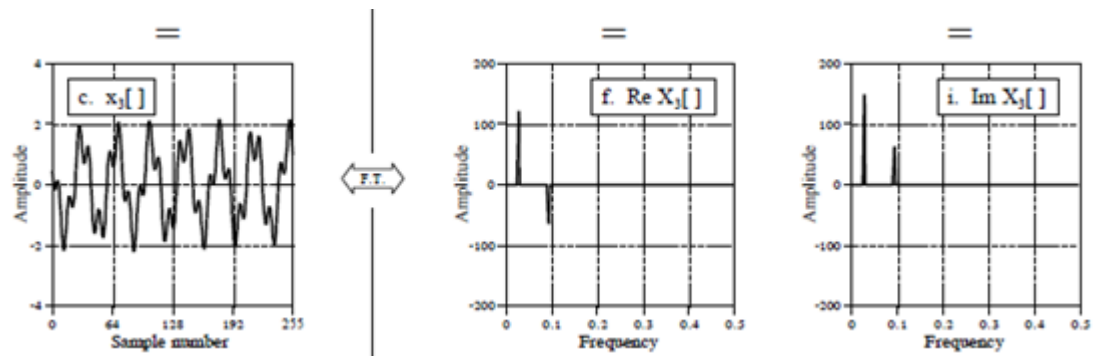
Homogeneidad





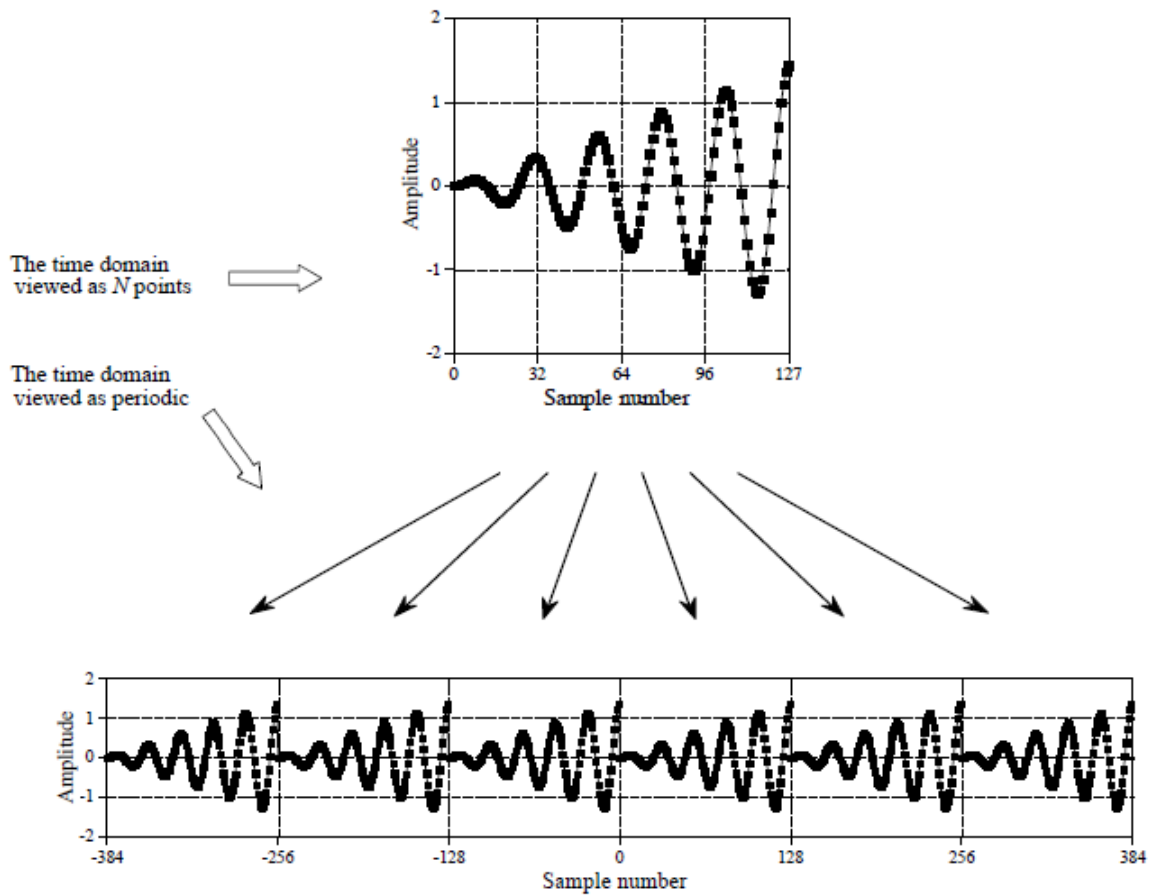
Aditibilidad





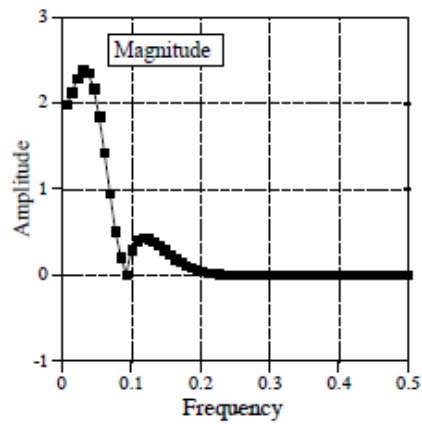
2. PERIODICIDAD

Dominio del tiempo

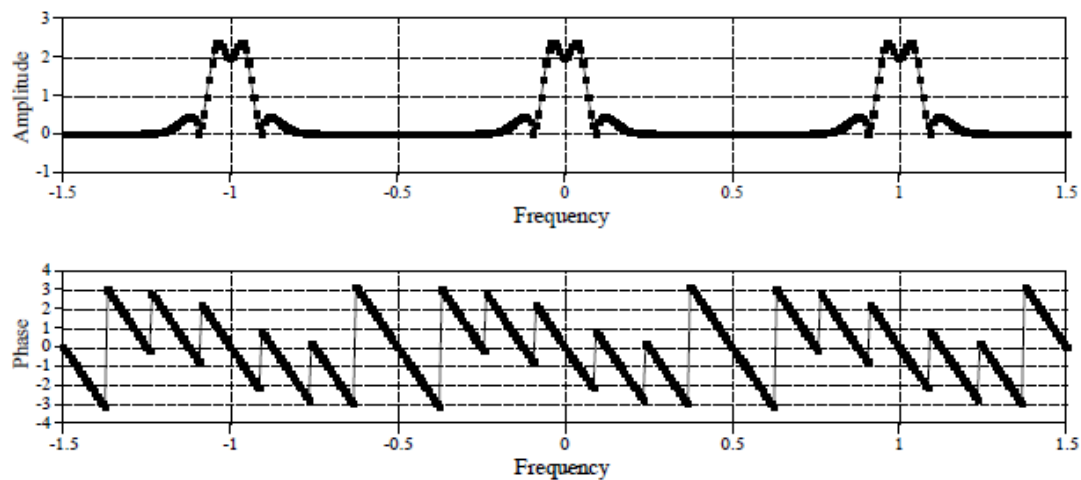
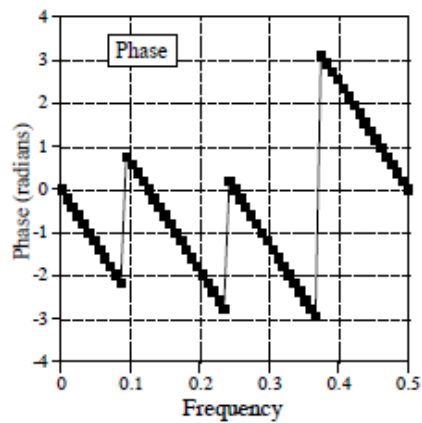


Dominio de la frecuencia

The frequency domain
viewed as 0 to 0.5 of
the sampling rate

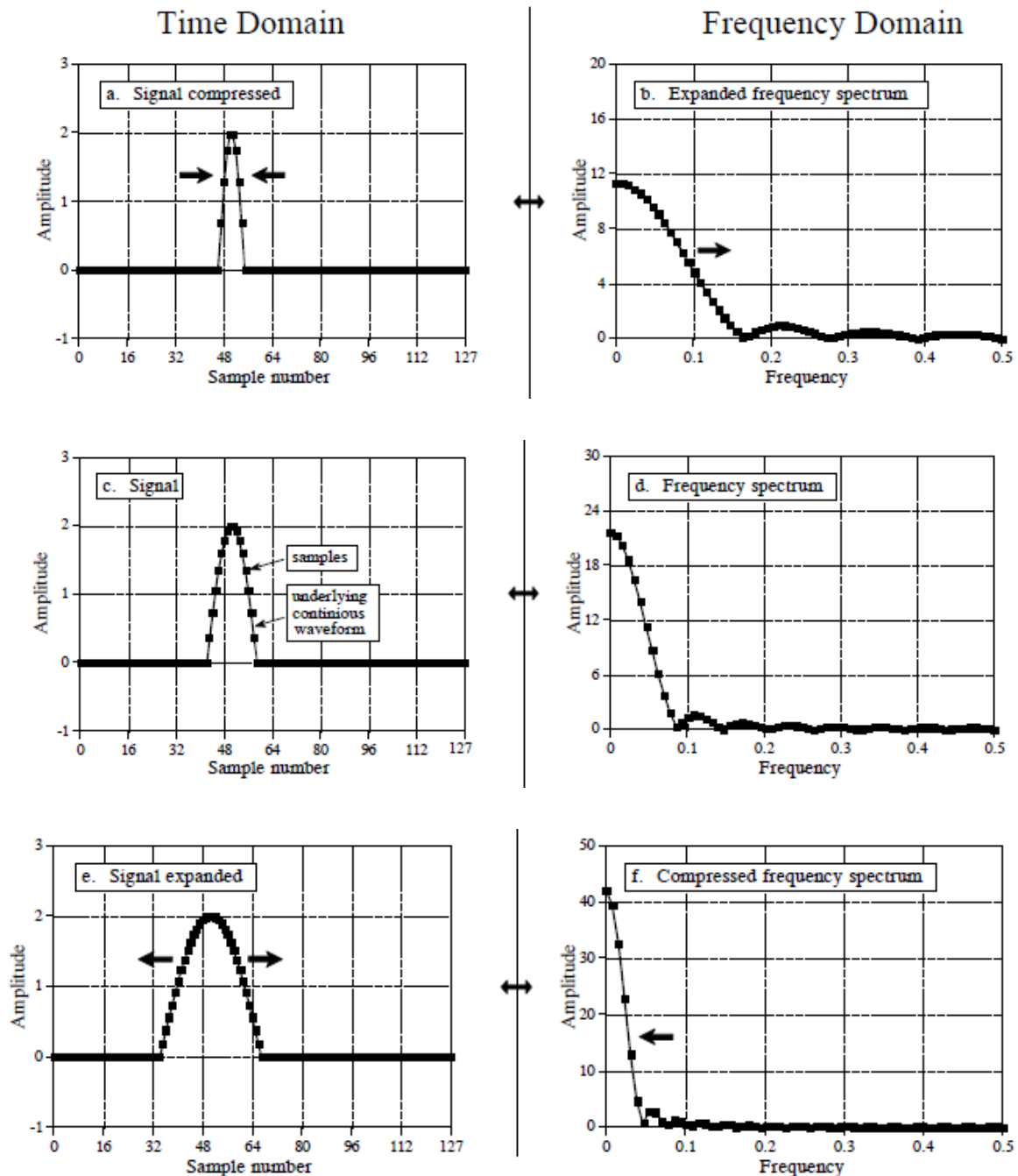


The frequency domain
viewed as periodic



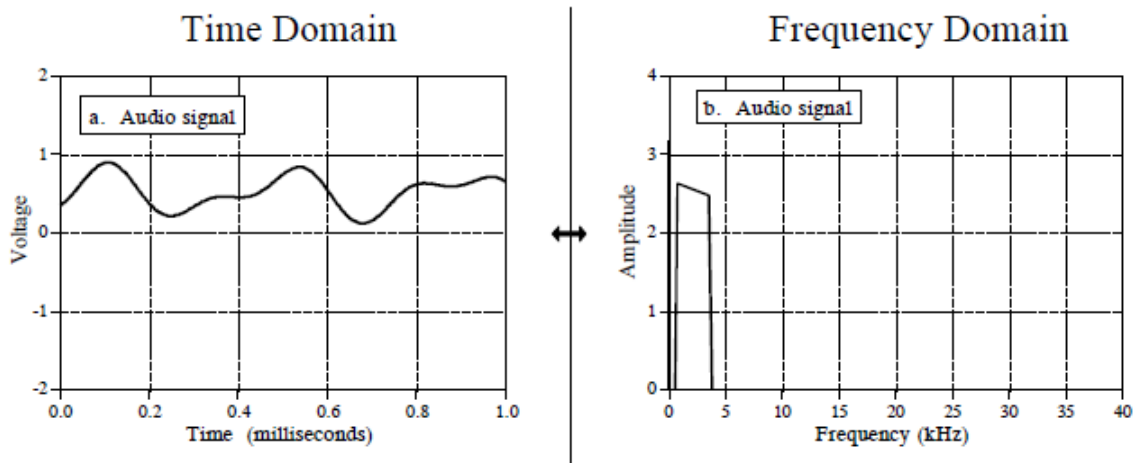
3. COMPRESIÓN Y EXPANSIÓN

Una compresión de la señal en un dominio resulta en una expansión en el otro y viceversa. Para señales continuas, si $X(f)$ es la Transformada de Fourier de $x(t)$, entonces, $1/k \cdot X(f/k)$ es la Transformada de Fourier de $x(kt)$ donde k es el parámetro que controla la expansión o compresión.

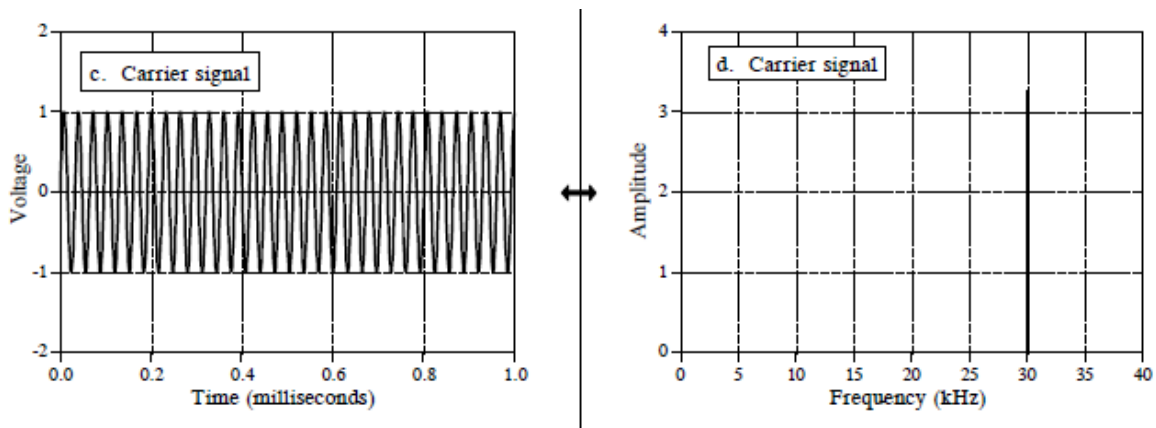


4. MODULACIÓN DE AMPLITUD

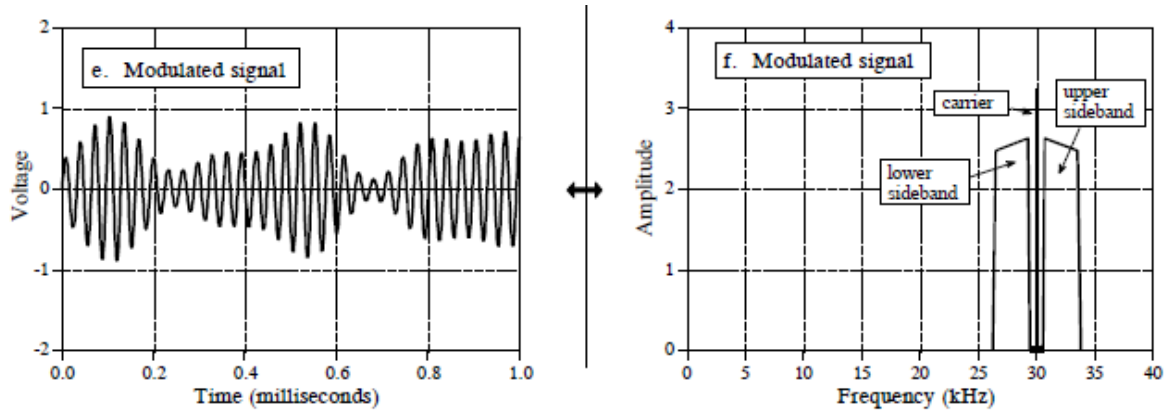
Inicialmente se tiene una señal de audio con polarización DC en la fig a. En la fig b. muestra el espectro de frecuencia de 300 Hz a 3 KHz más un pico de componente DC. Todas las demás frecuencias son filtradas.



Las figuras c y d muestran la señal portadora.



En el dominio del tiempo la modulación consiste de multiplicar la señal de audio por la onda portadora.



Puesto que la multiplicación en el dominio del tiempo corresponde a la convolución en el dominio de la frecuencia, el espectro de la señal modulada es el espectro de la señal modulada corrida a la frecuencia de la portadora. El resultado es un espectro modulado de tres componentes: Onda portadora y las dos bandas laterales.

TRANSFORMADA COSENO DISCRETO – DCT

Tiene mejor compactación de energía que la DFT con pocos coeficientes transformados útil en aplicaciones de comunicación de datos. La DCT es una transformada completamente real a diferencia de la DFT que requiere números complejos. Matemáticamente para una secuencia de entrada $x(n)$, la DCT es:

$$y(k) = w(k) \sum_{n=1}^N x(n) \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right); \quad k = 1, 2, \dots, N$$

Donde,

$$w(k) = \begin{cases} \sqrt{\frac{1}{N}}, & k = 1 \\ \sqrt{\frac{2}{N}}, & 2 \leq k \leq N \end{cases}$$

La transformada Inversa coseno discreto IDCT es;

$$x(n) = \sum_{k=1}^N w(k) y(k) \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right); \quad k = 1, 2, \dots, N$$

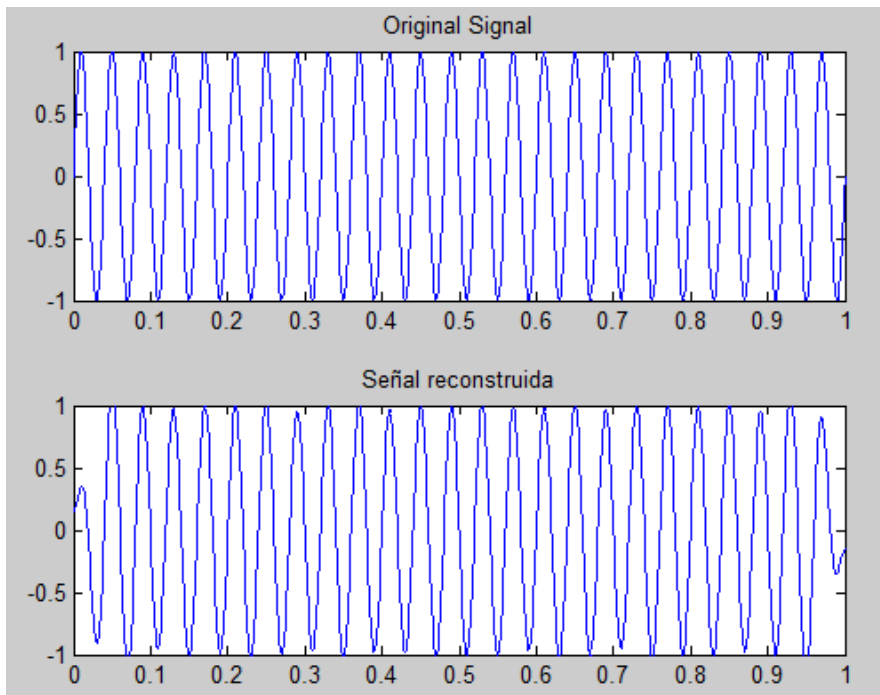
Donde,

$$w(n) = \begin{cases} \sqrt{\frac{1}{N}}, & k = 1 \\ \sqrt{\frac{2}{N}}, & 2 \leq k \leq N \end{cases}$$

EJEMPLO:

Generar una secuencia senoidal de 25 Hz con frecuencia de muestreo de 1000 Hz, calcular la DCT y reconstruir la señal usando solamente los componentes con valor mayor a 0.9 (17 coeficientes de los 1000 originales)

```
%ejemplo de dct
t = (0:1/999:1); % 1000 puntos
x = sin(2*pi*25*t);
y = dct(x); % calcula la dct
y2 = find(abs(y) < 0.9);
y(y2) = zeros(size(y2));
%size(y2)=983, solo se usarán 17 componentes
z = idct(y); % reconstruir señal
subplot(2,1,1); plot(t,x);
title('Original Signal')
subplot(2,1,2); plot(t,z), axis([0 1 -1 1])
title('Señal reconstruida')
```



TRANSFORMADA DE HILBERT

Facilita la formación de la señal analítica útil en el área de comunicaciones especialmente en el procesamiento de señales de pasa banda. Calcula la transformada de una secuencia de entrada real $x(n)$ y retorna un resultado complejo de igual longitud.

Para aproximarse a la señal analítica, Hilbert calcula la FFT de la secuencia de entrada, reemplaza los coeficientes que corresponden a las frecuencias negativas con ceros y calcula la IFFT del resultado.

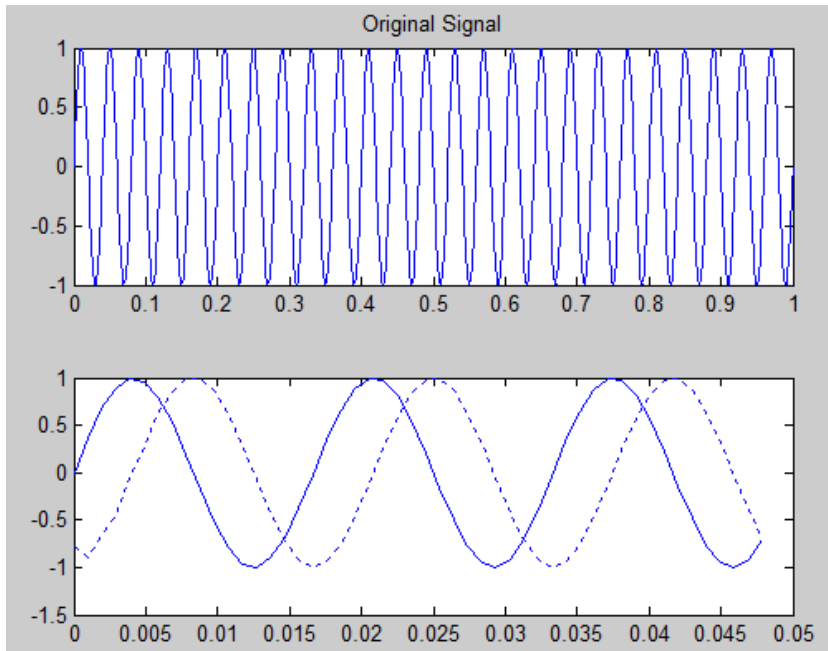
`y = hilbert(x)`

La parte real es el dato real y la parte imaginaria es la transformada.

```
xr = [1 2 3 4];
x = hilbert(xr)
% x =1.0000+1.0000i 2.0000-1.0000i 3.0000-1.0000i 4.0000+1.0000i
```

EJEMPLO:

```
%ejemplo hilbert
t = (0:1/1023:1);
x = sin(2*pi*60*t);
y = hilbert(x);
plot(t(1:50),real(y(1:50))), hold on
plot(t(1:50),imag(y(1:50)),':'), hold off
```



La transformada o parte imaginaria tiene un desfase de 90°

TRANSFORMADA RÁPIDA DE FOURIER - FFT

En las últimas décadas se han investigado muchos métodos para reducir el número de multiplicaciones en el cálculo de la DFT. La técnica más importante y popularizada es la de J. W. Cooley – John Tukey (1915-2000) basada en romper la transformada en transformadas más pequeñas y luego combinarlas para obtener la transformada total (1965). El método es el proceso de decimación en el tiempo y en frecuencia.

FFT Decimación en el tiempo

El número de puntos se asume potencia de 2, o sea, $N=2^y$. La decimación en el dominio consiste en romper la transformada de N puntos en dos transformadas de N/2 puntos, luego cada transformada de N/2 puntos en dos transformadas de N/4 puntos y continuar la división hasta que se llegue a una transformada de 2 puntos.

La señal discreta se divide en la parte par e impar:

$$x(n) = [x(0), x(1), x(2), x(3), \dots, x(N/2-1), \dots, x(N-1)]$$

Pares: $[x(0), x(2), x(4), \dots, x(N-2)]$

Impares: $[x(1), x(3), x(5), \dots, x(N-1)]$

$$DFT[x(n)] = X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}$$

Rompiendo las pares e impares:

$$X(k) = \sum_{n=0}^{N-2} x(n)W_N^{kn} + \sum_{n=1}^{N-1} x(n)W_N^{kn}$$

(n = par) (n = Impar)

Pares: $n = 2r \quad \Psi_r = n/2$ Impares: $n = 2r + 1 \quad \Psi_r = (n-1)/2$

$$X(k) = \sum_{r=0}^{N/2-1} x(2r)W_N^{2rk} + \sum_{r=0}^{N/2-1} x(2r+1)W_N^{(2r+1)k}$$

$$W_N^{2rk} = (W_N^2)^{rk} = e^{-j\frac{2\pi}{N}2rk} = e^{-j\frac{2\pi}{N}rk} = W_{N/2}^{rk}$$

$$X(k) = \underbrace{\sum_{r=0}^{N/2-1} x(2r)W_{N/2}^{rk}}_{G(k) = \text{DFT de } \frac{N}{2} \text{ puntos pares}} + W_N^k \underbrace{\sum_{r=0}^{N/2-1} x(2r+1)W_{N/2}^{rk}}_{H(k) = \text{DFT de } \frac{N}{2} \text{ puntos impares}}$$

$$\mathbf{X(k) = G(k) + W_N^k H(k)}$$

$$X(0) = G(0) + W_N^0 H(0)$$

$$X(1) = G(1) + W_N^1 H(1)$$

$$X(2) = G(2) + W_N^2 H(2)$$

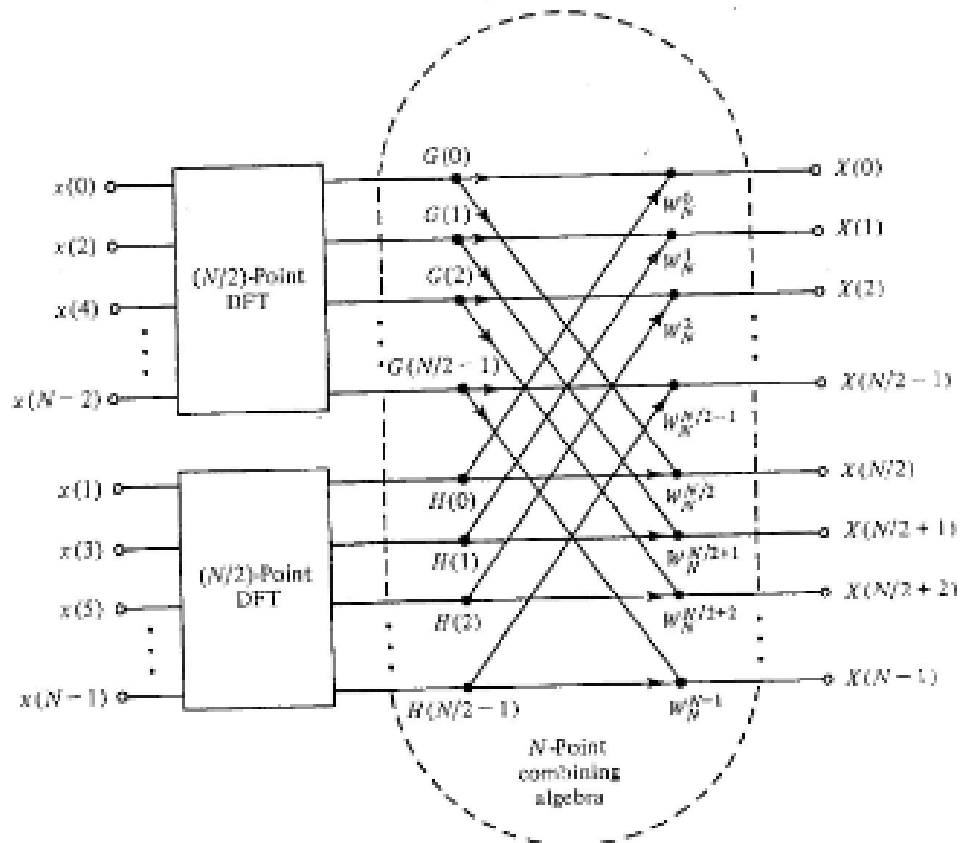
$$X(3) = G(3) + W_N^3 H(3)$$

.....

$$X\left(\frac{N}{2}-1\right) = G\left(\frac{N}{2}-1\right) + W_N^{\frac{N}{2}-1} H\left(\frac{N}{2}-1\right)$$

.....

$$X(N-1) = G\left(\frac{N}{2}-1\right) + W_N^{N-1} H\left(\frac{N}{2}-1\right)$$

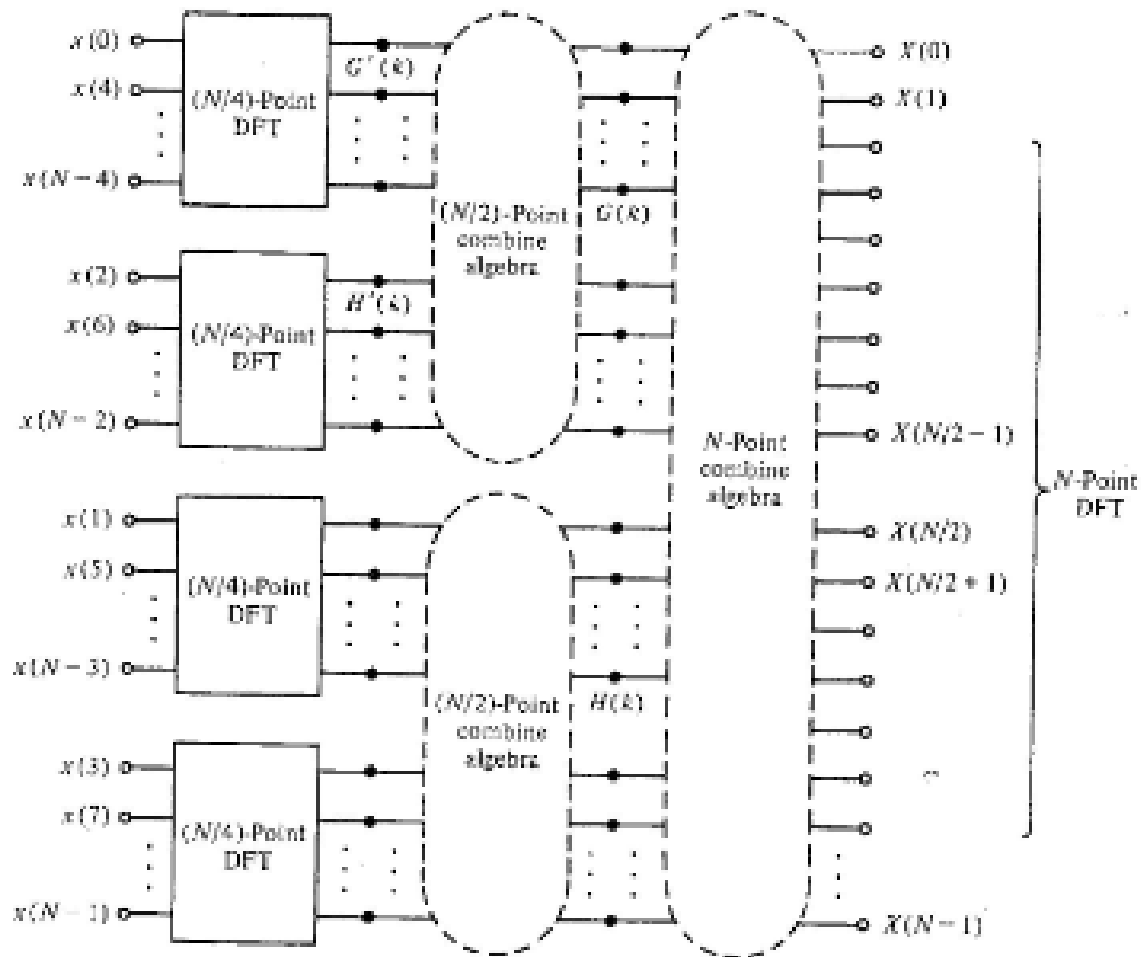


El número de multiplicaciones:

$$\# \text{ Multiplicaciones} = \left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 + N = N + \frac{N^2}{2} < N^2$$

Si $N=8$, el # multiplicaciones = $40 < 64$

Ahora cada secuencia de $N/2$ puntos puede ser decimada en dos secuencias de longitud $N/4$.



Ahora el número de multiplicaciones es:

$$\# \text{ Multiplicaciones} = 4 \left(\frac{N}{4} \right)^2 + 2 \left(\frac{N}{2} \right) + N = \frac{N^2}{4} + 2N$$

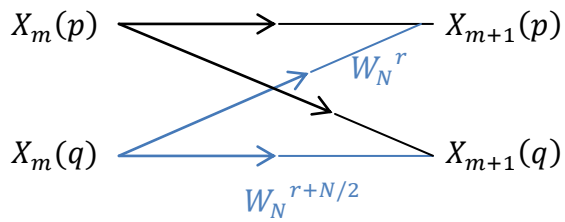
Para $N=8$; #multiplicaciones = 32 < 40

En general:

$$X_{m+1}(p) = X_m(p) + W_N^r X_m(q)$$

$$X_{m+1}(q) = X_m(p) + W_N^{r+N/2} X_m(q)$$

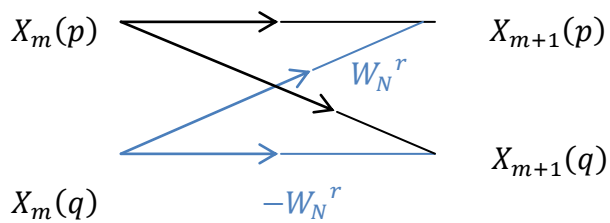
Que se puede representar como:



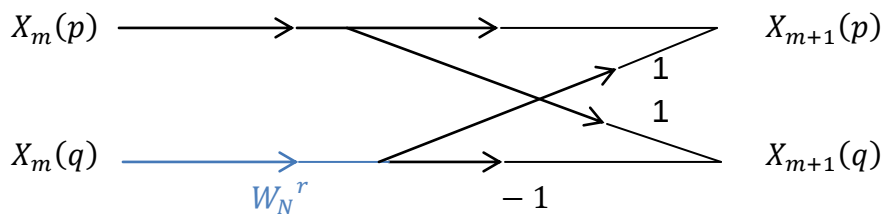
Simplificando:

$$W_N^{r+N/2} = W_N^r \cdot W_N^{N/2} = W_N^r \cdot e^{-j\left(\frac{2\pi}{N}\right)\frac{N}{2}} = W_N^r \cdot e^{-j\pi} = -W_N^r$$

El esquema queda de la siguiente forma:



Arreglando de otra forma el esquema:



Se reduce el número de operaciones (Cooley – Tukey) a:

$$\# \text{ Multiplicaciones} = \frac{N}{2} \log_2 N$$

$$\# \text{ Sumas} = N \cdot \log_2 N$$

DTF CON MATLAB

En Matlab la Transformada de Fourier Discreta DFT de $x(n)$ se calcula con el comando:

```
xn = fft (Xk)
```

y la Transformada Inversa de Fourier Discreta $X(k)$:

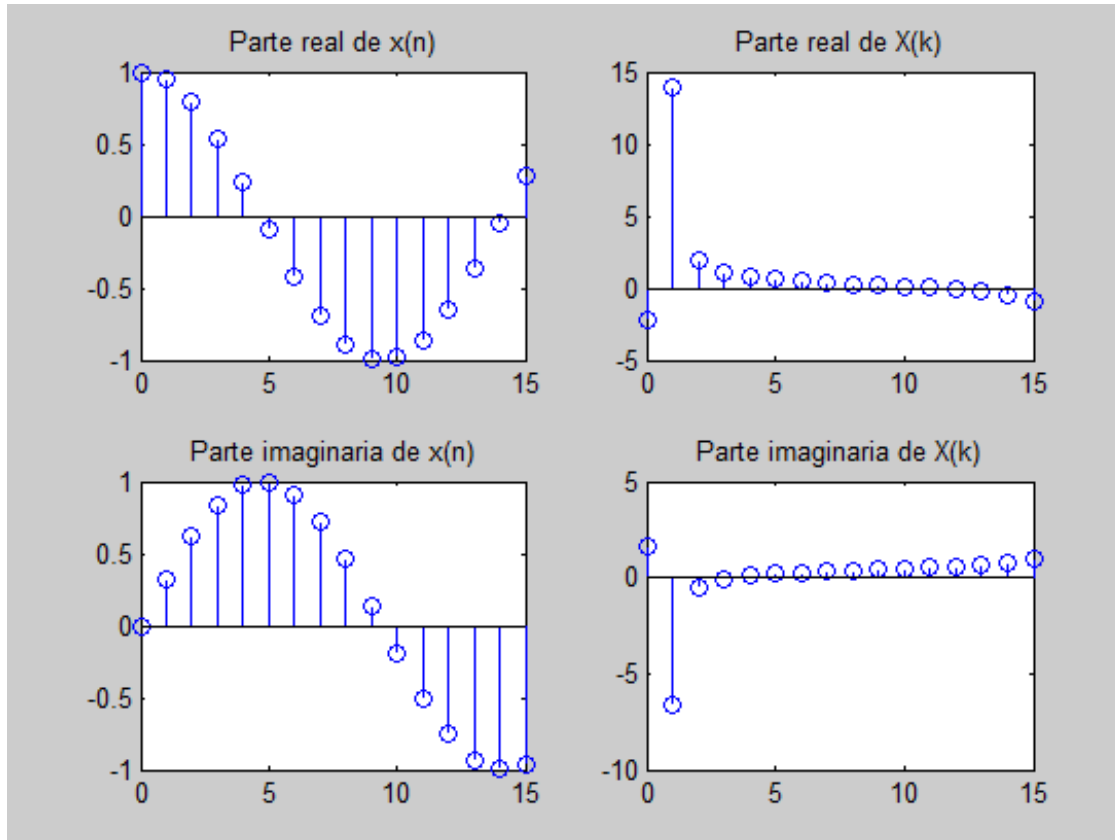
```
Xk = ifft (xn)
```

EJEMPLO 2: Obtener parte real y parte imaginaria

A continuación se tiene la parte real e imaginaria de la señal y de su DFT

$$x(n) = e^{j\frac{n}{3}}$$

```
%DFT PARTE REAL E IMAGINARIA
clear all
n=0:15;
xn=exp(j*n/3);
Xk=fft(xn);
k=n;
subplot(221)
stem(k,real(Xk));
title('Parte real de X(k)')
subplot(223)
stem(k,imag(Xk));
title('Parte imaginaria de X(k)')
%recuperar la señal x(n)
xn=ifft(Xk);
subplot(222)
stem(n,real(xn));
title('Parte real de x(n)')
subplot(224)
stem(n,imag(xn));
title('Parte imaginaria de x(n)')
```



EJEMPLO 3: Obtener la magnitud y la fase

Obtener la magnitud y fase de la DFT de la señal,

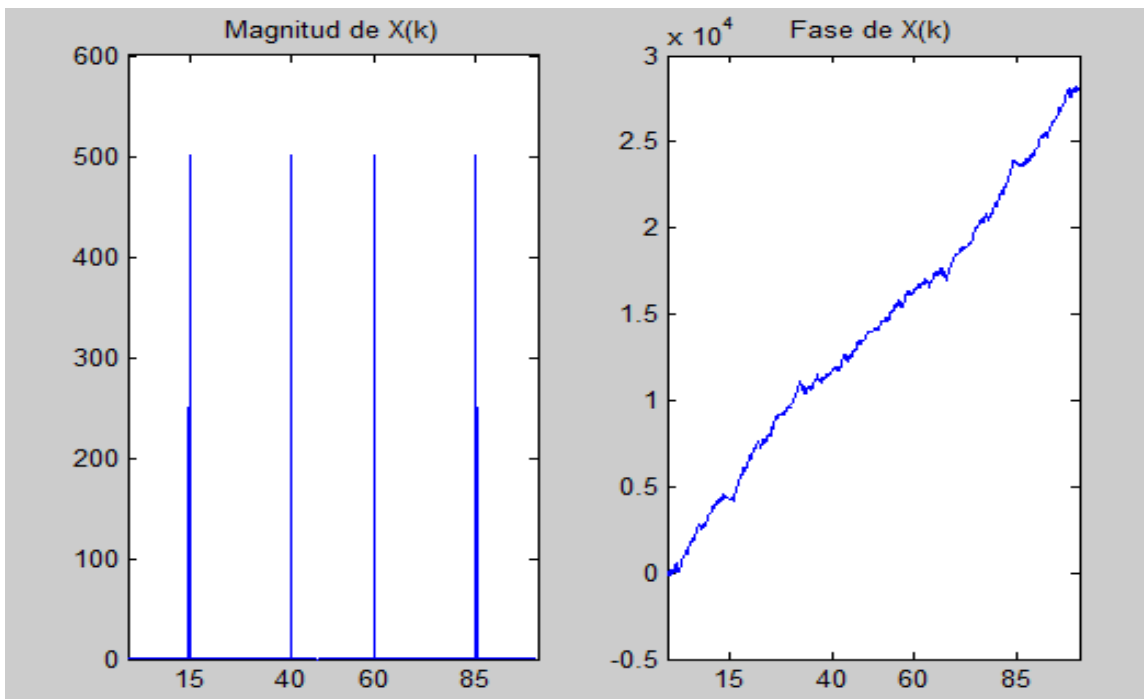
$$x(n) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t), \quad f_1 = 15 \text{ Hz} \text{ y } f_2 = 40 \text{ Hz}$$

```
%MAGNITUD Y FASE
clear all
t=0:0.01:9.99; % 1000 puntos
f1=15;f2=40;
xn=sin(2*pi*f1*t)+sin(2*pi*f2*t);
Xk=fft(xn);
mag=abs(Xk);
f=0:0.1:99.9;
```

```

subplot(121)
plot(f,mag)
title('Magnitud de X(k)')
set(gca,'Xtick',[15 40 60 85])
subplot(122)
fase=unwrap(angle(Xk));
plot(f,fase*180/pi)
title('Fase de X(k)')
set(gca,'Xtick',[15 40 60 85])

```



APLICACIONES DE LA FFT

1. RESPUESTA EN FRECUENCIA

Obtener la FFT de una señal senoidal de 20 Hz sumada a una señal tipo chirp con un desplazamiento desde 5 hasta 40 Hz en un tiempo D.

```

clear all
D=1; N=128;
ts=D/N;

```

```

d=ts/2;
t=0:ts:D-d;
x=sin(2*pi*20*t)+chirp(t,5,D,40);
X=fft(x);

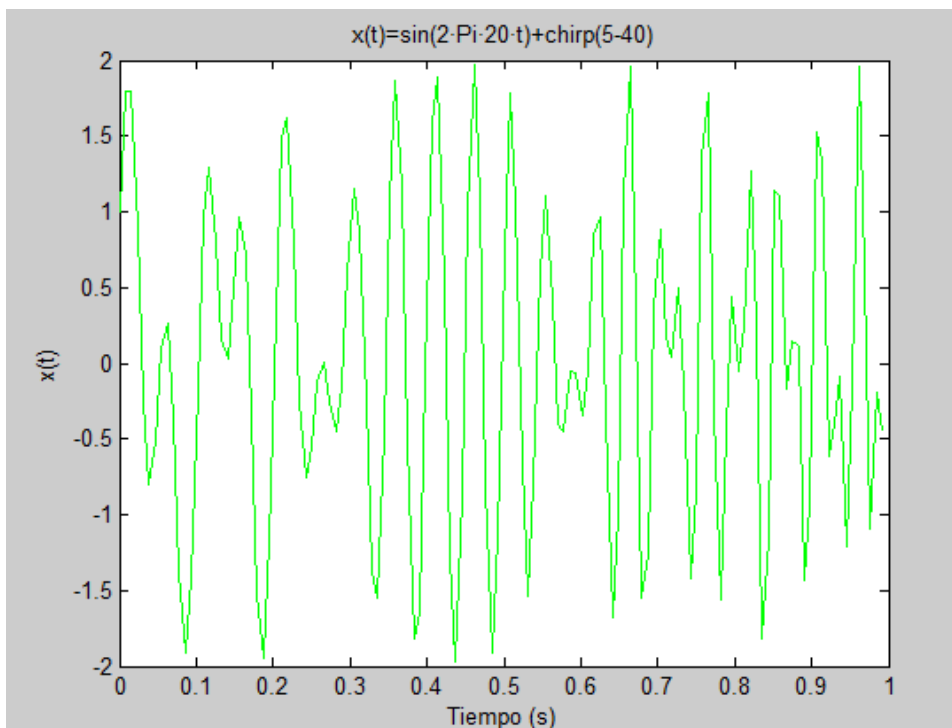
%Reordenar X
M=N/2;
Xaux=X;
X(M+1:N)=Xaux(1:M);
X(1:M)=Xaux(M+1:N);

Xm=abs(X)/N;
Xf=unwrap(angle(X))*180/pi;

%Reordenar los indices k
faux(M+1:N)=0:M-1;
faux(1:M)=-M:-1;
f=faux/D;

% Gráfica de la señal
figure;
plot(t,x,'-g');zoom;
xlabel('Tiempo (s)');ylabel('x(t)');
title('x(t)=sin(2·Pi·20·t)+chirp(5-40)');

```

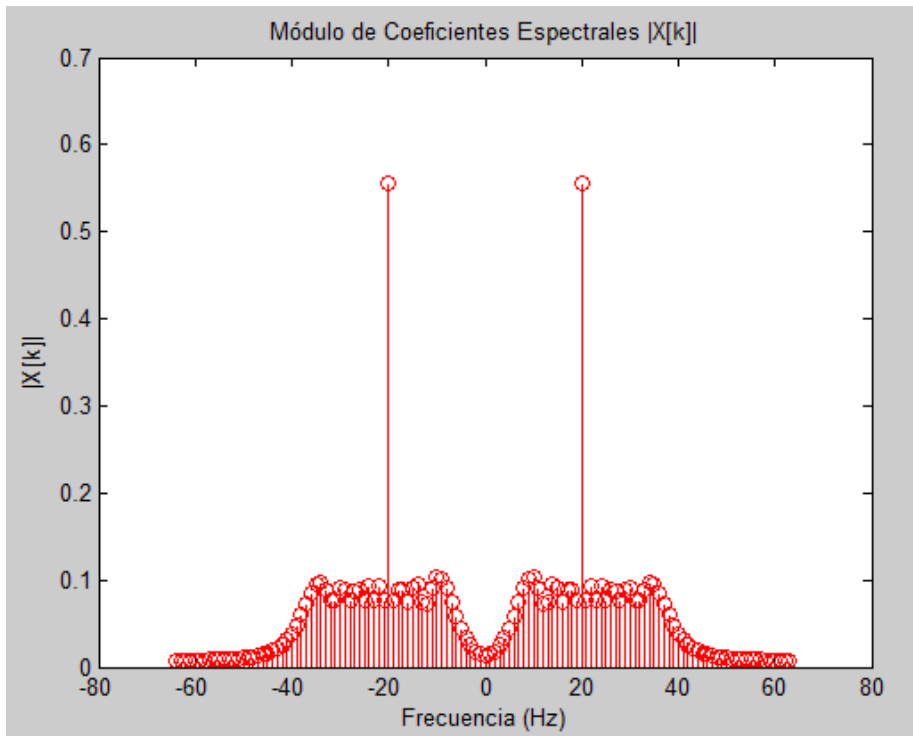


```

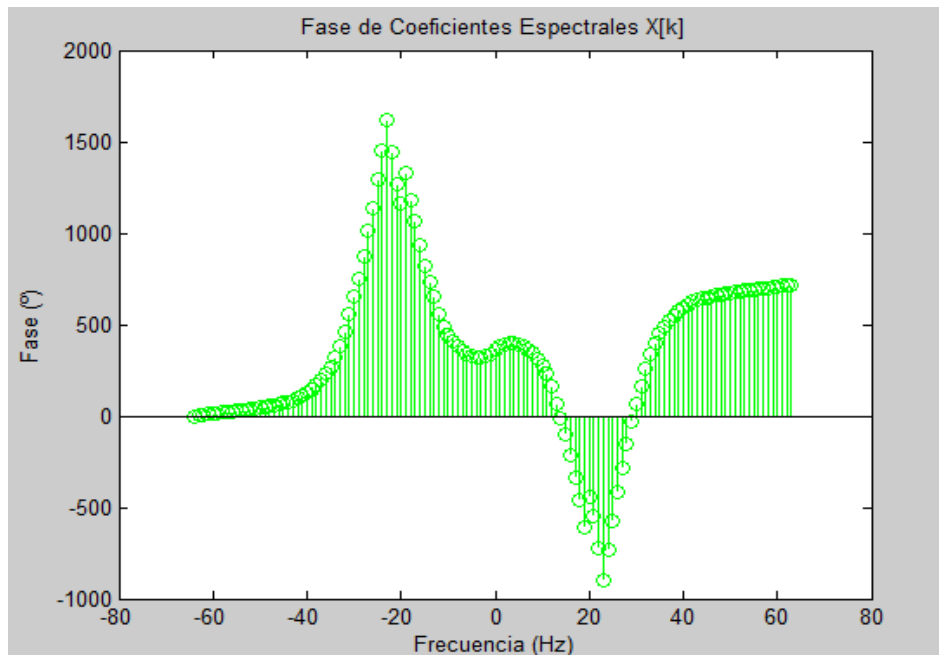
% Graficar fft de xn Xk
figure;
stem(f,Xm,'r')
zoom;

```

```
xlabel('Frecuencia (Hz)');ylabel('|X[k]|');
title('Módulo de Coeficientes Espectrales |X[k]|');
```



```
% Grafica de la fase
figure;
stem(f,Xf,'g');zoom;
xlabel('Frecuencia (Hz)');ylabel('Fase (°)');
title('Fase de Coeficientes Espectrales X[k]');
```



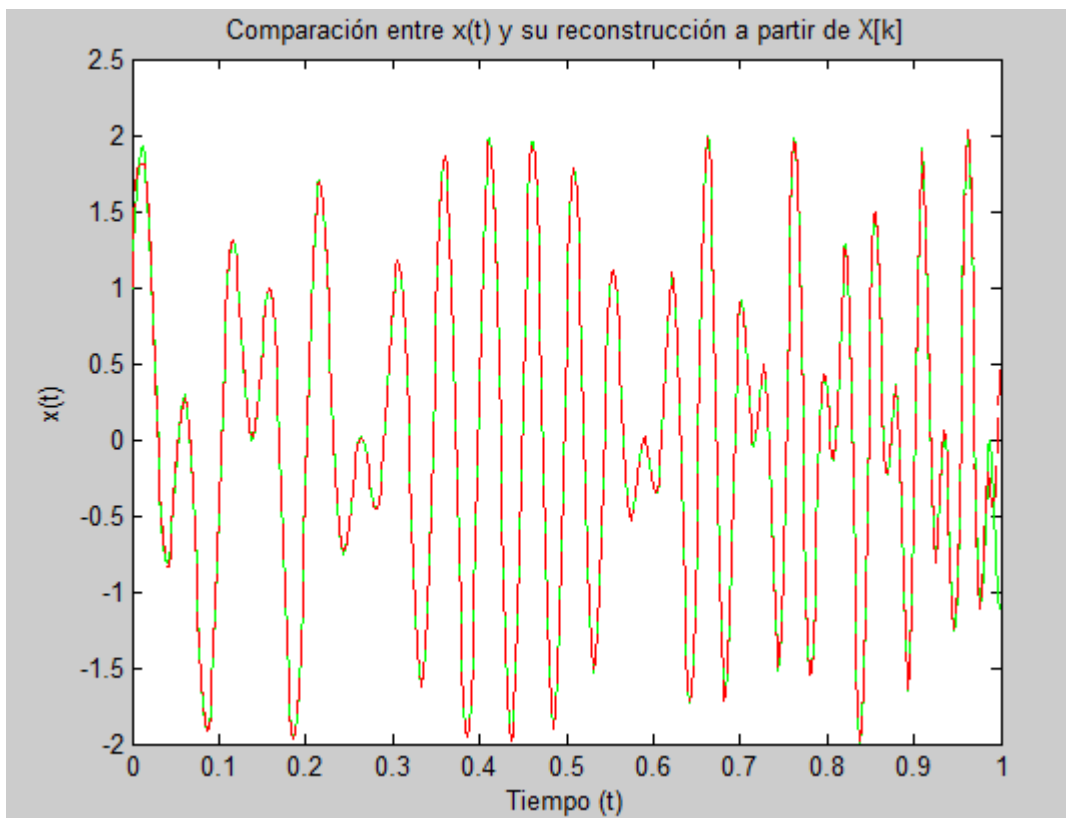
```

% Reconstrucción de la señal a partir de los X[k]
% Utilizamos un mayor número de puntos fs=500 Hz

fs=500;
ts=1/fs;
d=ts/2;
t=0:ts:D-d;
Ns=length(t);
%x=sin(2*pi*20*t)+chirp([5 40]*ts,Ns);
x=sin(2*pi*20*t)+chirp(t,5,D,40);
xr=zeros(1,Ns);
for i=1:Ns
    for k=1:N
        xr(i)=xr(i)+X(k)*exp(j*2*pi*f(k)*ts*(i-1))/N;
    end
end

figure;plot(t,x,'g-');hold on;plot(t,xr,'r--');zoom;
title('Comparación entre x(t) y su reconstrucción a partir de X[k]');
xlabel('Tiempo (t)');ylabel('x(t)');

```



2. MODULACIÓN EN AMPLITUD

Obtener la FFT de una señal exponencial modulada en amplitud con una frecuencia portadora de 200 Hz. N es el número de puntos de muestreo durante D segundos de la señal. Se requiere una frecuencia de muestreo de por lo menos 400 Hz, $N/D > 400$

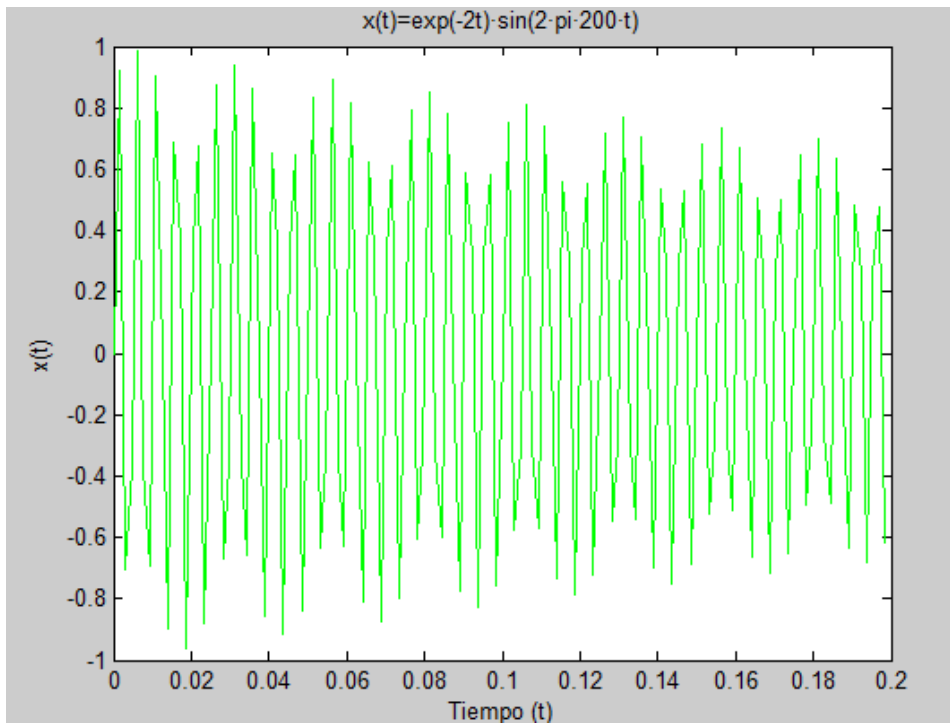
```
clear all
N=128; D=0.2;
ts=D/N;
d=ts/2;
t=0:ts:D-d;
x=exp(-2*t).*sin(2*pi*200*t);
X=fft(x);

%Reordenar X
M=N/2;
Xaux=X;
X(M+1:N)=Xaux(1:M); X(1:M)=Xaux(M+1:N);

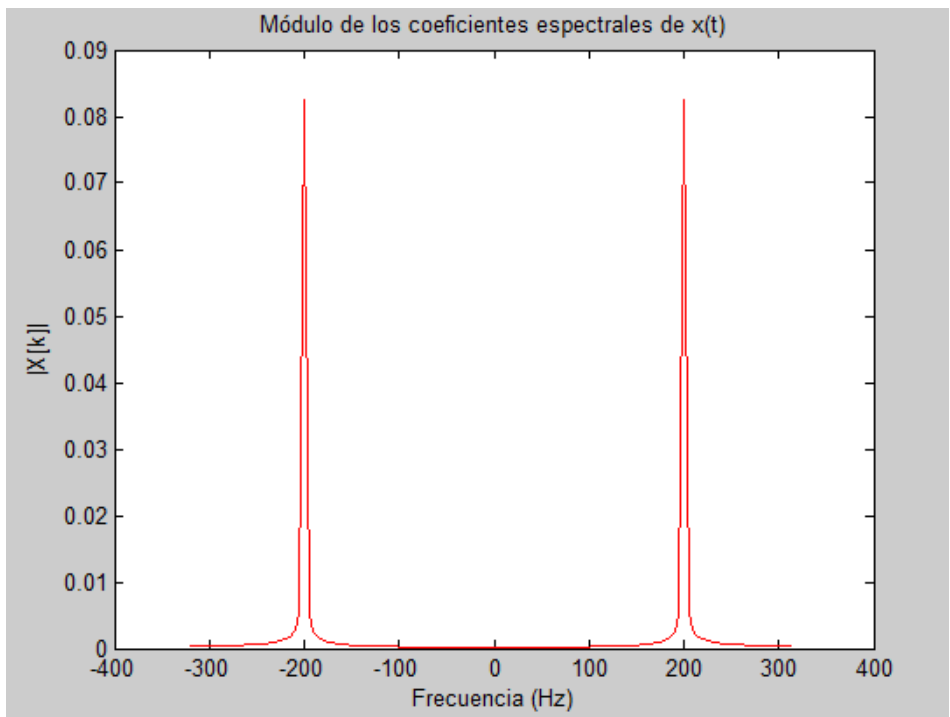
Xm=abs(X)*ts;
Xf=unwrap(angle(X))*180/pi;

%Reordenar los indices k
faux(M+1:N)=0:M-1; faux(1:M)=-M:-1;
f=faux/D;

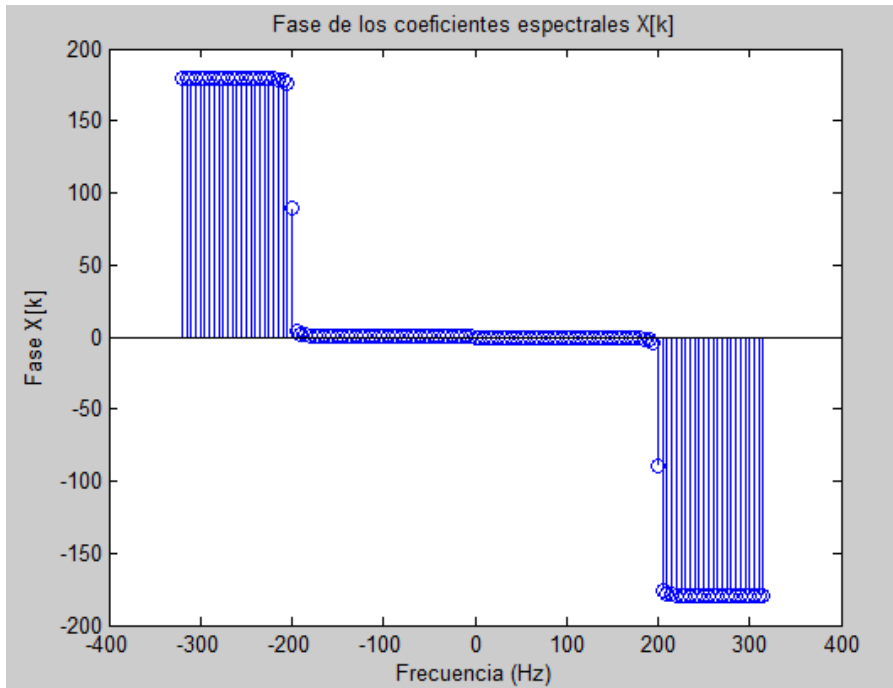
figure;
plot(t,x,'-g'); zoom;
title('x(t)=exp(-2t)·sin(2·pi·200·t)');
xlabel('Tiempo (t)'); ylabel('x(t)');
```



```
figure;
plot(f,Xm,'-r')
title('Módulo de los coeficientes espectrales de x(t)');
xlabel('Frecuencia (Hz)');ylabel('|X[k]|');
```



```
figure;
stem(f,Xf,'-b');zoom;
title('Fase de los coeficientes espectrales X[k]');
xlabel('Frecuencia (Hz)');ylabel('Fase X[k]');
```



3. MODULACIÓN EN FRECUENCIA

```
clear all
D=0.5;N=256;
ts=D/N;
d=ts/2;
t=0:ts:D-d;
x=sin(2*pi*200*t+5*sin(2*pi*2*t));
X=fft(x);

%Reordenar X
M=N/2;
Xaux=X;X(M+1:N)=Xaux(1:M);
X(1:M)=Xaux(M+1:N);

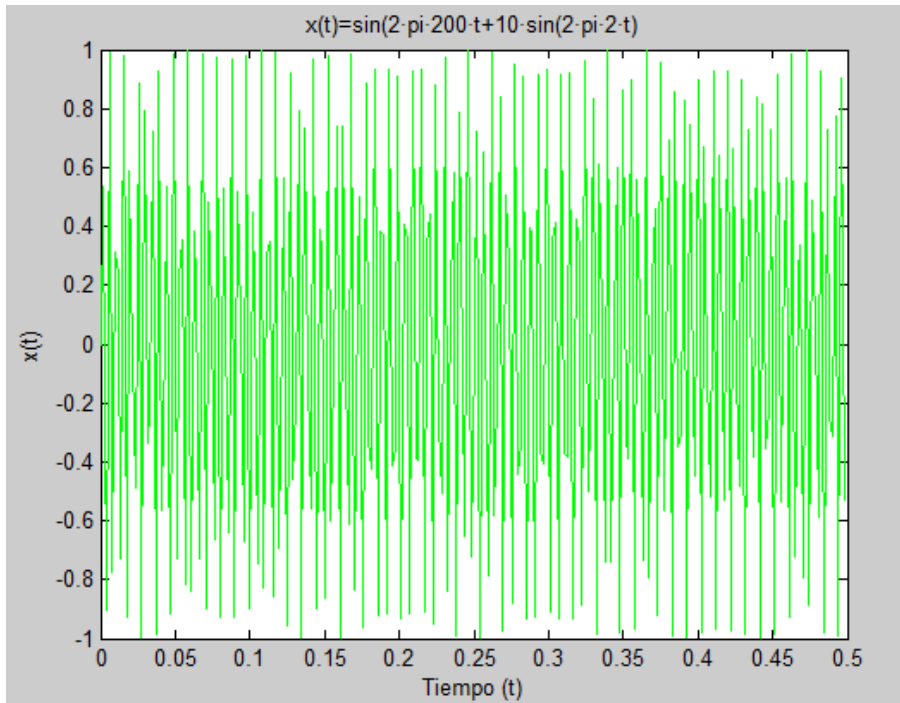
Xm=abs(X)*ts;
Xf=unwrap(angle(X))*180/pi;
```

```

%Reordenar los indices k
faux(M+1:N)=0:M-1;faux(1:M)=-M:-1;
f=faux/D;

figure;
plot(t,x,'-g');zoom;
title('x(t)=sin(2·pi·200·t+10·sin(2·pi·2·t))');
xlabel('Tiempo (t)');ylabel('x(t)');

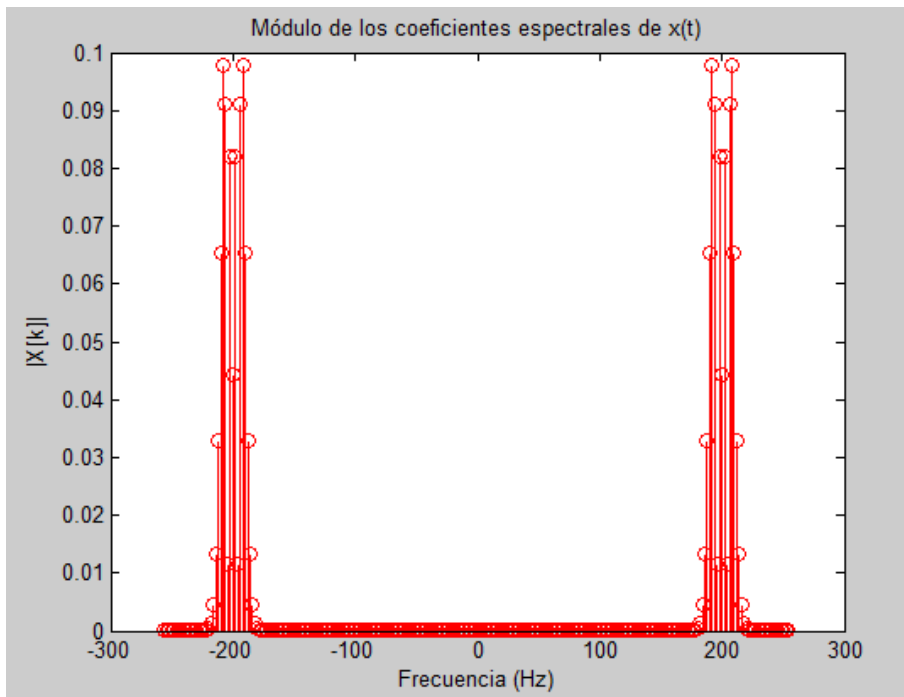
```



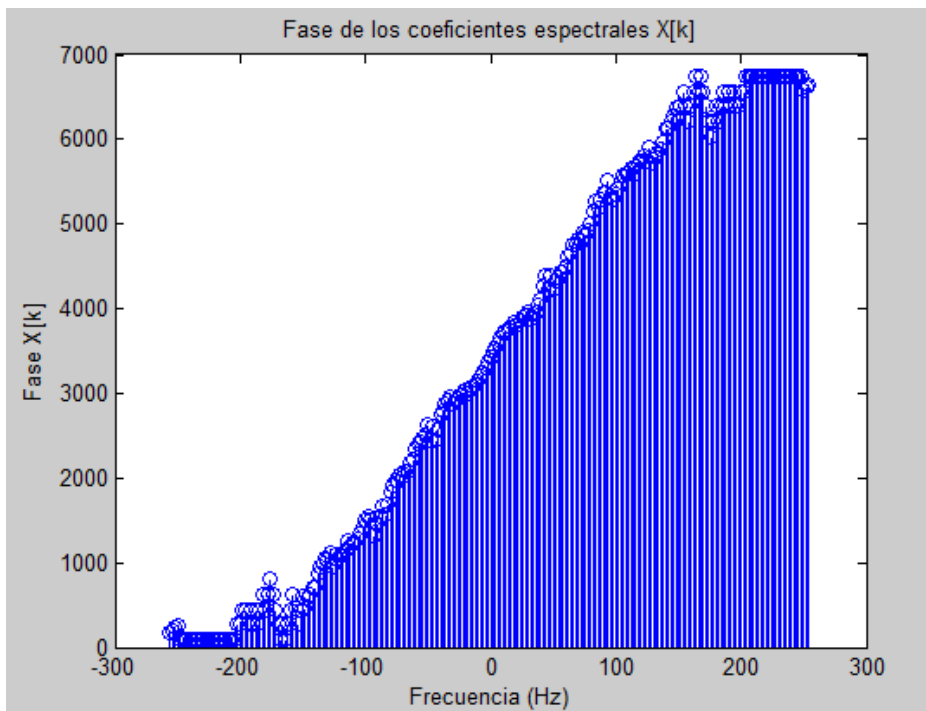
```

figure;
stem(f,Xm,'-r');zoom;
title('Módulo de los coeficientes espectrales de x(t)');
xlabel('Frecuencia (Hz)');ylabel('|X[k]|');

```



```
figure;
stem(f,Xf,'-b');zoom;
title('Fase de los coeficientes espectrales X[k]');
xlabel('Frecuencia (Hz)');ylabel('Fase X[k]');
```



4. ANÁLISIS ESPECTRAL

El análisis espectral describe la distribución en función de la frecuencia de la potencia contenida en una señal, basado en un conjunto finito de datos. En términos generales, la manera de estimar la PSD (densidad de potencia espectral) de un proceso es encontrar la Transformada de Fourier discreta DFT (usando la FFT) y tomar la magnitud al cuadrado del resultado. Esta estimación es llamada Periodograma.

El Periodograma estimado de la PSD de una señal de longitud L para un número de puntos de frecuencia N ($N > L$) es:

$$\hat{P}_{xx}[f_k] = \frac{|X_L[f_k]|^2}{f_s L}, \quad f_k = \frac{k f_s}{N}, \quad k = 0, 1, \dots, N-1$$

Donde,

$$X_L[f_k] = \sum_{n=0}^{N-1} x_L[n] e^{-2\pi j k n / N}$$

Para una señal coseno de 200 Hz, añadir ruido y observar su contenido espectral:

```
clear all
Fs=1000;
t=0:1/Fs:.3;
x=cos(2*pi*t*200)+randn(size(t));
Hs=spectrum.periodogram('Hamming');
psd(Hs,x,'Fs',Fs)
% Potencia promedio
Hdsp=psd(Hs,x,'Fs',Fs);
Pow = avgpower(Hdsp)
% Pow = 1.6727
```

