**Ejercicio 1.1**

[t,S1] = senoidal(1,10,0,1,1000);

[t,S2] = senoidal(4,20,0,1,1000);

S=S1+S2;

N=length(S);

X=tdf(S);

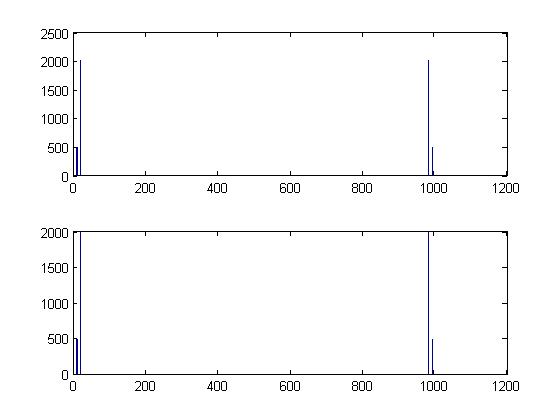
subplot(2,1,1);

bar(abs(X));

subplot(2,1,2);

bar(abs(fft(S)))

%la funcion tdf parece andar bien



**ejercicio 1.2**

[t,S1] = senoidal(1,10,0,1,1000);

[t,S2] = senoidal(4,20,0,1,1000);

S=S1+S2;

N=length(S);

X=fft(S);

sumaS = 0;

sumaX = 0;

for t=1:N

sumaS = sumaS + S(t)\*S(t);

sumaX = sumaX + abs(X(t))\*abs(X(t));

end

sumaX = sumaX/N;

sumaS

sumaX

**ejercicio 1.3**

% %NOTA: recordar que la fft nos retorna primero la parte positiva

% %luego desde la mitad en adelante la parte negativa

% %empenzando desde cero

[t,S1] = senoidal(1,1000,0,1,1000);

[t,S2] = senoidal(4,20,0,1,1000);

S=S1+S2+4;

% N=length(S);

% %1

X=fft(S);

XX=zeros(N);

XX=zeros(N);

XX(1:(N/2))= X(N/2:N-1);

XX((N/2):N-1) = X(1:N/2);

%display(length(X(1:N/2)));

%display(length(X(N/2:N-1)));

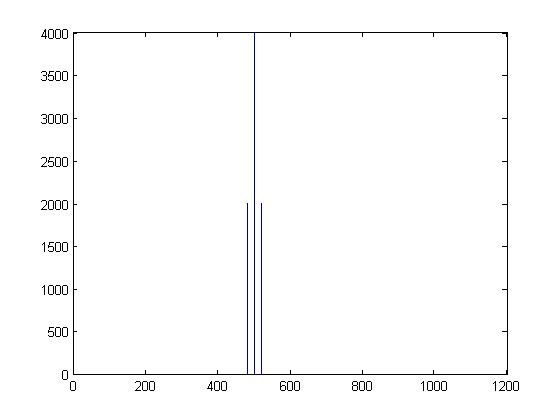
%en el eje de las absisas, el 1, representa la frecuencia cero

%la frecuancia cero--> constantes!

%bar(abs(X),'hist')

%al sumar la cte 4 se agrega una frecuencia cero al principio del espectro

%igual a d\*fm\*cte



%2

[t,S3] = senoidal(1,10,0,1,1000);

[t,S4] = senoidal(4,11,0,1,1000);

Sn2=S3+S4+4;

Y=fft(Sn2);

YY=zeros(N);

YY(1:(N/2))= Y(N/2:N-1);

YY((N/2):N-1) = Y(1:N/2);

subplot(2,1,1)

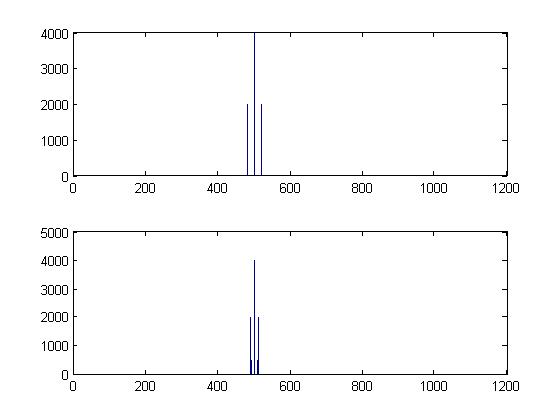
bar(abs(XX))

subplot(2,1,2)

bar(abs(YY))

%las barras se encuentras mas juntas debido a que son frecuencias muy

%similares



%

% %3

[t,S5] = senoidal(1,10,0,1,1000);

[t,S6] = senoidal(4,10.5,0,1,1000);

Sn3=S5+S6;

Z=tdf(Sn3);

ZZ=zeros(N);

ZZ(1:(N/2))= Z(N/2:N-1);

ZZ((N/2):N-1) = Z(1:N/2);

subplot(2,1,1)

bar(abs(XX))

subplot(2,1,2)

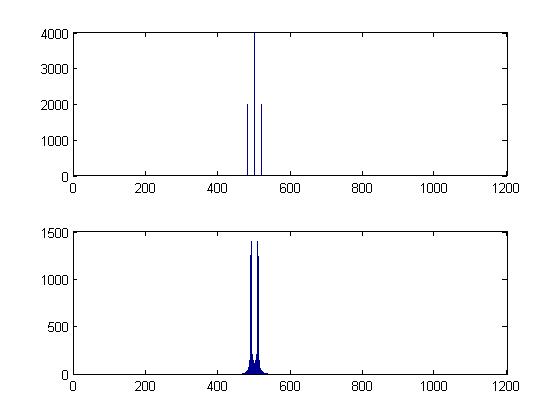
bar(abs(ZZ))

% %es un asco, se pegan y se deforman en la cercania

% %eso pasa porq la diferencia de frecuencias es menor que el paso dado por:

% %dF = fm/N . df>0.5(no da la resolucion), por lo cual aparece fruta jaja :P . la resolución espectral es mala!!--> genera alias->

% %eso se llama aliasing (?)



% %4

[t,S7] = senoidal(1,10,0,0.72,1000);

[t,S8] = senoidal(4,20,0,0.72,1000);

Sn4=S7+S8;

W=tdf(Sn4);

subplot(3,1,1)

bar(abs(X))

subplot(3,1,2)

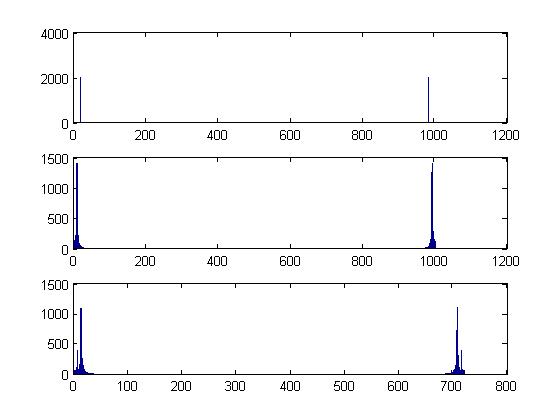
bar(abs(Z))

subplot(3,1,3)

bar(abs(W))

% %pasa algo parecido a lo del punto 3, disminuye la resolucion frecuencial

% %(aumente el deltaF)



**Ejercicio 2**

%señal A=S2

[t,S2]=senoidal(1,2,0,1,100);

N = length(S2);

%señal B=SQ

SQ = square(t\*2\*2\*pi);

%señal C=S4

[t,S4]=senoidal(1,4,0,1,100);

%1

es1 = interno(S2,SQ);%no es .Domino FRECUENCIAL

es2 = interno(S2,S4);%es. Domino FRECUENCIAL

es3 = interno(SQ,S4);%es. Domino FRECUENCIAL

%2

Tes1 = interno(fft(S2),fft(SQ)); %no es.Domino FRECUENCIAL

Tes2 = interno(fft(S2),fft(S4));%es.Domino FRECUENCIAL

Tes3 = interno(fft(SQ),fft(S4));%es.Domino FRECUENCIAL

%3

[t,S35]=senoidal(1,3.5,0,1,100);

es4 = interno(S2,S35);%es

Tes4 = interno(fft(S2),fft(S35));%es Domino FRECUENCIAL

Test44 = interno(S2,S35); % es dominio TEMPORAL

stem(abs(fft(S35)))

%plot(t,S2,t,S4,t,SQ)

**Ejercicio 3**

e1 = zeros(100,1);

e2 = zeros(100,1);

f1 = zeros(100,1);

for t=0:100

e1(t+1) = exp(-(1i\*2\*pi\*1\*t)/100);

e2(t+1) = exp(-(1i\*2\*pi\*2\*t)/100);

f1(t+1) = exp(-(1i\*2\*pi\*1.2\*t)/100);

end

SI = [e1; e1];

NO = [f1; f1];

DUDA = [e1; e2];

subplot(3,2,1)

bar(real(SI))

subplot(3,2,3)

bar(real(NO))

subplot(3,2,5)

bar(real(DUDA))

subplot(3,2,2)

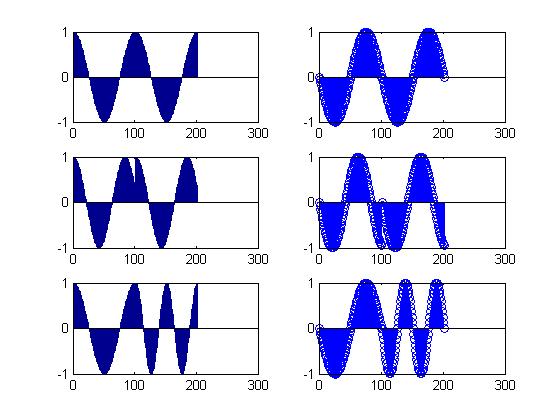
stem(imag(SI))

subplot(3,2,4)

stem(imag(NO))

subplot(3,2,6)

stem(imag(DUDA))



Como la base de Fourier es ortogonal, no cambia las relaciones entre los ángulos, por lo que preserva los ángulos relativos. Entonces, si se verifica ortogonalidad en el tiempo, se debería mantener la ortogonalidad en la frecuencia.

a) senoidal de 2hz y c) senoidal de 3.5hz. En ese caso, en el tiempo dio ortogonal y en la frecuencia no. Pero por qué pasa eso? Pasa porque para la cantidad de muestras que estamos considerando (100) la senoidal discreta de 3.5 Hz no resulta periódica y como Fourier supone periodicidad, aparecen muchas frecuencias que no estaban para hacer el salto entre el fin del período y el principio del siguiente. Entonces aparecerá una frecuencia de 2 Hz (entre otras) y por eso el grado de parecido. Para entenderlo, graficá la FFT de la senoidal de 3.5 hz y comparala con la de 3 hz.

**ejercicio 4**

%1 - genero señal cuadrada

t=0:0.01:0.99;

SQ = square(t\*4\*2\*pi);

%2 - agrego ceros al final de la misma

SQ2 = zeros(1,length(SQ)\*2);

SQ2(1:length(SQ)) = SQ;

%3 - calcculo su transformada

X = fft(SQ2);

N = length(X);

XX=zeros(1,N);

for k=0:N-1

XX(k+1) = X(k+1)\*exp(-(1i\*2\*pi\*k\*10)/N);

end

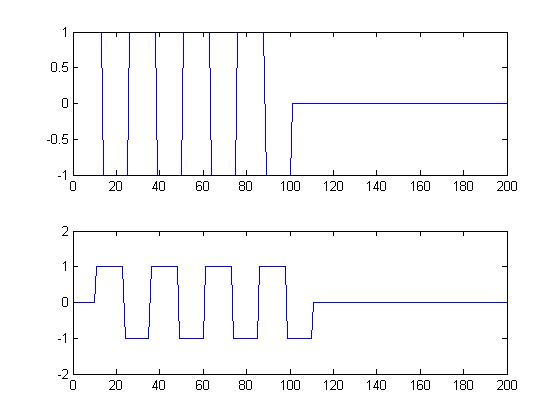
X2 = ifft(XX);

%subplot(2,1,1)

%plot(SQ2)

%subplot(2,1,2)

%plot(real(X2))



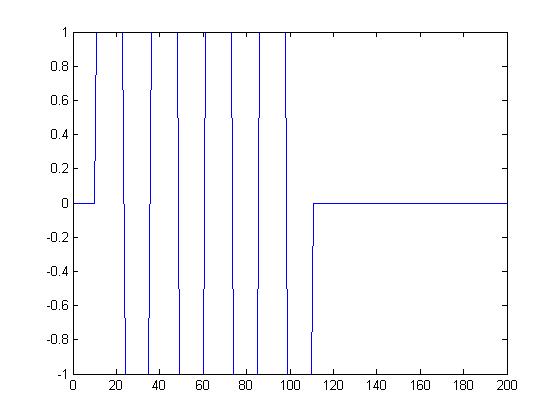
%4 - misma señal pero retardada 10 muestras

SQ10 = square(t\*4\*2\*pi);

SQ10a = zeros(1,2\*length(SQ10));

SQ10a(11:110) = SQ10;

%plot(SQ10a)



%5

Y = fft(SQ10a);

subplot(2,1,1)

resu = Y - XX;

%generada con 10 muestras de retardo

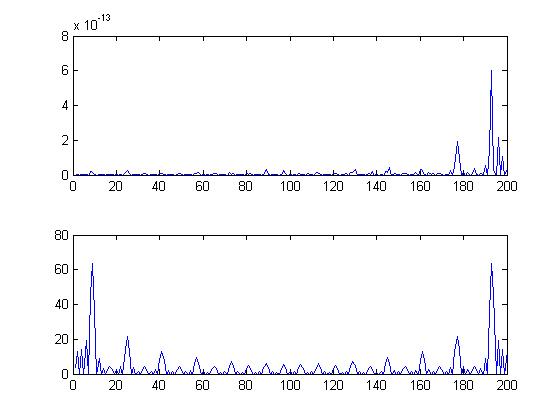
plot(abs(resu))

subplot(2,1,2)

%desplazada mediante propiedades de la transformada

plot(abs(XX))

% conclusion .... MUY BIEN GRACIAS!!!



**Ejercicio 5**

[t,xs] = senoidal(1,10,0,2,100);

xd = delta\_dirac(length(xs));

%xs es la señal senoidal, xd es la delta de dirac

%xsf y xdf son sus respectivas transformadas de Fourier

xsf = fft(xs); %transf senoidal

xdf = fft(xd); %tranformada delta

%xsfCortada = fft(xs); %

%xdfCortada = fft(xd); %

%sin ventana

subplot(2,1,1);

plot(xs);

subplot(2,1,2);

plot(abs(fft(xs)));

%%ventana de hanning

%% con la senoidal

corte = length(xs)/3; %sera la longitud de la ventana

ventana = ventana\_hanning(corte); %calculo la ventana

vect\_ventana=zeros(length(xs),1); %vector de ceros para poner la ventana

vect\_ventana(corte:(corte\*2)-1) = ventana(1:corte); %sumo la ventana al vector anterior.coloco la ventana en la mitad

xsVentana = xs.\* vect\_ventana'; %utilizo la ventana

xsfVentana = fft(xsVentana); %su transformada

% sin ventana

subplot(4,1,1);

plot(xs);

subplot(4,1,2);

plot(abs(fft(xs)));

% con ventana

subplot(4,1,3);

plot(xsVentana);

subplot(4,1,4);

plot(abs(xsfVentana));

%% con la delta de dirac

corte = length(xd)/3; %sera la longitud de la ventana

ventana = ventana\_hanning(corte); %calculo la ventana

vect\_ventana=zeros(length(xd),1); %vector de ceros para poner la gentana

%vect\_ventana(1:corte) = ventana(1:corte); %sumo la ventana al vector

%anterior

%bolazo para tirar el delta en el medio

vect\_ventana(corte:(corte\*2)-1) = ventana(1:corte);

xd(length(xd)/2)=1;

%

xdVentana = xd.\* vect\_ventana; %utilizo la ventana

xdfVentana = fft(xdVentana); %su transformada

% %sin ventana (ACA HACE FRUTA !!! )

% subplot(4,1,1);

% plot(xd);

% subplot(4,1,2);

% plot(abs(fft(xd)));

% %con ventana

% subplot(4,1,3);

% plot(xdVentana);

% subplot(4,1,4);

% plot(abs(xdfVentana));

%%

%%ventana de blackman

%con la senoidal

corte = length(xs)/3; %sera la longitud de la ventana

ventana = ventana\_blackman(corte); %calculo la ventana

vect\_ventana=zeros(length(xs),1); %vector de ceros para poner la gentana

vect\_ventana(corte:(corte\*2)-1) = ventana(1:corte); %sumo la ventana al vector anterior

xsVentana5B = xs.\* vect\_ventana'; %utilizo la ventana

xsfVentanaB = fft(xsVentana); %su transformada

% %sin ventana

% subplot(4,1,1);

% plot(xs);

% subplot(4,1,2);

% plot(abs(fft(xs)));

% %con ventana

% subplot(4,1,3);

% plot(xsVentana);

% subplot(4,1,4);

% plot(abs(xsfVentana));

**Ejercicio 6**

ecg = load('ecg.txt');

eeg = load('eeg.txt');

emg = load('emg.txt');

pres = load('presion.txt');

resp = load('respiracion.txt');

Tecg = fft(ecg);

Teeg = fft(eeg);

Temg = fft(emg);

Tpres = fft(pres);

Tresp = fft(resp);

subplot(5,2,3);stem(ecg);grid on;title('ecg');

subplot(5,2,4);bar(abs(Tecg)); title('Tecg');

subplot(5,2,5);stem(eeg);title('eeg');

subplot(5,2,6);bar(abs(Teeg));title('Teeg');

subplot(5,2,1);stem(emg);title('emg');

subplot(5,2,2);bar(abs(Temg));title('Temg');

subplot(5,2,7);stem(pres);title('pres');

subplot(5,2,8);bar(abs(Tpres));title('Tpres');

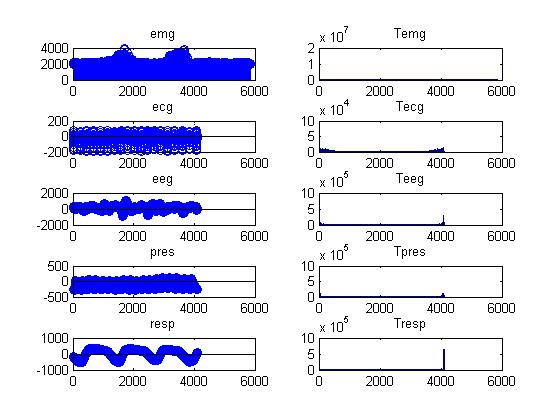
subplot(5,2,9);stem(resp);title('resp');

subplot(5,2,10);bar(abs(Tresp));title('Tresp');

%cual tiene mas ancho de banda de las ultimas... ?? como obtenemos el

%ancho??... ordenamos nada !! MUY BIEN GRACIAS

%Para [señales analógicas](http://es.wikipedia.org/wiki/Se%C3%B1al_anal%C3%B3gica), el **ancho de banda** es la longitud, medida en [Hz](http://es.wikipedia.org/wiki/Hz), del rango de %frecuencias en el que se concentra la mayor parte de la potencia de la señal. Puede ser calculado %a partir de una señal temporal mediante el [análisis de Fourier](http://es.wikipedia.org/wiki/An%C3%A1lisis_de_Fourier). También son llamadas frecuencias %efectivas las pertenecientes a este rango.



**Ejercicio 7🡪 en papel**

**Ejercicio 8. Incompleto**

%1

[t,S27] = senoidal(2,27,0,1,50);

subplot(2,1,1)

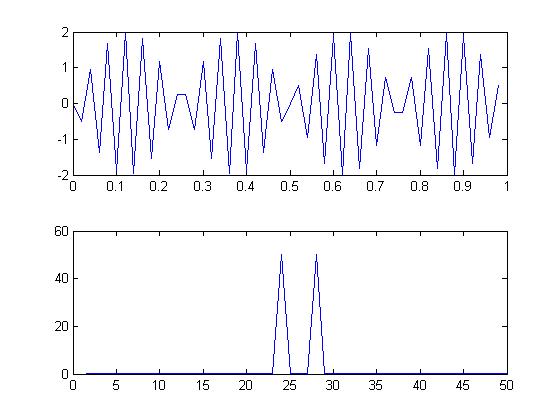
plot(t,S27)

subplot(2,1,2)

plot(abs(fft(S27)))

%las discrepancias se deben a que la fm es menor que 2\*f de la senoidal, es

%decir no se respeta el teorema del muestreo



%2

**Ejercicio 9**

Conclusión : Utilizando la función ‘FFT’ resulta muy sencillo aplicar este método de interpolación con el que

además, como ya se comentó, si la señal original ha sido muestreada cumpliendo el teorema del

muestreo, se asegura que la señal será reconstruida con exactitud. Sin embargo, el inconveniente que

posee el método es que no puede aplicarse en tiempo real (no pueden obtenerse nuevos puntos a

medida que van obteniéndose las muestras de la señal), sino que sólo puede realizarse la interpolación

cuando se dispone de todas las muestras. Esto es lógico ya que para calcular la Transformada se

necesitan todos los puntos.

m1 = load('merval1.txt');

m2 = load('merval2.txt');

Tm1 = fft(m1);

N = length(m1);

X = [Tm1(1:N/2); zeros(N,1); Tm1(N/2+1:N);];

inter = ifft(X).\*2;

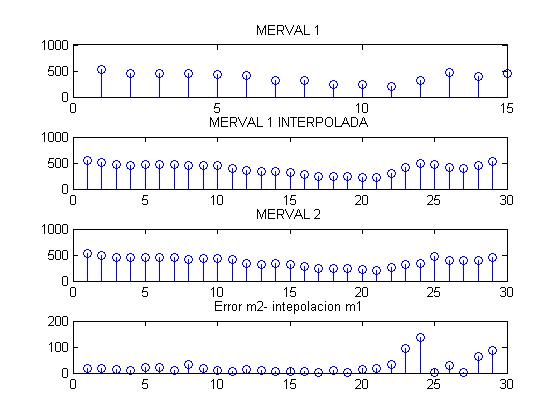
subplot(4,1,1);stem(m1);title('MERVAL 1')

subplot(4,1,2);stem(real(inter));title('MERVAL 1 INTERPOLADA')

subplot(4,1,3);stem(m2);title('MERVAL 2')

subplot(4,1,4);error= m2 - inter;stem(abs(error));title('Error m2- intepolacion m1');

%CHANNNNNN jajajaj.



**Ejercicio 10**

necg = load('necg.txt');

N = length(necg);

Tnecg = fft(necg);

subplot(4,1,1);plot(abs(Tnecg));title('Tnecg- tranformada señal sin filtro');

subplot(4,1,2);plot(abs(necg));title('abs(necg) - señal sin filtro' );

%se aplica "filtro" se colocan ceros en la frecuencias que van desde 40 a 180 Hz

%forma de calculo para colocar los ceros en el intevalo correcto

% N=1024 muestras

%Dt=T=1/fm

% Df=fm/N=1/NT=1/T0

%T0=NT=1024/360=2.84 duracion total

%Df=306/1024=0.3515

% fm/N=1/NT --> fm=1/T --> T=1/fm=1/360

%FORMULITA QUE RELACIONA EL INDICE “k” de la grafica con la frecuencia de la señal

% k\*df=f

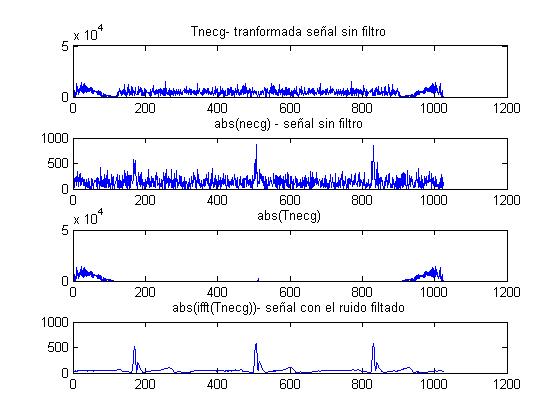
%donde k es el índice perteneciente al vector

Tnecg(113:511) = zeros(511-113+1,1);

Tnecg(N-511:N-113) = zeros(511-113+1,1);

subplot(4,1,3);plot(abs(Tnecg));title('abs(Tnecg)');

subplot(4,1,4);plot(abs(ifft(Tnecg)));title('abs(ifft(Tnecg))- señal con el ruido filtado');

****

**Ejercicio 11 : MAL**

**VENTANAS**

function [w] = ventana\_blackman(N)

w=zeros(N,1);

for i=1 : 1 : N

w(i)=21/50 - 0.5\*cos((2\*pi\*i)/N)+ (2/25)\*cos((4\*pi\*i)/N);

end

function [w] = ventana\_hanning(N)

w=zeros(N,1);

for i=1 : 1 : N

w(i)=0.5 - 0.5\*cos((2\*pi\*i)/N);

end

**furier**

function X = tdf(y)

N=length(y);

X=zeros(N,1);

for k=0:N-1

suma=0;

for ii=0:N-1

suma = suma + y(ii+1)\*exp(-(i\*2\*pi\*k\*ii)/N);

end

X(k+1)=suma;

end

end

function X = tdfi(y)

N=length(y);

X=zeros(N,1);

for k=0:N-1

suma=0;

for ii=0:N-1

suma = suma + y(ii+1)\*exp((1i\*2\*pi\*k\*ii)/N);

end

X(k+1)=suma/N;

end

end