Ejercicio Práctico: Manejo y Visualización de la Transformada Rápida de Fourier (FFT) y la Transformada de Fourier de Tiempo Corto (STFT)

Asignatura: Procesamiento de Señales

Universidad del Rosario - Escuela de Ingeniería, Ciencia y Tecnología

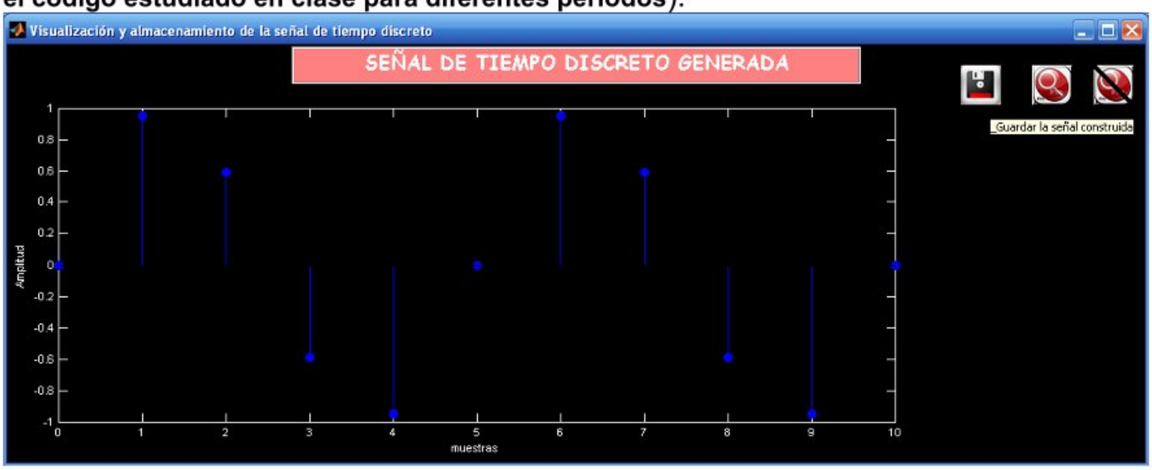
Objetivo:

- Calculo de la Transformada Rápida de Fourier (FFT).
- Calculo de la Transformada de Fourier de Tiempo Corto (STFT)

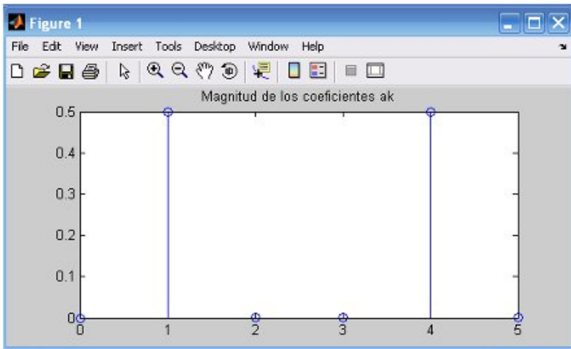
Procedimiento:

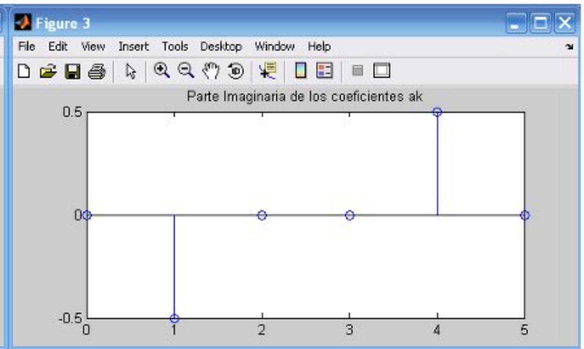
Como punto de partida, recuerde que, para una señal sinusoidal, las componentes de la Serie de Fourier son dos, es decir, cuando k = 1 y cuando k=-1. Esto es con base en lo obtenido en clase cuando se calculó la Transformada de Fourier Discreta para una señal senoidal.
Resumiendo, estas dos componentes aparecen cuando se tiene una señal sinusoidal.

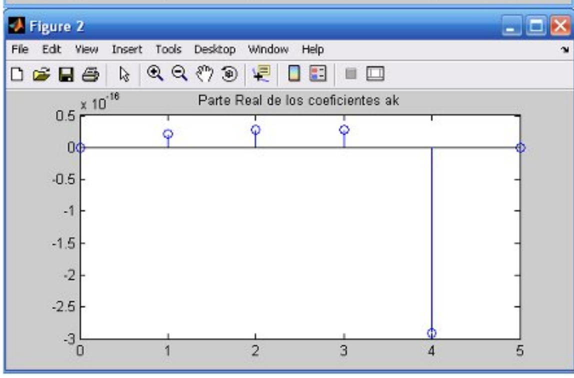
Como segundo aspecto, se parte de una señal senoidal (**$y[n] = Sen(\Omega_0 n)$**), que es de tiempo discreto; además, considere un período de cinco muestras (**N = 5**). A continuación, se muestra la señal de tiempo discreto para dos períodos (10 muestras) (**Sugerencia: utilice el código estudiado en clase para diferentes períodos**).



Se obtienen las componentes de la serie de Fourier,





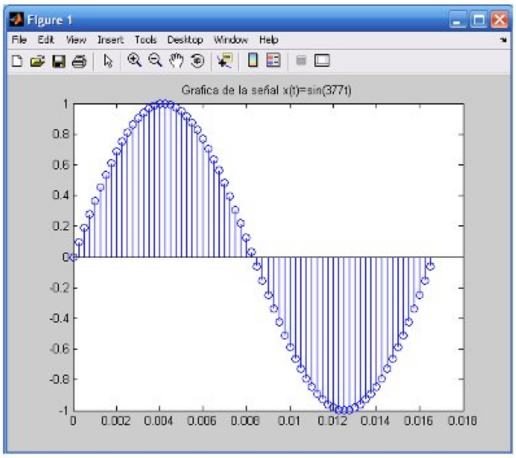


Ahora, se construirá una señal sinusoidal de tiempo continuo y se realizará el muestreo.

Para construir una señal sinusoidal muestreada, se utiliza el siguiente código.

```
function muestreo

%se establece la frecuencia de muestreo
frecsam=4000;

%contrucción del eje de tiempo
tiempo=0:1/frecsam:1/60;

%construcción de la señal
y=sin(377*tiempo);

%visualización de la señal
figure;,stem(tiempo,y)
title('Grafica de la señal x(t)=sin(377t)')
```

Al ejecutar la función se tiene,

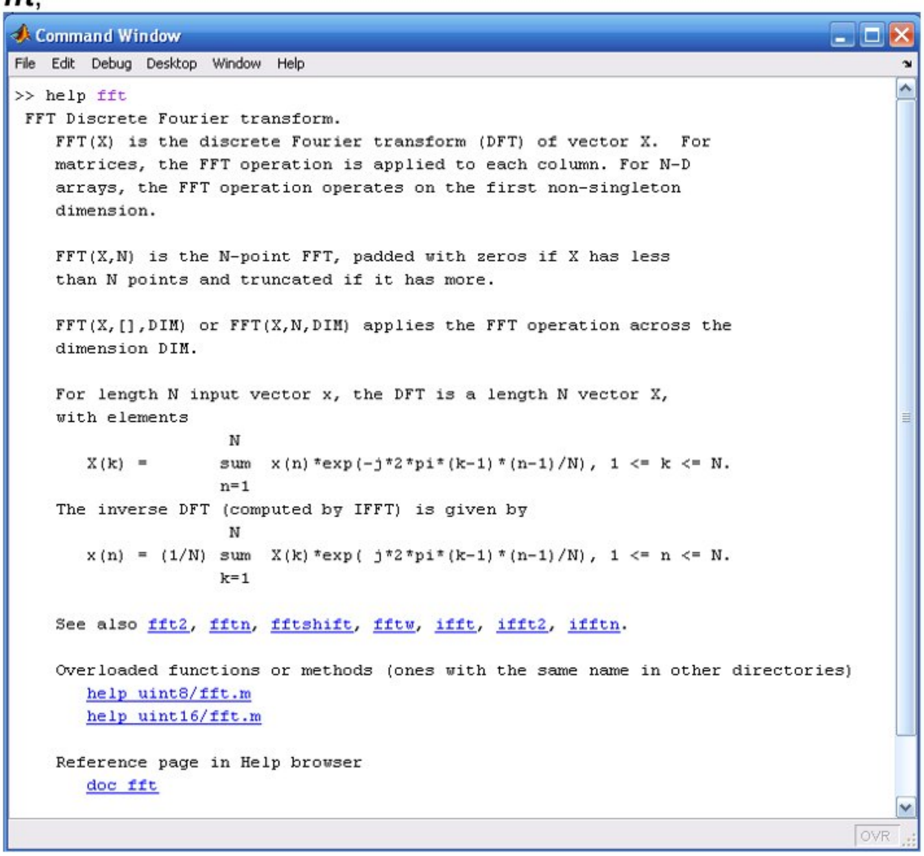


Se observa que la señal tiene una frecuencia de w=377[rad/s], que equivale a un período T = 16.66 [ms], y se utiliza una frecuencia de muestreo de 4000 [Hz]. De acuerdo con la figura, se observa que la señal sinusoidal tiene amplitud 1, lo cual lleva a que las componentes de frecuencia posean una magnitud de 0.5.

Ahora, se obtiene la Transformada Rápida de Fourier (fft) para esta señal, el cual acelera el cálculo de la Transformada Discreta de Fourier cuando la señal tiene una longitud múltiplo de 2^n.

Responda: ¿Esta señal tiene un número de muestras múltiplo de 2^n? Explique el procedimiento que utilizó para determinar si la señal es múltiplo de 2^n.

La ayuda de la función FFT muestra, cuando se escribe en el *prompt* de Matlab ***>>help fft***,



Command Window
File Edit Debug Desktop Window Help

```
>> help fft
 FFT Discrete Fourier transform.
    FFT(X) is the discrete Fourier transform (DFT) of vector X.  For
    matrices, the FFT operation is applied to each column. For N-D
    arrays, the FFT operation operates on the first non-singleton
    dimension.

    FFT(X,N) is the N-point FFT, padded with zeros if X has less
    than N points and truncated if it has more.

    FFT(X,[],DIM) or FFT(X,N,DIM) applies the FFT operation across the
    dimension DIM.

    For length N input vector x, the DFT is a length N vector X,
    with elements
                     N
       X(k) =       sum  x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N.
                    n=1
    The inverse DFT (computed by IFFT) is given by
                     N
       x(n) = (1/N) sum  X(k)*exp( j*2*pi*(k-1)*(n-1)/N), 1 <= n <= N.
                    k=1

    See also fft2, fftn, fftshift, fftw, ifft, ifft2, ifftn.

    Overloaded functions or methods (ones with the same name in other directories)
       help uint8/fft.m
       help uint16/fft.m

    Reference page in Help browser
       doc fft
```

OVR

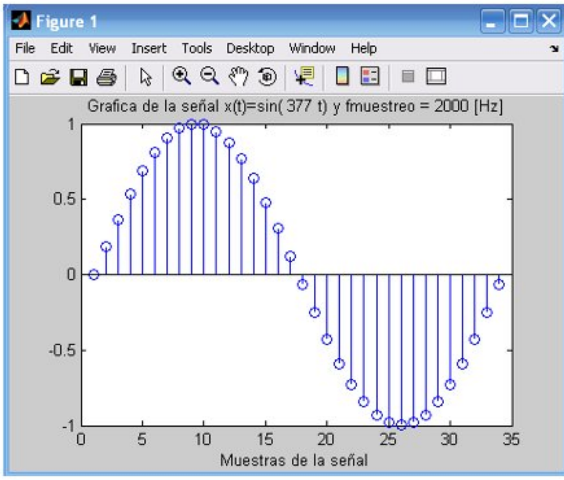
La función tiene como variable de entrada la señal discretizada y como variable de salida las componentes de la Transformada Discreta de Fourier. (**Sugerencia: utilice el código analizado y estudiado en clase para diferentes períodos y que, además, utiliza la función FFTSHIFT de Matlab**).

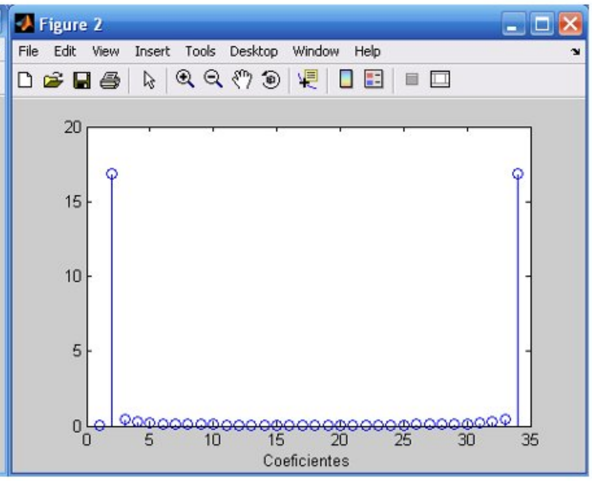
A continuación, se utiliza la función "**fft.m**" para una señal senoidal, lo cual requiere construir el eje de frecuencia con el fin de dibujar el resultado de la Transformada Discreta de Fourier con los valores de frecuencia.

El código es,

```
function muestreo_sinusoidal_fft

%se establece la frecuencia de la señal sinusoidal [rad/s]
w=377;
%se establece la frecuencia de muestreo [Hz]
frecsam=2000;

%contrucción del eje de tiempo
tiempo=0:1/frecsam:1/60;

%construcción de la señal
y=sin(377*tiempo);

%visualización de la señal
figure;,stem(y)
title(['Grafica de la señal x(t)=sin( ' num2str(w),...
    ' t) y fmuestreo = ' num2str(frecsam) ' [Hz]'])
xlabel('Muestras de la señal')

%trasnformada discreta de fourier para el tiempo de muestreo seleccionado
fourier=fft(y);%algoritmo fft
figure;,stem(abs(fourier))%se dibuja la magnitud
xlabel('Coeficientes')
```

Se ejecuta la función "**muestreo_sinusoidal_fft.m**",





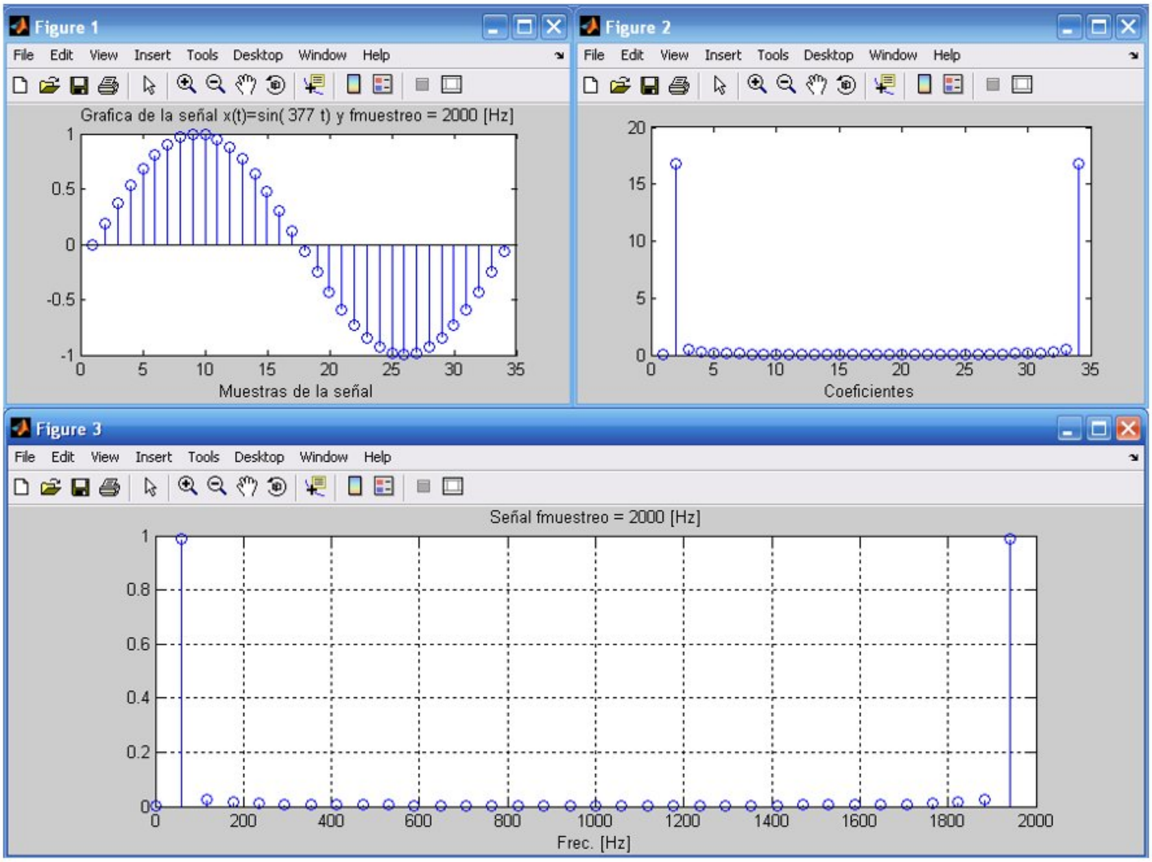
Es importante observar y considerar los siguientes aspectos:

- La señal discretizada posee 34 muestras y tiene una amplitud de 1.
- El número de componentes de la Transformada Discreta de Fourier son 34, que es igual a la longitud de la señal.
- Como es una señal sinusoidal solo aparecen dos componentes predominantes con una magnitud cercana a los 17. Esto quiere decir que el algoritmo "`fft.m`" de Matlab multiplica el resultado por una escala que es igual al número de muestras que conforman la señal x[n]. Por consiguiente, es necesario dividir todas las componentes obtenidas de la Transformada de Fourier por el número de muestras que componen la señal x[n] para que las componentes predominantes presenten una magnitud de ½.
- Es necesario construir con base en toda la información anterior el eje de frecuencia.

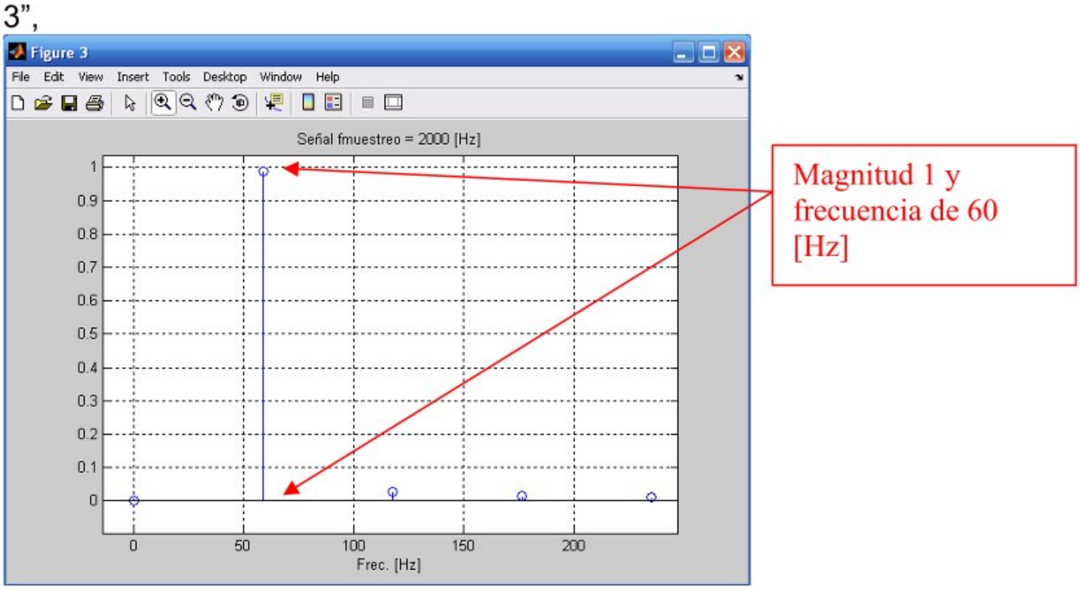
El siguiente código tiene presente todos los aspectos anteriores,

```
function muestreo_sinusoidal_fft

%se establece la frecuencia de la señal sinusoidal [rad/s]
w=377;,f=w/(2*pi);
%se establece la frecuencia de muestreo [Hz]
frecsam=2000;

%contrucción del eje de tiempo
tiempo=0:1/frecsam:1/60;

%construcción de la señal
y=sin(377*tiempo);

%visualización de la señal
figure;,stem(y)
title(['Grafica de la señal x(t)=sin( ' num2str(w),...
    ' t) y fmuestreo = ' num2str(frecsam) ' [Hz]'])
xlabel('Muestras de la señal')

%trasnformada discreta de fourier para el tiempo de muestreo seleccionado
fourier=fft(y);%algoritmo fft
figure;,stem(abs(fourier))%se dibuja la magnitud
xlabel('Coeficientes')

%transformada de fourier con magnitud cambiada y construcción
%del vector de frecuencia.
l_fou=length(fourier);%longitud del vector
abs_fou=abs(fourier)/(l_fou/2);%valor absoluto de la FFT con
%cambio de escala
delta_frec=frecsam/l_fou;%delta de frecuencia para construir el
%eje de frecuencia
eje_frec=[0:1:l_fou-1]*delta_frec;%eje de frecuencia

figure;,stem(eje_frec,abs_fou),grid
title(['Señal fmuestreo = ' num2str(frecsam) ' [Hz]']),xlabel('Frec. [Hz]')
```

El resultado de ejecutar la función se muestra a continuación, en donde la gráfica nombrada “Figure 3”.



Como la señal tiene frecuencia f = 60 [Hz], esta componente debe aparecer en la gráfica de la Transformada Discreta de Fourier. Se hace un acercamiento en la ventana "Figure 3",



Responda: En el código anterior modifique las líneas necesarias para que usted pueda observar las componentes Fourier con valor de ½.

1. Se construye una señal sinusoidal con diferentes tiempos de muestreo.

Considere la siguiente señal sinusoidal,

$$x(t) = sen(w * t)$$

$$w = 2\pi f = 377 \left[{}^{rad}\!/_{sg} \right]$$

$$f = 60 \left[Hz \right]$$

Es necesario realizar un muestreo sobre la señal x(t) con el fin de obtener x[n], para ello, se realiza el muestreo de acuerdo al teorema del muestreo,

$$f_m \geq 2f$$

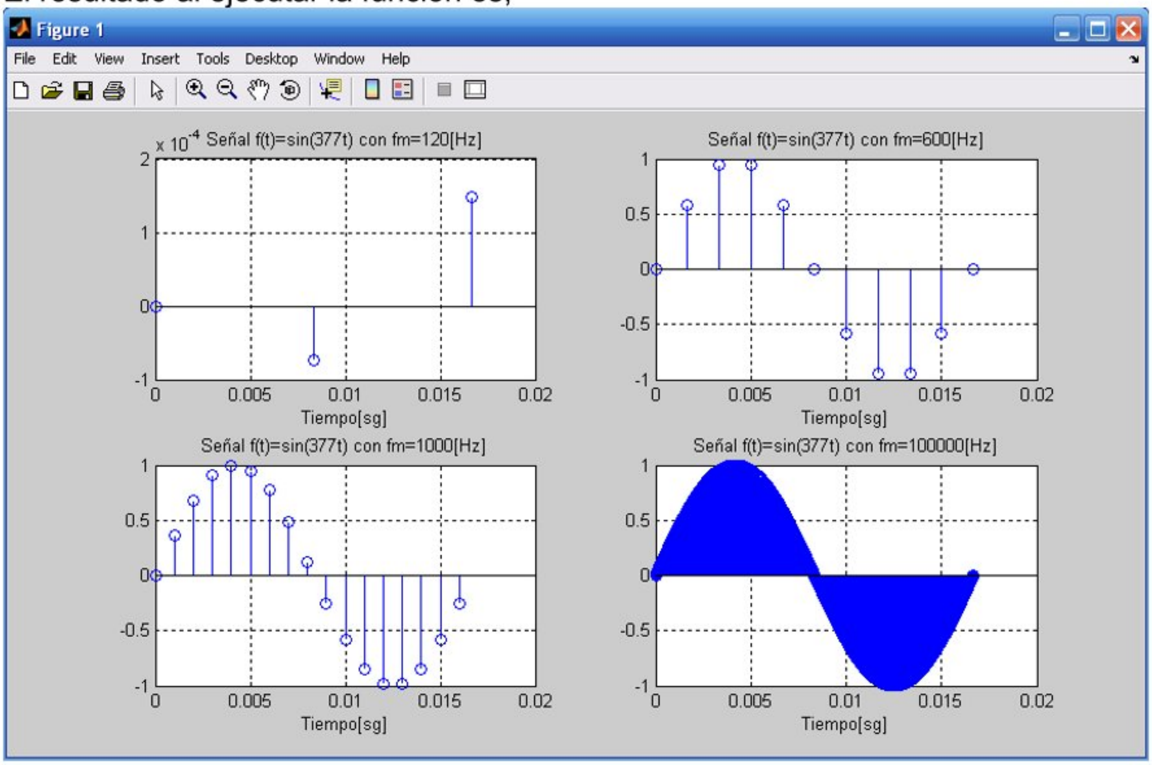
La señal se muestreará utilizando diferentes tiempos de muestreo, con el fin de observar el efecto en la Transformada Rápida de Fourier (fft). Para ello se utilizará la función "**`fft.m`**" de Matlab. Además, se utilizará la herramienta de Matlab "vector tipo estructura", lo cual facilitará la construcción/manipulación de variables.

- Ingrese el siguiente código en el editor de Matlab, tenga en cuenta que es una función en Matlab ya que en el encabezado se ubica "`function`".

```
function muestreo_dibujo

%se establece la frecuencia de muestreo
frecsam=[120 600 1000 100000];

%contrucción del eje de tiempo
tiempo.caso1=0:1/frecsam(1):1/60;
tiempo.caso2=0:1/frecsam(2):1/60;
tiempo.caso3=0:1/frecsam(3):1/60;
tiempo.caso4=0:1/frecsam(4):1/60;

%construcción de la señal
y.caso1=sin(377*tiempo.caso1);
y.caso2=sin(377*tiempo.caso2);
y.caso3=sin(377*tiempo.caso3);
y.caso4=sin(377*tiempo.caso4);

% se dibujan los resultados
subplot(2,2,1), stem(tiempo.caso1,y.caso1),grid
title('Señal f(t)=sin(377t) con fm=120[Hz]'),xlabel('Tiempo[sg]')
subplot(2,2,3), stem(tiempo.caso3,y.caso3),grid
title('Señal f(t)=sin(377t) con fm=1000[Hz]'), xlabel('Tiempo[sg]')
subplot(2,2,2), stem(tiempo.caso2,y.caso2),grid
```

```
title('Señal f(t)=sin(377t) con fm=600[Hz]'), xlabel('Tiempo[sg]')
subplot(2,2,4), stem(tiempo.caso4,y.caso4),grid
title('Señal f(t)=sin(377t) con fm=100000[Hz]'), xlabel('Tiempo[sg]')
```

El resultado al ejecutar la función es,



Responda: De acuerdo con los resultados visualizados, ¿Cómo es el eje x – segundos/muestras? ¿Qué cambio realiza al código para observar muestras en lugar de segundos?

2. **Se aplica el algoritmo de la FFT para la señal a diferentes tiempos de muestreo, para ello, se muestra a continuación el código que debe ubicarse en el editor de Matlab,**

- Se ubica el encabezado de la función y se establecen las 4 frecuencias de muestreo a utilizar.

```
function muestreo_fft_dibujo

%se toma como base una señal senoidal de 60 Hz
%se establece la frecuencia de muestreo
frecsam=[120 600 1000 100000];
```

- Se construyen los 4 ejes de tiempo para cada uno de las frecuencias de muestreo.

```
%contrucción del eje de tiempo
```

```
tiempo.caso1=0:1/frecsam(1):1/60;%para fm=120Hz
tiempo.caso2=0:1/frecsam(2):1/60;%para fm=600Hz
tiempo.caso3=0:1/frecsam(3):1/60;%para fm=1000Hz
tiempo.caso4=0:1/frecsam(4):1/60;%para fm=100000Hz
```

- Se construyen las 4 señales con cada uno de los ejes de tiempos construidos.

```
%construcción de las señales
y.caso1=sin(377*tiempo.caso1);%para fm=120Hz
y.caso2=sin(377*tiempo.caso2);%para fm=600Hz
y.caso3=sin(377*tiempo.caso3);%para fm=1000Hz
y.caso4=sin(377*tiempo.caso4);%para fm=100000Hz
```

- Se aplica el algoritmo FFT para los diferentes tiempos de muestreo.

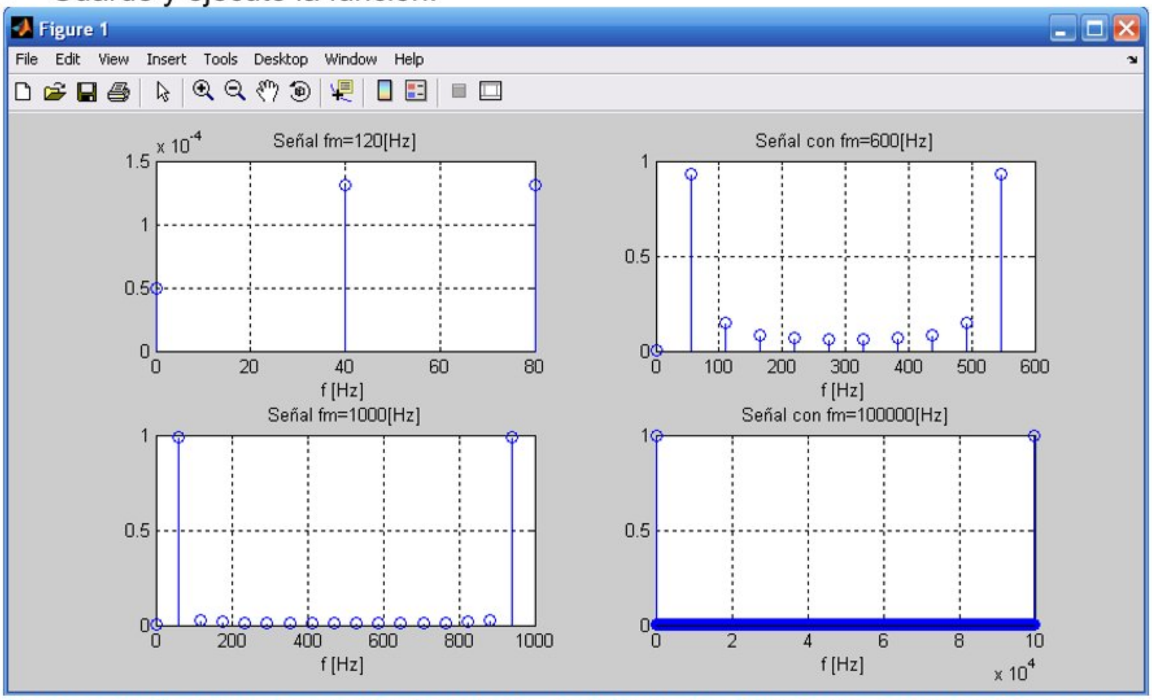
```
%trasnformada de fourier para los diferentes tiempo de muestreo
%para fm=120Hz
fourier.caso1=fft(y.caso1);%algoritmo fft
l_fou.caso1=length(fourier.caso1);%longitud del vector
abs_fou.caso1=abs(fourier.caso1)/(l_fou.caso1/2);%valor absoluto de la FFT
delta_frec.caso1=frecsam(1)/l_fou.caso1;%delta de frecuencia
eje_frec.caso1=[0:1:l_fou.caso1-1]*delta_frec.caso1;%eje de frecuencia

%para fm=600Hz
fourier.caso2=fft(y.caso2);%algoritmo fft
l_fou.caso2=length(fourier.caso2);%longitud del vector
abs_fou.caso2=abs(fourier.caso2)/(l_fou.caso2/2);%valor absoluto de la FFT
delta_frec.caso2=frecsam(2)/l_fou.caso2;%delta de frecuencia
eje_frec.caso2=[0:1:l_fou.caso2-1]*delta_frec.caso2;%eje de frecuencia

%para fm=1000Hz
fourier.caso3=fft(y.caso3);%algoritmo fft
l_fou.caso3=length(fourier.caso3);%longitud del vector
abs_fou.caso3=abs(fourier.caso3)/(l_fou.caso3/2);%valor absoluto de la FFT
delta_frec.caso3=frecsam(3)/l_fou.caso3;%delta de frecuencia
eje_frec.caso3=[0:1:l_fou.caso3-1]*delta_frec.caso3;%eje de frecuencia

%para fm=100000Hz
fourier.caso4=fft(y.caso4);%algoritmo fft
l_fou.caso4=length(fourier.caso4);%longitud del vector
abs_fou.caso4=abs(fourier.caso4)/(l_fou.caso4/2);%valor absoluto de la FFT
delta_frec.caso4=frecsam(4)/l_fou.caso4;%delta de frecuencia
eje_frec.caso4=[0:1:l_fou.caso4-1]*delta_frec.caso4;%eje de frecuencia
```

- Visualización de los resultados.

```
% se dibujan los resultados
subplot(2,2,1), stem(eje_frec.caso1,abs_fou.caso1),grid
title('Señal fm=120[Hz]'),xlabel('f [Hz]')

subplot(2,2,3), stem(eje_frec.caso3,abs_fou.caso3),grid
title('Señal fm=1000[Hz]'), xlabel('f [Hz]')
```

```
subplot(2,2,2), stem(eje_frec.caso2,abs_fou.caso2),grid
title('Señal con fm=600[Hz]'), xlabel('f [Hz]')

subplot(2,2,4), stem(eje_frec.caso4,abs_fou.caso4),grid
title('Señal con fm=100000[Hz]'), xlabel('f [Hz]')
```

- Guarde y ejecute la función.



Responda: De acuerdo con lo observado, modifique el código para observar las componentes de Fourier con valor de ½ que es lo que debe observarse; además, ¿A partir de qué valor de frecuencia se pueden observar las componentes de Fourier que deben ser? ¿Todas las componentes de Fourier que parecen ser de valor cero, realmente son cero?

3. Se construye una señal estacionaria en el tiempo, con 4 componentes armónicas.

- Se ubica el encabezado y se definen: numero de periodos a visualizar, frecuencia fundamental, frecuencia de muestreo, eje de tiempo, delta de frecuencia y vector de frecuencia.

```
function armonicas_fft

%Se construye el eje de tiempo
k=4;%número de periodos a visualizar
w=377;%frecuencia fundamental
f=377/(2*pi);%frecuencia en Hz
fm=1000*f;%frecuencia de muestreo
t=0:1/fm:k*(2*pi)/377;%eje de tiempo
ll=length(t);
```

```
factorr=ll/2
delta=(fm/(2*factorr));%delta de frecuencias
freq_vector=[0:1:length(t)-1]*delta;%vector de frecuencias
```

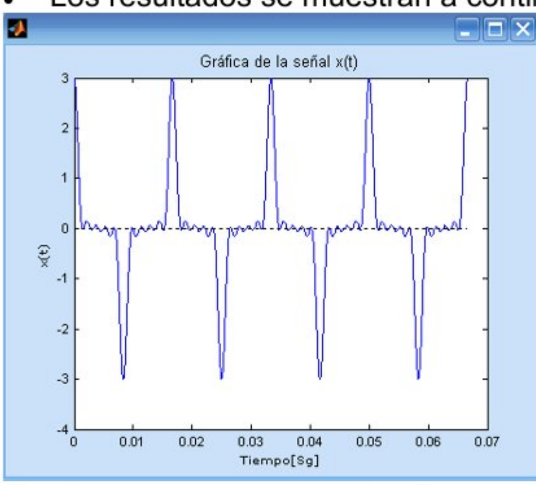
- Se construye la señal a visualizar con las 5 componentes armónicas.

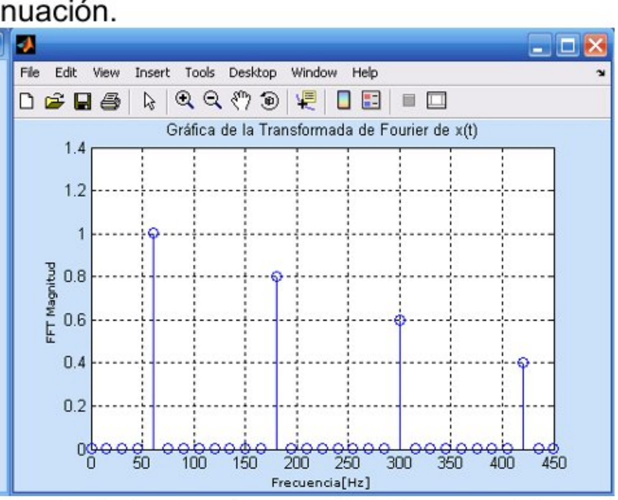
```
wo=377;%frecuencia fundamental
y=0;
kk=1;
for ii=1:2:9,%cuatro componentes armónicas
    y=kk*cos(ii*wo*t)+y;%suma de las componentes armónicas
    kk=kk-0.2;
end

figure('color',[202 225 249]/255,'menubar','none','numbertitle','off')
plot(t,y)
xlabel('Tiempo[Sg]','fontname','verdana','fontsize',8),
ylabel('x(t)','fontname','verdana','fontsize',8)
title('Gráfica de la señal x(t)')
gg=gca;
set(gg,'fontsize',8)
line([t(1) t(length(t))],[0 0],'color',[0 0 0],'linestyle',':')
pause
```

- Se calcula la FFT y se visualiza el resultado.

```
ftt=fft(y);%transformada rápida de fourier
figure('color',[202 225 249]/255,'numbertitle','off');
stem(freq_vector(1:31),abs(ftt(1:31))/factorr),grid
xlabel('Frecuencia[Hz]','fontname','verdana','fontsize',8)
ylabel('FFT Magnitud','fontname','verdana','fontsize',8)
title('Gráfica de la Transformada de Fourier de x(t)')
```

- Guarde y ejecute el programa.
- Los resultados se muestran a continuación.



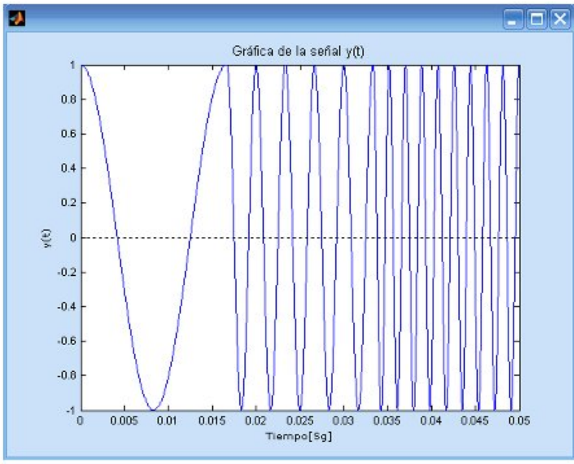


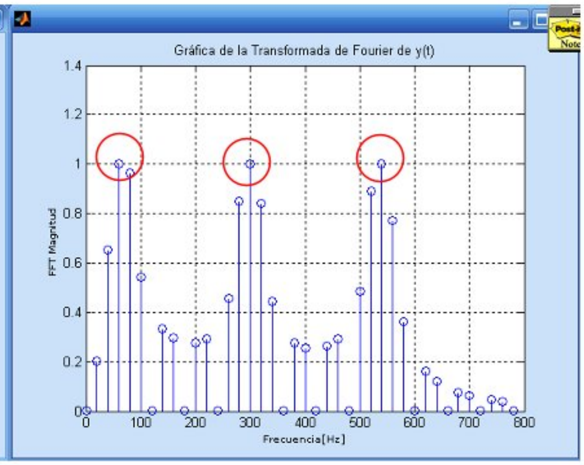
Se observa que aparecen las frecuencias de todas las 5 componentes armónicas. Realice un acercamiento y compruebe que aparecen las frecuencias.

4. Se construye una señal no estacionaria en el tiempo, con 3 componentes armónicas.

```
function armo_no_est_fft

close all%cierra todas las ventanas abiertas
fm=1000000;%frecuencia de muestreo
t=0:(1/fm):(2*pi)/377;%eje de tiempo
ll=length(t);
factorr=ll/2;
wo=377;%frecuencias fundamental
y=[];
for ii=1:4:9,%3 componentes armónicas
    y=[y cos(ii*wo*t)];%una señal seguida de otra
end
tt=length(y);
factorrr=tt/2;
delta=(1/(1/fm))/(2*factorrr);%delta de frecuencias en Hz
tiempo=(0:1:tt-1)*(1/fm);%eje de tiempo

ftt=fft(y);%calculo de la fft de Matlab
n=0:tt-1;
nn=n*round(delta);%eje de frecuencias en Hz

%Visualización de la señal en el tiempo
figure('color',[202 225 249]/255,'menubar','none','numbertitle','off')
plot(tiempo,y)
xlabel('Tiempo[Sg]','fontname','verdana','fontsize',8)
ylabel('y(t)','fontname','verdana','fontsize',8)
title('Gráfica de la señal y(t)')
gg=gca;
set(gg,'fontsize',8)
line([tiempo(1) tiempo(length(tiempo))],[0 0],'color',[0 0 0],'linestyle',':')

%Visualización de componentes
figure('color',[202 225 249]/255,'menubar','none','numbertitle','off');
stem(nn(1:40),abs(ftt(1:40))/factorr),grid
xlabel('Frecuencia[Hz]','fontname','verdana','fontsize',8)
ylabel('FFT Magnitud','fontname','verdana','fontsize',8)
```

- Se guarda y ejecuta la función en el editor de Matlab.





Se observa que se generan las frecuencias presentes en la señal pero no existe información donde se encuentran las frecuencias.

5. Se construye una señal no estacionaria en el tiempo, con 3 componentes armónicas y se utiliza una ventana de Gauss.

Ubique el siguiente código en el editor de Matlab

```
function windowed_no_est_gauss_fft

close all
paso=0.000001;,t=0:paso:(2*pi)/377;,ll=length(t);,factorr=ll/2;
%inicaliza valores
wo=377;,y=[];
for ii=1:4:9,
    y=[y cos(ii*wo*t)];
end
%calculo de la fft de la señal sin ventaneo
tt=length(y);
factorrr=tt/2;,delta=(1/paso)/(2*factorrr);
tiempo=(0:1:tt-1)*paso;%eje de tiempo
ftt=fft(y);
n=0:tt-1;
nn=n*round(delta);%vector de frecuencias
figure('color',[202 225 249]/255,'menubar','none','numbertitle','off');
stem(nn(1:40),abs(ftt(1:40))/factorr),grid
xlabel('Frecuencia[Hz]','fontname','verdana','fontsize',8)
ylabel('FFT Magnitud','fontname','verdana','fontsize',8)
title('Gráfica de la Transformada de Fourier de y(t)')
%se define una ventana de gauss
x=0:1:length(y)-1;
miu=20000;
sigma=5000;
muestras=length(y);
windowgauss=(1)*exp(-0.5*(((x-miu)/sigma).^2));%construcción Gaussiana
figure('color',[202 225 249]/255,'menubar','none','numbertitle','off')
```

```
plot(tiempo,y,'b',tiempo,windowgauss,'r')
xlabel('Tiempo[Sg]','fontname','verdana','fontsize',8)
ylabel('y(t)','fontname','verdana','fontsize',8)
title('Gráfica de la señal y(t)')
gg=gca;
set(gg,'fontsize',8),line([tiempo(1) tiempo(length(tiempo))],[0 0],'color',[0 0
0],'linestyle',':')
%se obtiene la señal ventaneada
ymodf=windowgauss.*y;%señal ventaneada
figure('color',[202 225 249]/255,'menubar','none','numbertitle','off')
plot(tiempo,ymodf)
xlabel('Tiempo[Sg]','fontname','verdana','fontsize',8)
ylabel('y(t)','fontname','verdana','fontsize',8)
title('Gráfica de la señal y(t) ventaneada (Windowed)')
gg=gca;,set(gg,'fontsize',8),line([tiempo(1) tiempo(length(tiempo))],[0
0],'color',[0 0 0],'linestyle',':')
ymodffft=fft(ymodf);%FFT de señal ventaneada
figure('color',[202 225 249]/255,'menubar','none','numbertitle','off');
stem(nn(1:40),abs(ymodffft(1:40))/factorr),grid
xlabel('Frecuencia[Hz]','fontname','verdana','fontsize',8)
ylabel('FFT Magnitud','fontname','verdana','fontsize',8)
title('Gráfica de la Transformada de Fourier de y(t) ventaneada (Windowed)')
```