

Informe 2 de Matlab

David Ricardo Martínez Hernández Código: 261931
 Juan David Ramos López Código: 261667

I. DESCRIPCIÓN DEL PROGRAMA

Se genero la señal de entrada de la fig 1, se multiplico en tiempo con una señal $\cos(2\pi fx)$ donde:

f es una frecuencia de 40 KHz

x es el vector o el intervalo de frecuencia que va desde $-1\ ms$ hasta $1\ ms$,

como se muestra en la fig 2.

Para que el programa funcionara la frecuencia de muestreo de la señal de entrada es mayor que la frecuencia del coseno para que se obtenga la señal deseada,

```
f=40000;%frecuencia del coseno
fm=41000;%frecuencia de muestreo
x=-1/1000:1/(2*pi*fm):1/1000;
```

si no se obtendrá un coseno con diferentes amplitudes, si se hace de esa manera no se obtendrá una correcta modulación y demodulación.

El parámetro M se escogió de 1000, debido a que la señal tiene como mínimo 80 muestras en tiempo y tendrá un máximo de 360.

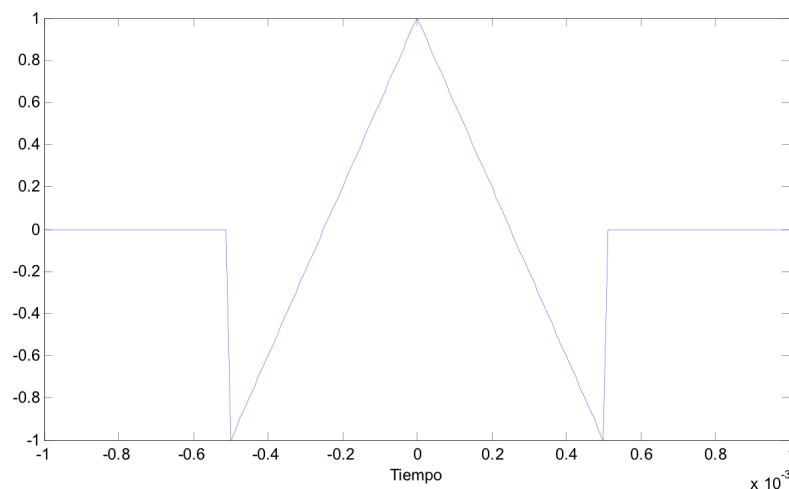


Fig. 1: Señal de entrada

```
y=funcion(x);
y1=cos(2*pi*f*x);
y2=y.*y1;
YX2=fftshift((1/(2*pi*fm))*(fft(y2,M)));
```

donde

y es la señal de entrada

y_1 es la señal coseno

y_2 es la multiplicación de y con y_1

YX_2 es la transformada de Fourier de la señal y_2

Después se procedió a multiplicar nuevamente la señal de entrada figura 1 con el coseno dando como resultado la figura 3

```
y3=y.*y1.^2
YX3=fftshift(1/(2*pi*fm)*(fft(y3,M)))
```

donde

y_3 es la multiplicación de $y * y_1^2$

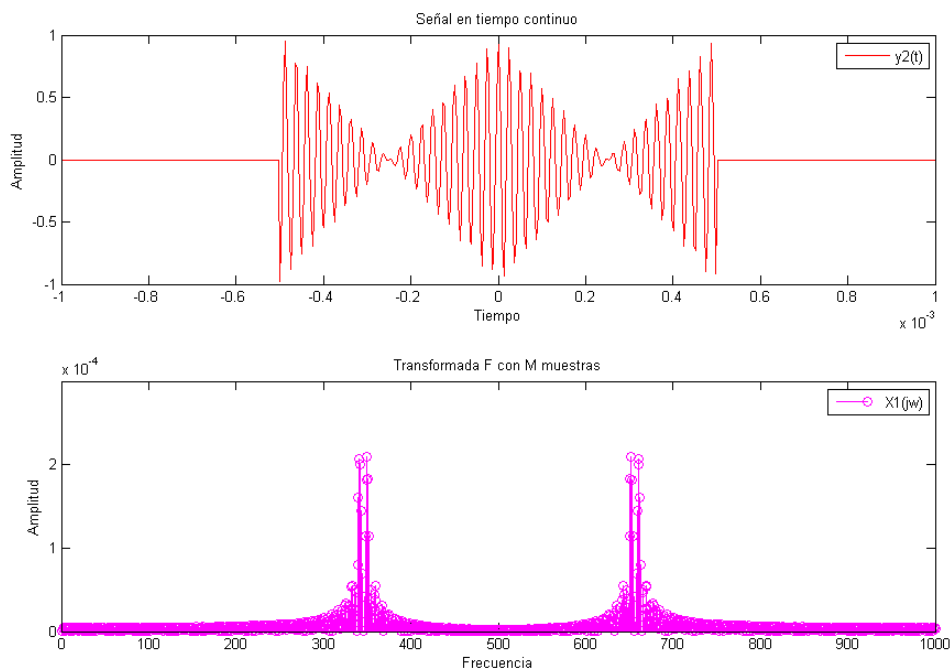


Fig. 2: Señal de entrada multiplicada por $\cos(2\pi fx)$ en tiempo y en frecuencia

YX_3 es la transformada de Fourier de la señal y_3

Como la transformada de $y_1 * \cos^2(2\pi fx)$ tiene tres “impulsos” iguales pero con diferente magnitud y corridas en frecuencia se debe aplicar un filtro pasa-bajos para que solo se obtenga el “impulso” del medio, como se muestra en la figura 4

```
filtro=ones(1,M);
for i=1:M
if (i<M/2-200 || i>M/2+200)
filtro(i)=0;
end
end
y5=2*filtro.*YX3;
```

Para este filtro se utilizó una matriz de 1's para que al momento de ser filtrada con el `for i=1:M` y con el `if (i<M/2-200 || i>M/2+200)` quedara la señal original en el intervalo deseado, dando como resultado la fig 4, en el condicional el ancho del filtro puede ser modificado, para volverlo mas fino, cambiando el valor de 200 por cualquier otro número.

La señal filtrada se encuentra en la figura 5

```
X=ifft(y5,M/2)/(1/(2*pi*fm)); %transformada inversa filtrada
xreal=-1/1000:2/(500*M/2-1):3/1000;
```

donde

X es la Transformada inversa de la señal filtrada (y_5)

x_{real} es el vector para el ajuste de los índices de la transformada inversa

Los índices fueron calculados de la siguiente manera.

- Como la ifft de y_5 tiene un total de 500 muestras, la señal original se encuentra entre la muestra 1 y la 250, es decir en ese intervalo se encuentra la señal original.
- Como la señal de entrada tiene 516 muestras en su totalidad, desde -1 ms hasta 1 ms , la señal y_5 tiene un intervalo de tiempo desde -1 ms hasta 3 ms , es decir 2 veces más de tiempo, pero la duración de la señal sigue siendo de

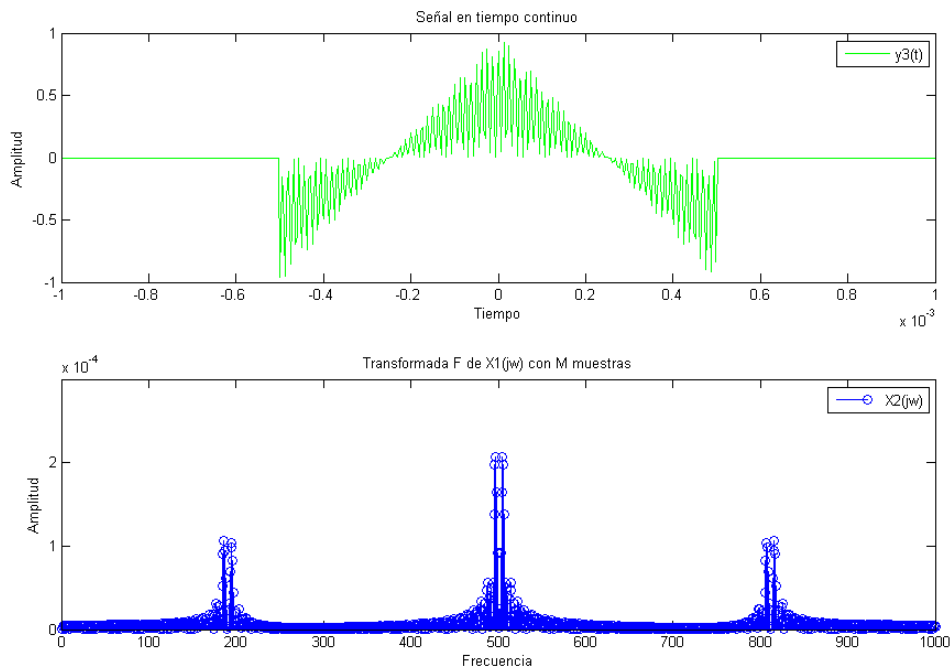


Fig. 3: Señal de entrada multiplicada por $\cos^2(2\pi fx)$ en tiempo y en frecuencia

2 ms. Como el tiempo total es de 4 ms y el tamaño total de la ifft es de 500 muestras, se calculo con una regla de 3 simple, es decir:

$$\frac{4 * 10^{-3}}{\frac{M}{2} - 1}$$

El denominador $\frac{M}{2} - 1$ se le resta 1 debido a que se debe tomar el total de la muestra menos 1 para obtener el valor deseado.

II. CONCLUSIONES

- La frecuencia de muestreo debe ser mayor a la mayor de las frecuencias establecidas en el problema para que no hallan errores al momento de calcular las transformadas de Fourier.
- Para poder demodular la señal es necesario hacerlo con la mitad de las muestras para obtener la señal original.
- MatLab es una gran herramienta para ayudarnos a calcular la modulación y demodulación de señales y aproximarlas a modelos muy reales de una manera muy sencilla y eficaz.

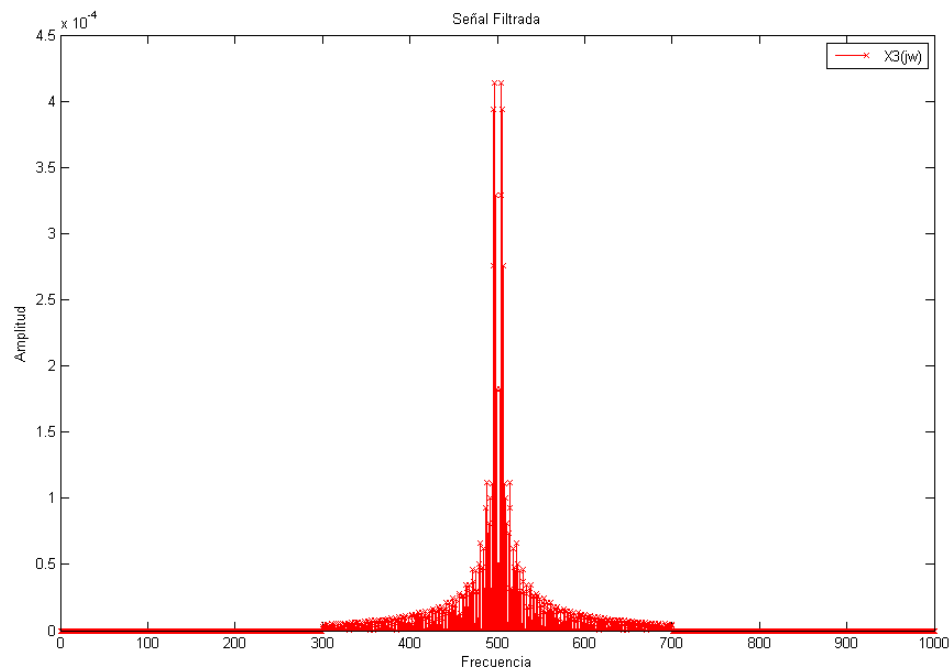
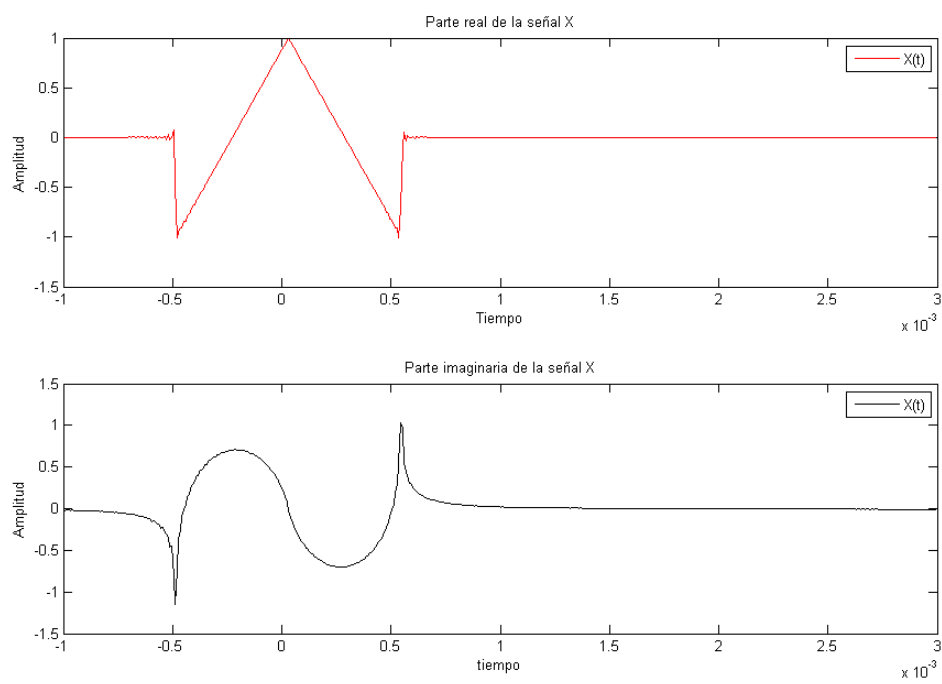


Fig. 4: Señal filtrada

Fig. 5: Salida de la transformada inversa de la función y_5 , parte real y parte imaginaria