|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | |
| **ASIGNATURA: Fundamentos para el Análisis de Señales** | | | | | | | | | | | | | |
| **CODIGO:** 950522 | | | | | | **CURSO:** Q - 3051 | | | | | | **CICLO LECTIVO:** 2019 | |
|  | | | | | | | |  | | | | | |
| **DOCENTE:** Dr. Walter Edgardo LEGNANI | | | | | | | | | | **A.T.P.:** Ing. Federico MUIÑO  Mg. Ing. Javier CHINCUINI | | | |
|  | | | |  | | | | | |  | | | |
| LABORATORIO N° 5  “TRANSFORMADA DE FOURIER” | | | | | | | | | | | | | |
| INTEGRANTES GRUPO N° 3 | | | | | | | | FECHA LÍMITE: 18/11/2019 | | | | | |
| LEGAJO | | APELLIDO Y NOMBRES | | | | | | | E – MAIL | | | | RESP. |
| 163.437-9 | | ARGAÑARAZ, Mauro Jesús | | | | | | | mauroarg60@gmail.com | | | |  |
| 163.872-5 | | BALDIVIEZO, Marco Gabriel | | | | | | | baldiviezomg@gmail.com | | | | ● |
| 163.825-7 | | RODRIGUEZ, Matías Ricardo | | | | | | | matias.rodriguez.97.13@gmail.com | | | |  |
| 156.409-2 | | SANTILLAN, José Emanuel | | | | | | | emanuel\_2TM\_1\_7@outlook.com.ar | | | |  |
| PRES. | FECHA ENTREGA | | FECHA DEVOLUCION | | | | OBSERVACIONES | | | | | | |
| 1RA | 18/11/19 | |  | | | |  | | | | | | |
| 2DA |  | |  | | | |  | | | | | | |
| 3RA |  | |  | | | |  | | | | | | |
| FECHA APROBACIÓN | | | | | CALIFICACIÓN | | | | | | FIRMA | | |
|  | | | | |  | | | | | |  | | |

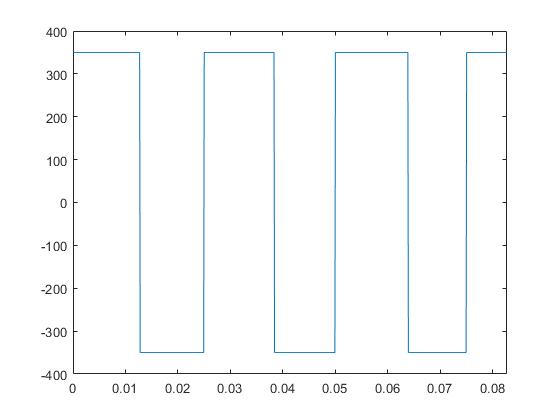
**OBJETIVO:**

Realizar el análisis de una serie de datos discretos por medio del análisis de su espectro de frecuencias, utilizando la Transformada de Fourier, y aplicar el concepto de filtro.

**INTRODUCCION:**

En el archivo “PWM\_MOD\_MOD\_3\_1MHz.txt” encontramos un vector con 333333 datos discretos que describen una señal cuadrada con frecuencia f = 1MHz y amplitud ±350

Gráfico:



Ya sabemos que la Transformada de Fouriery su correspondiente transformada inversa, se utilizan para el análisis para funciones continuas. Esto resulta muy útil para definir conceptos y evaluar analíticamente las mismas, pero para su aplicación en ingeniería no resulta de mucha utilidad ya que las señales a analizar son señales muestreadas y, por lo tanto, son señales discretas o también pueden considerarse como vectores (como es nuestro caso ahora)

Para esto, Matlab tiene una herramienta llamada Fast Fourier Transform (FFT), que trabaja con este tipo de datos.

**DESARROLLO:**

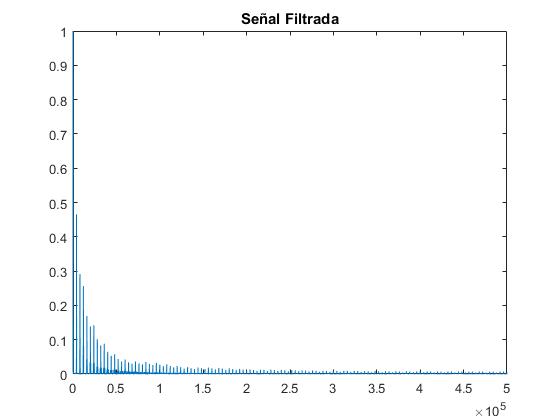
1) Cargamos los datos y los trasponemos así quedan como vector columna. Definimos la longitud de la señal N y en función de este, se crea un vector que representa el tiempo. Ploteamos la señal que es la que se presentó en la introducción.

|  |
| --- |
| senal\_fila = load('PWM\_MOD\_MOD\_3\_1MHz.txt');  senal = transpose(senal\_fila);    N = 333333;  dt = 0.000001;  t = 0:dt:(N-1)\*dt;    % subplot(221)  plot(t,senal)  hold on  title('Señal Original')  xlim([0 0.1]) |

2) Utilizamos el comando FFT para transformar los datos a Fourier. También definimos un vector de frecuencia, para poder graficar el resultado y ver el contenido de armónicos que tienen los datos. También debemos definir el módulo del vector que representa la FFT de la señal ya que la misma está compuesta por números complejos los cuales no podríamos representar en una gráfica de dos dimensiones.

|  |
| --- |
| senal\_tf\_filtrada = senal\_tf.\*H;  % subplot(224)  % plot(f,abs(senal\_tf\_filtrada)/max(abs(senal\_tf)),'r')  % xlim([0 (1e6)/2])  title('Espectro de Señal Filtrada') |

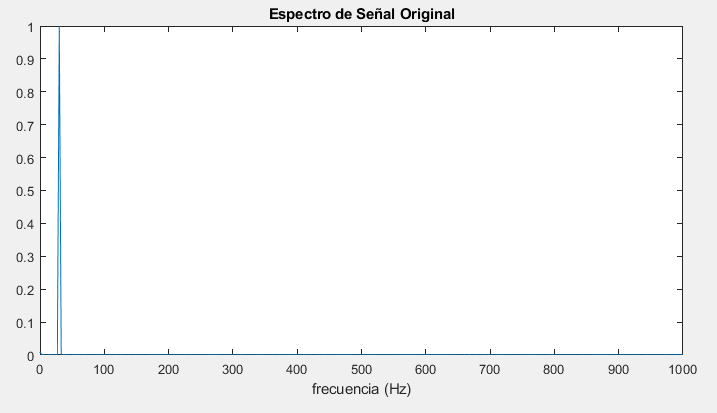
Gráfico:



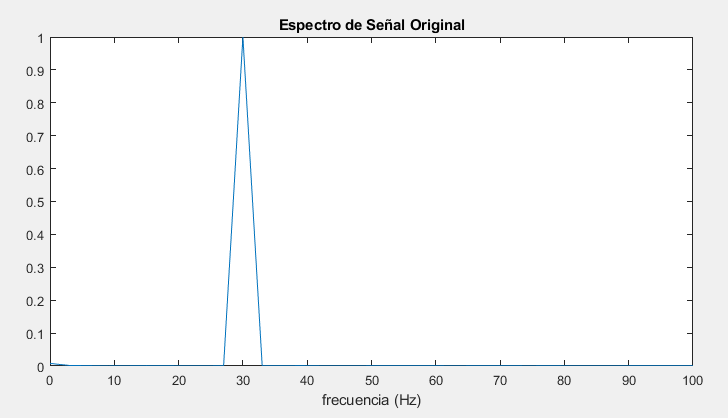
La mayoría de ruido se concentra en las frecuencias bajas.

Analizamos la señal con un rango de muestreo más chico para conocer la frecuencia principal

Elegimos 1000hz.



Procedemos a realizar un zoom entre el rango 0 y 100

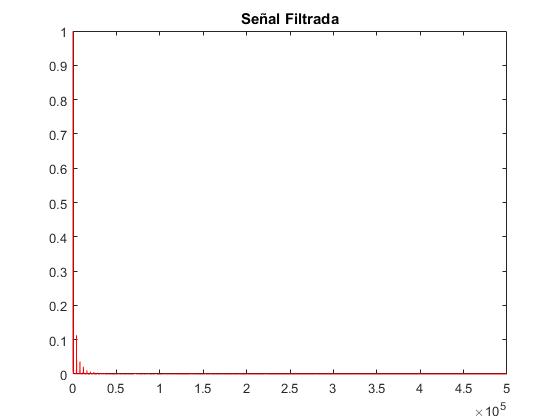


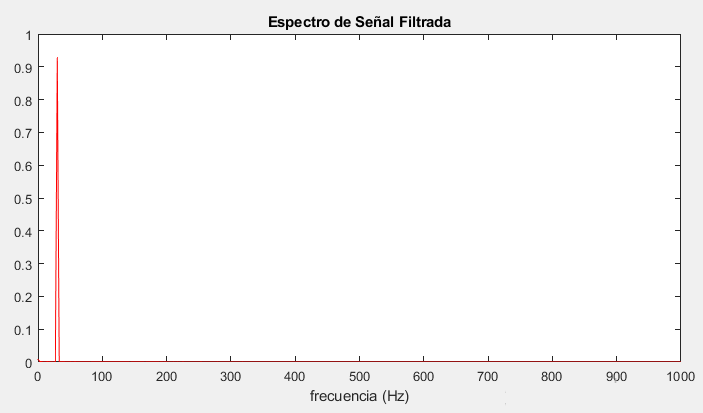
Observamos que el valor de la frecuencia principal es aproximadamente 30hz.

3) Vamos a aplicar un filtro pasa bajos a la señal, de frecuencia de corte 75hz

|  |
| --- |
| fc = 75;  wc = 2\*pi\*fc;  w = 2\*pi\*f;    H = 1./(1+1i\*w/wc);  H(N/2+1:end) = conj(fliplr(H(2:N/2+1)));    senal\_tf\_filtrada = senal\_tf.\*H;  subplot(224)  plot(f,abs(senal\_tf\_filtrada)/max(abs(senal\_tf)),'r')  xlim([0 500])  title('Espectro de Señal Filtrada') |

Gráfico:



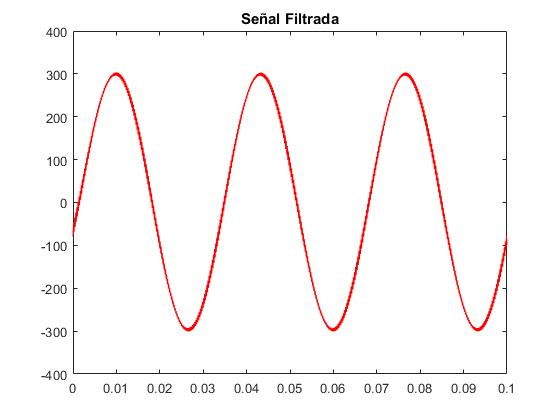


Vemos como esta cumpliendo su función el filtro. Se aplico una frecuencia de corte de 75hz. Las armónicas de frecuencias altas no pasan por el filtro y disminuyen su intensidad. Entonces el contenido de potencia se concentra en las armónicas de baja frecuencia.

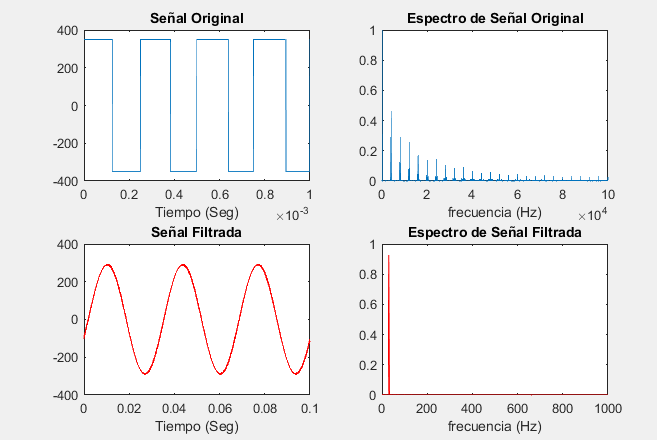
4) Rearmamos la función anti transformando con los datos filtrados.

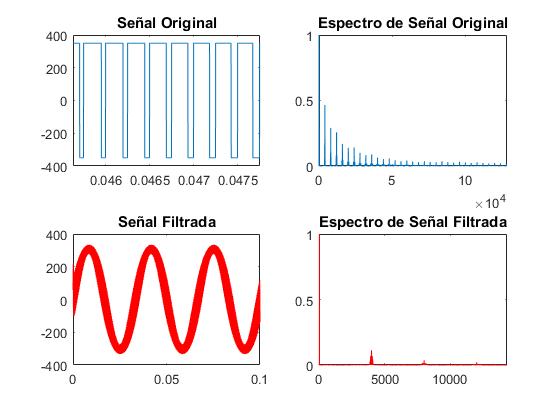
|  |
| --- |
| senal\_filtrada = ifft(senal\_tf\_filtrada);  subplot(223)  plot(t,real(senal\_filtrada),'r')  xlim([0 0.1])  title('Señal Filtrada') |

Gráfico:



Para ver un poco más la mejora, graficamos las señales juntas:

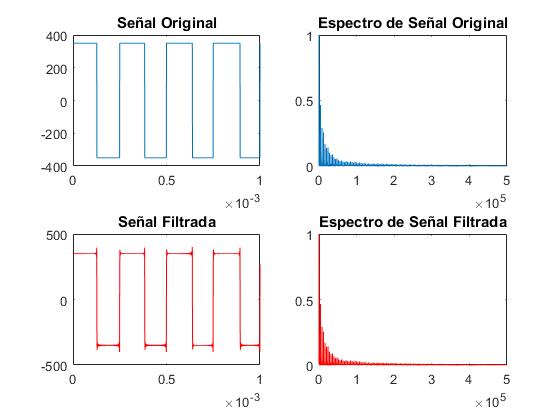




**Conclusiones:**

El filtro cumple su función recortando los armónicos de altas frecuencias.

Para verificar que el filtro este bien aplicado, filtramos a partir de 1e6 que es la frecuencia de la función. De esta forma no debería filtrar nada y dar la misma señal de vuelta.



Al rearmar la función, vemos la aparición del fenómeno de Gibbs. Esto es debido a que cuando definimos los vectores con los que calculamos, algunos arrancan del índice 1 y otros cuentan a partir de 0, entonces tenemos una muy pequeña aproximación.