



Instituto Politécnico Nacional
Escuela Superior De Cómputo



Entregable 18

Limpiar señal con FFT

Nombre de los integrantes:

Hernandez Rodriguez Juan Uriel

Vergara Martinez Brenda

García Quiroz Gustavo Ivan

Gutiérrez Jiménez Cinthia Nayelli

Ramírez Carrillo José Emilio

Iturbide Serrano Uriel

Grupo: 5CV1

Procesamiento Digital de Señales

Nombre del profesor: José Antonio Flores Escobar

Índice de Contenidos

Marco teórico	1
Transformada rápida de Fourier (FFT)	1
Limpiar señal (<i>denoising</i>).....	1
Desarrollo.....	2
Simulación	3
Conclusión	4
Referencias.....	4
Anexo.....	5

Marco teórico

Transformada rápida de Fourier (FFT)

Como se vio en las prácticas anteriores, la FFT es un algoritmo eficiente utilizado para calcular la Transformada Discreta de Fourier (DFT) y su inversa. La DFT convierte una señal en el dominio del tiempo en una señal en el dominio de la frecuencia, lo cual es fundamental para el análisis de señales en campos como procesamiento de audio, imagen y comunicaciones. La FFT también reduce drásticamente el número de cálculos necesarios para realizar esta transformación, pasando de un orden cuadrático a uno logarítmico, lo que la hace extremadamente útil para el procesamiento en tiempo real y aplicaciones que requieren alta velocidad y eficiencia computacional.

Limpiar señal (*denoising*)

Es el proceso mediante el cual se eliminan o reducen los ruidos indeseados presentes en una señal, preservando la información útil y relevante. Este proceso es crucial en diversas aplicaciones, como la mejora de la calidad del audio y la imagen, la extracción de características en señales biomédicas, y la transmisión de datos en comunicaciones. Se utilizan diversas técnicas para lograr el *denoising*, como filtros digitales (pasabajas, pasaaltas, etc.), transformadas de dominio (como la Transformada de Fourier y la Transformada Wavelet), y métodos estadísticos. El objetivo principal es mejorar la claridad y la precisión de la señal original, facilitando su análisis y uso posterior. En este caso se usó la FFT para limpiar la señal, cuyo algoritmo es directo y fácil de aplicar en Matlab.

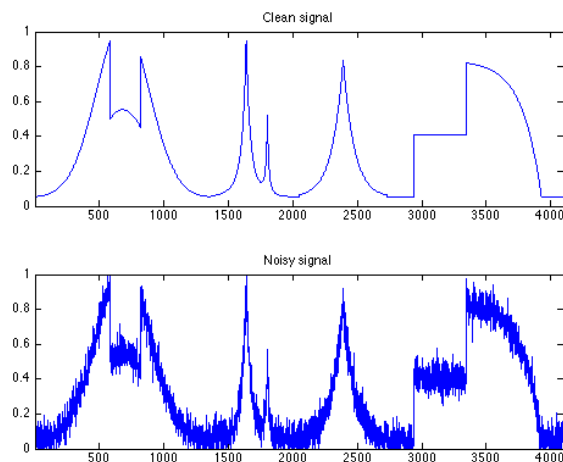


Ilustración 1 - Diferencia entre una señal ruidosa y una señal sin ruido

Desarrollo

Para el desarrollo de esta práctica se tuvo que aplicar la FFT a una señal, obtener su espectro, determinar qué frecuencias se desean eliminar, eliminarlas y aplicar la FFT inversa para recuperar la señal.

Para éste método primero se debe de crear una señal con dos frecuencias, sumar dos señales en diferente frecuencia y aplicar ruido a ambas señales.

Posteriormente se debe de obtener la longitud de T, que es imperativo para calcular su FFT. Teniendo eso, se calcula la densidad espectral de potencia (PSD) de

ambas señales, lo cual permite analizar la distribución de energía en el dominio de la frecuencia.

También se define el vector de frecuencias *freq* y se toma la mitad de las frecuencias para graficar,

```
% Obtener la longitud de T
n = length(t);
% Calcular la FFT
fhat = fft(f, n);
fhatclean = fft(fclean, n);
% Obtener el espectro de nuestra señal
% Obtener la densidad de energia de nuestra señal
PSD = fhat.*conj(fhat)/n;
PSDclean = fhatclean.*conj(fhatclean)/n;
% Obtener la frecuencia de la señal en Hz para eje de las x
freq = 1/(dt*n)*(0:n);
% Obtener la primera parte de la señal
L = 1:floor(n/2);
```

Ilustración 3 - Cálculo de la FFT y PSD

Finalmente, se debe de reconstruir la señal filtrada por lo que se calcula la IFFT de la señal filtrada para obtener la señal en el dominio del tiempo.

```
% Para la practica se crea una señal con dos frecuencias
dt = .001;
t = 0:dt:1;
% Sumar dos señales en diferente frecuencia
fclean = sin(2*pi*t*50) + sin(2*pi*t*120);
% A continuacion se agrega ruido a la señal
f = fclean + 2.5*randn(size(t));
% Graficar ambas señales
figure(1);
subplot(2,1,1);
plot(fclean); title("Señal Original Limpia");
subplot(2, 1, 2);
plot(f); title("Señal Original Limpia + Ruido");
```

Ilustración 2 - Creación de señales ruidosas

ya que el espectro es simétrico. Después se identifican las frecuencias con una densidad de potencia superior a un umbral (50 en este caso) y se eliminan las componentes de frecuencia con una densidad de potencia inferior a este umbral.

```
%% Filtrar y calcular la IFFT
% Encontrar las frecuencias que deseamos eliminar
indices = PSD > 50;
% Hacer ceros las frecuencias menores a 50
PSD2 = PSD.*indices;
% Hacer ceros los coeficientes de la transformada
fhat2 = fhat.*indices;
% Graficar el espectro "limpio"
figure(3);
plot(freq(L), PSD2(L));
title("Se eliminan frecuencias menores a 50");
% Obtener la fft inversa de fhat
ifftfhat2 = ifft(fhat2);
% Obtener la FFT inversa de PSD2
ifftPSD2 = ifft(PSD2);
```

Ilustración 4 - Filtrado y reconstrucción de las señales

Simulación

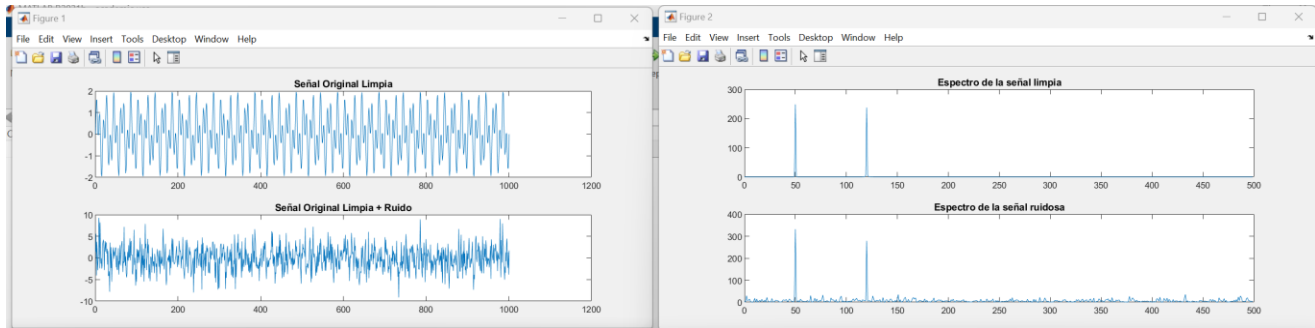


Ilustración 5 - Gráfica y espectro de ambas señales

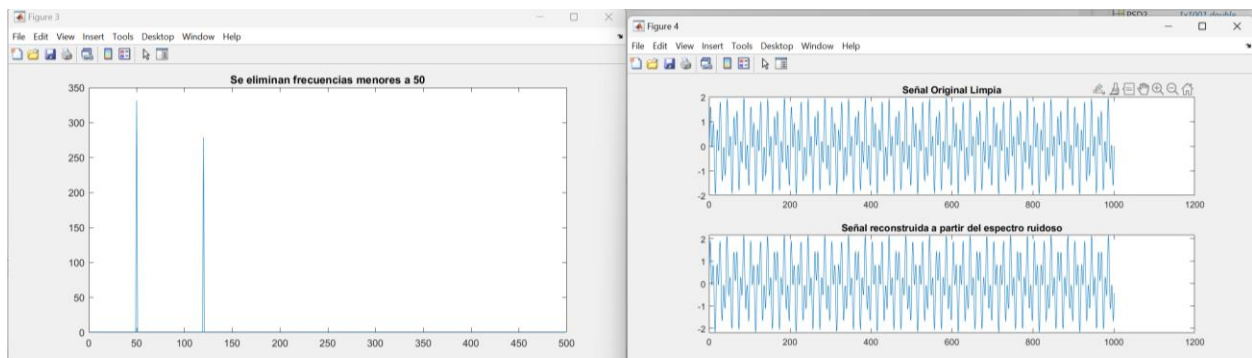


Ilustración 4 - Eliminación de frecuencias y reconstrucción de señal y espectro

Conclusión

Tras realizar esta práctica pudimos observar cómo funciona en práctica la teoría de limpiar una señal utilizando la FFT. A grandes rasgos lo único que se debe hacer es determinar qué frecuencias consideramos como ruido en nuestra señal y, tras haber obtenido su espectro, aplicar la FFT. Una vez obtenida se debe de reconstruir la señal con el umbral de frecuencias ya removida.

A pesar de que se puede hacer manualmente, Matlab cuenta con la función *ddencomp*, que simplifica el proceso y limpia el ruido de una señal de manera automática. Tiene varios parámetros por lo que es importante leer su documentación antes de emplearla. Si bien es muy útil para futuros usos, es imperativo saber cómo funciona de manera fundamental la limpieza de una señal ya que teniendo clara la teoría se puede implementar en otros programas y no sólo en Matlab.

Referencias

- [1] Jirafe, N. (2022, 30 marzo). Using FFT to analyse and cleanse time series data - Neha Jirafe - Medium. Medium. <https://nehajirafe.medium.com/using-fft-to-analyse-and-cleanse-time-series-data-d0c793bb82e3>
- [2] Remove noise signal from signal using fft. (s. f.). Signal Processing Stack Exchange. <https://dsp.stackexchange.com/questions/86585/remove-noise-signal-from-signal-using-fft>
- [3] Islam, S., Elmekki, H., Elsebai, A., Bentahar, J., Drawel, N., Rjoub, G., & Pedrycz, W. (2024). A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems With Applications*, 241, 122666. <https://doi.org/10.1016/j.eswa.2023.122666>

Anexo

Código.

```
% Entregable    18 - Limpiar señal con FFT
% Grupo        5CV1
% Equipo        7
% Profesor: José Antonio Flores Escobar
% Integrantes:
%              Hernandez Rodriguez Juan Uriel
%              Vergara Martinez Brenda
%              García Quiroz Gustavo Ivan
%              Gutiérrez Jiménez Cinthia Nayelli
%              Ramírez Carrillo José Emilio
%              Iturbide Serrano Uriel
% Limpiar una señal ruidosa
clear all;
close all;
clc;

% Para la practica se crea una señal con dos frecuencias
dt = .001;
t = 0:dt:1;
% Sumar dos señales en diferente frecuencia
fclean = sin(2*pi*t*50) + sin(2*pi*t*120);
% A continuacion se agrega ruido a la señal
f = fclean + 2.5*randn(size(t));
% Graficar ambas señales
figure(1);
subplot(2,1,1);
plot(fclean); title("Señal Original Limpia");
subplot(2, 1, 2);
plot(f); title("Señal Original Limpia + Ruido");

% Obtener la longitud de T
n = length(t);
% Calcular la FFT
fhat = fft(f, n);
fhatclean = fft(fclean, n);
% Obtener el espectro de nuestra señal
% Obtener la densidad de energia de nuestra señal
PSD = fhat.*conj(fhat)/n;
PSDclean = fhatclean.*conj(fhatclean)/n;
% Obtener la frecuencia de la señal en Hz para eje de las x
freq = 1/(dt*n)*(0:n);
% Obtener la primera parte de la señal
L = 1:floor(n/2);

% Graficar PSD para señal limpia y con ruido
figure(2);
subplot(2,1,1)
plot(freq(L), PSDclean(L));
title ("Espectro de la señal limpia");
```

```

subplot(2,1,2);
plot(freq(L), PSD(L));
title ("Espectro de la señal ruidosa");

%%% Filtrar y calcular la IFFT
% Encontrar las frecuencias que deseamos eliminar
indices = PSD > 50;
% Hacer ceros las frecuencias menores a 50
PSD2 = PSD.*indices;
% Hacer ceros los coeficientes de la transformada
fhat2 = fhat.*indices;
% Graficar el espectro "limpio"
figure(3);
plot(freq(L), PSD2(L));
title("Se eliminan frecuencias menores a 50");
% Obtener la fft inversa de fhat
ifftfhat2 = ifft(fhat2);
% Obtener la FFT inversa de PSD2
ifftPSD2 = ifft(PSD2);
% Graficar la señal con espectro limpio
figure(4);
subplot(2,1,1);
plot(fclean); title("Señal Original Limpia");
subplot(2,1,2);
plot(ifftfhat2); title("Señal reconstruida a partir del espectro ruidoso");

```