



## Centro Universitario de los Valles

**Vladimir Ortiz Jiménez** 221380156

**Ignacio Andrade Salazar** 221380334

**Edgar Geovany** 221380237

### **Análisis de Señales y Sistemas**

Profesora: Dra. Teresa Efigenia Alarcón Martínez

12 de mayo del 2024

## Contenido

1. Introducción .....	3
2. Objetivos.....	4
3. Fundamento teórico.....	4
4. Descripción de los experimentos realizados. Interpretación ...	7
5. Conclusiones .....	16
Bibliografía .....	17

# 1. Introducción

En la sociedad moderna se ha extendido el uso de la tecnología para detectar automáticamente actividades o individuos de interés, con el objetivo de fortalecer la seguridad e incrementar la calidad de vida de los ciudadanos. De esta manera vemos que se diseñan sistemas automatizados que utilizan las cámaras de vigilancia instaladas para detectar vehículos robados, personas perdidas o inclusive individuos buscados por el departamento de policía. Por otra parte, las llamadas telefónicas y conversaciones grabadas en ambientes de trabajo pueden ser monitoreadas para encontrar indicios de una potencial amenaza a la seguridad pública o privada. Otro ejemplo es el área de la educación a distancia, donde es común encontrar retos como la verificación de identidad para reducir la posibilidad de que un estudiante haga trampa al contestar un examen desde su hogar. Además de estos ejemplos, la aplicación de los sistemas basados en técnicas de procesamiento de señales puede extenderse tanto como la imaginación lo permita. Estos sistemas pertenecen a un área conocida como biometría.

En este proyecto se da un primer acercamiento al análisis de la señal de voz. Se consideran audios resultados de decir los números del 0 al 9 y se necesita caracterizar estas señales en el dominio del tiempo y en el dominio de la frecuencia.

En el presente proyecto, se lleva a cabo un acercamiento primordial al análisis de señales de voz, donde se explora la caracterización de estas señales tanto en el dominio del tiempo como en el de la frecuencia. A través de una serie de experimentos prácticos, se busca comprender y explicar el comportamiento de las señales de voz, evaluando la eficacia de técnicas de filtrado, en particular el filtrado mediante la transformada de Fourier, para la eliminación del ruido y la recuperación de la señal original. Este enfoque no solo permite comprender los aspectos teóricos fundamentales del análisis de señales de voz, sino también su aplicación práctica en la mejora de la calidad y la precisión en diversos contextos, desde la seguridad hasta la comunicación y la educación.

## 2. Objetivos.

1. Que el estudiante comprenda y explique el análisis de la señal de voz en el tiempo.
2. Que el estudiante comprenda y explique el análisis de la señal voz en el dominio de la frecuencia (Fourier).

## 3. Fundamento teórico.

### Señal

Una señal es una representación matemática de un fenómeno físico o abstracto que lleva información. Una señal se puede representar como una función matemática que varía en el tiempo, como una onda eléctrica, una vibración mecánica, una señal de audio o una señal digital. Las señales pueden ser continuas en el tiempo, como las señales analógicas, o discretas en el tiempo, como las señales digitales. Las señales pueden describirse en términos de su amplitud, frecuencia, fase y otras características.

### Señal de audio

Una señal de audio es una representación eléctrica de las variaciones en la presión del aire a lo largo del tiempo. Puede ser analógica o digital y captura sonidos como música, voz o ruidos ambientales. Se utiliza en aplicaciones como reproducción de sonido, comunicaciones y procesamiento digital de audio.

### Máximo y Mínimo

**Máximo:** Es el valor más alto que alcanza una señal en un intervalo de tiempo específico o en toda su duración. Representa el punto más alto de una onda en su ciclo.

**Mínimo:** Es el valor más bajo que alcanza una señal en un intervalo de tiempo específico o en toda su duración. Representa el punto más bajo de una onda en su ciclo.

Son importantes en el análisis de señales para determinar características como la amplitud máxima y mínima de una señal, la cantidad de variación de la señal en un intervalo de tiempo y para identificar patrones de comportamiento de la señal. Estos valores pueden ser útiles en una variedad de aplicaciones, como el procesamiento de señales, la detección de eventos y la monitorización de sistemas.

## **MATLAB**

Es una herramienta versátil y poderosa que puede ser utilizada en todas las etapas del análisis de señales, desde la adquisición y visualización de datos hasta el procesamiento, diseño de sistemas y simulación. Su amplia gama de funciones y su facilidad de uso lo convierten en una opción popular entre los ingenieros y científicos que trabajan en el campo del análisis de señales.

## **La contaminación de ruido en una señal**

Es interferencia no deseada que se suma a la señal original. Esta interferencia puede distorsionar la señal, dificultando la extracción precisa de la información relevante. Es esencial identificar y mitigar el ruido para garantizar un análisis preciso de la señal y la extracción efectiva de la información deseada. Esto puede lograrse mediante técnicas de filtrado, procesamiento de señales o modelado para contrarrestar los efectos del ruido.

## **Ruido gaussiano**

Tipo de ruido aleatorio que sigue una distribución normal (o gaussiana). Se caracteriza por tener propiedades estadísticas específicas.

### **Distribución Normal:**

Los valores del ruido están distribuidos según una distribución normal, también conocida como distribución de campana de Gauss.

**Media y Desviación Estándar:**

La distribución normal está definida por dos parámetros principales: la media ( $\mu$ ) y la desviación estándar ( $\sigma$ ). La media representa el valor esperado o promedio de la distribución, mientras que la desviación estándar indica cuánto varían los valores alrededor de la media.

**Independencia y Linealidad:**

El ruido gaussiano es independiente y lineal. Esto significa que los valores de ruido en diferentes momentos son estadísticamente independientes entre sí

**Transformada de Fourier**

Es una herramienta matemática fundamental en el análisis de señales que descompone una señal en sus componentes de frecuencia. Esencialmente, convierte una señal del dominio del tiempo en su representación en el dominio de la frecuencia

La Transformada de Fourier puede ser continua o discreta, dependiendo de si se aplica a señales continuas en el tiempo (como señales analógicas) o señales discretas en el tiempo (como señales digitales).

**Filtrado**

En el dominio de la frecuencia: Una vez que una señal se ha transformado al dominio de la frecuencia mediante la Transformada de Fourier, es posible aplicar técnicas de filtrado directamente en este dominio. Esto permite filtrar selectivamente ciertas frecuencias de la señal, eliminando el ruido no deseado o resaltando las frecuencias de interés.

## 4. Descripción de los experimentos realizados.

### Interpretación

#### 1: Seleccionar un archivo de audio (señal).

Se ha seleccionado el archivo de audio con el cual se llevará a cabo el proceso de filtración mediante la transformada de Fourier. Este archivo lleva por nombre "5\_lucas\_3" y cuenta con la extensión ".wav".

```
%%  
[y,Fs] = audioread('5_lucas_3.wav');
```

Figura 1

En la figura 1 se observa el archivo de audio que se almacena en dos variables, la variable "y" almacena la señal mientras que "Fs" almacena la frecuencia de muestreo, es decir, el número de muestras por segundo que se tomó para grabar el audio.

```
%% calcula tiempo de la señal  
muestras = numel(y);  
tiempo = muestras/Fs;  
ejex = linspace(0,tiempo,muestras);  
  
plot(ejex, y)  
title("Señal de audio")  
xlabel('t')  
ylabel('Amplitud')  
sound(y,Fs);  
legend('5 lucas 3');  
grid on;
```

Figura 1.2

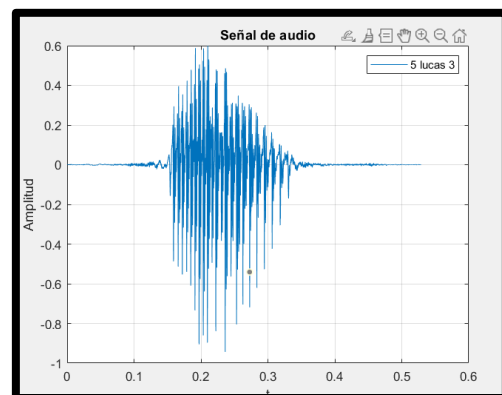


Figura 1.3

En las Figuras 1.2 se realizó el cálculo del tiempo para representar en el gráfico la duración total de la grabación. En la Figura 1.3, se puede apreciar dicha representación gráfica junto con la señal.

## 2: Calcular la amplitud máxima y mínima de la señal seleccionada.

```
% calcula máxima y mínima amplitud
[maximo,Indmax] = max(y);
[minimo,Indmin] = min(y);
% busca correspondencia entre los índices y el tiempo
t_for_max = Indmax * tiempo/muestras;
t_for_min = Indmin * tiempo/muestras;
```

Figura 2

Indmax	1682
Indmin	1890

Figura 2.1

Se calcularon tanto el punto máximo como el punto mínimo de la señal utilizando las funciones `max()` y `min()` proporcionadas por MATLAB. En el caso de la función `max()`, se obtendrá el valor máximo de la señal junto con el índice correspondiente. Para la función `min()`, se registró tanto el valor mínimo como el índice donde se localiza dicho valor.

## 3. Visualizar la señal obtenida y los valores de mínima y máxima amplitudes.

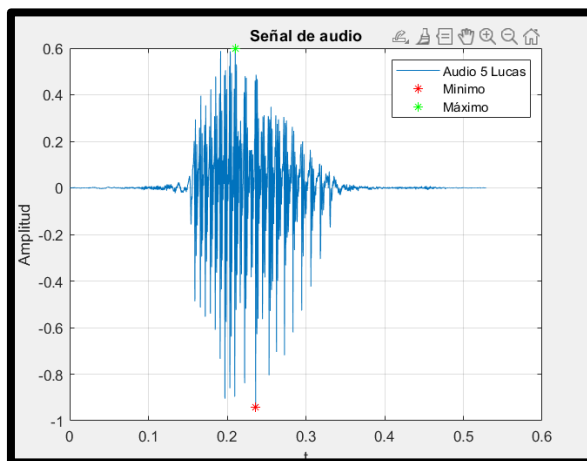


Figura 3

ejex	1x4229 double
Fs	8000
Indmax	1682
Indmin	1890
maximo	0.5995
minimo	-0.9435
muestras	4229
t_for_max	0.2103
t_for_min	0.2363
tiempo	0.5286
y	4229x1 double

Figura 3.1

En la figura 3 se ha destacado el punto máximo con un asterisco de color verde y el punto mínimo con otro de color rojo. En la Figura 3.1, se proporciona un análisis detallado de las posiciones del valor máximo, ubicadas en 0.5995 en el eje "Y" en  $t:0.2103$ , y el valor mínimo, que se encuentra en -0.9435 en  $t:0.2363$ .



#### 4. Reproducir la señal seleccionada

En MATLAB, existe una función denominada `sound` que permite la reproducción de una señal de audio. Los parámetros necesarios para esta función coinciden con los obtenidos inicialmente, tal como se muestra en la Figura 1.

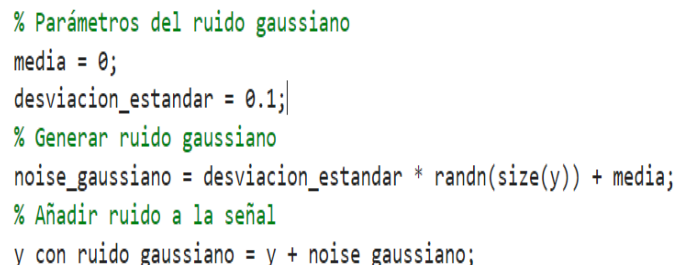
A screenshot of a MATLAB code snippet enclosed in a black rectangular border. The code is `sound(y, Fs);` written in a monospaced font with syntax highlighting: `sound` is blue, `(y, Fs)` is green, and the semicolon is red.

Figura 4

La función `sound` es una herramienta fundamental. Utilizando los parámetros recuperados inicialmente, que incluyen la señal de audio y la frecuencia de muestreo, esta función permite reproducir la señal

#### 5. Contaminar la señal con ruido uniforme. Visualice la señal contaminada.

Se optó por introducir un ruido gaussiano, el cual implica la adición de ruido aleatorio a una señal que sigue una distribución normal. Para implementar esto, se generó una señal independiente que contiene este tipo de ruido, y luego se sumó a la señal seleccionada para la adición de ruido. El código correspondiente a esta operación puede visualizarse en la Figura 5.

A screenshot of MATLAB code for adding Gaussian noise, enclosed in a black rectangular border. The code is as follows:

```
% Parámetros del ruido gaussiano
media = 0;
desviacion_estandar = 0.1;
% Generar ruido gaussiano
noise_gaussiano = desviacion_estandar * randn(size(y)) + media;
% Añadir ruido a la señal
y_con_ruido_gaussiano = y + noise_gaussiano;
```

The code uses syntax highlighting: comments are green, variable names and function names are blue, and operators and parentheses are black.

Figura 5

En la Figura 5.1, se aprecia de manera gráfica la representación de la señal original tras la adición de la señal de ruido gaussiano. Esta representación gráfica permite observar cómo se modifica la señal original con la inclusión del ruido gaussiano, manteniendo la estructura temporal de la señal inicial.

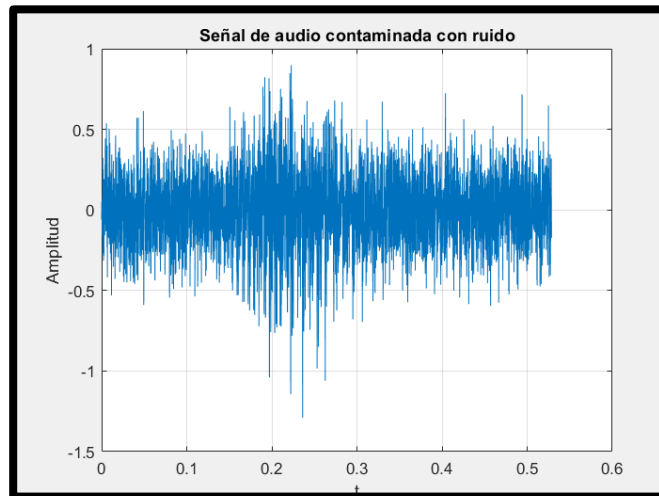


Figura 5.1

## 6. Filtrado de la señal ruidosa, en el dominio del espacio.

En la Figura 6, se exhibe el código utilizado para filtrar la señal en cuatro ocasiones, con el objetivo de lograr una mayor atenuación del ruido y así obtener una señal que se asemeje lo más posible a la original. Asimismo, se incluye la representación gráfica de la señal filtrada en la Figura 6.1. Esta visualización permite evaluar el efecto del filtrado en la señal, destacando los cambios realizados para mitigar el ruido y recuperar la forma original de la señal.

```
%% Se filtra la señal en el tiempo y se visualiza.
windowSize = 5;
b = (1/windowSize)*ones(1,windowSize);
a = 1;
y_filtered = filter(b,a,y_noise);
y_filtered = filter(b,a,y_filtered);
y_filtered = filter(b,a,y_filtered);
y_filtered = filter(b,a,y_filtered);

% visualización de la señal filtrada
figure
plot(ejex, y_filtered)
title("Señal de audio filtrada")
xlabel('t')
ylabel('Amplitud')
grid on;
```

Figura 6

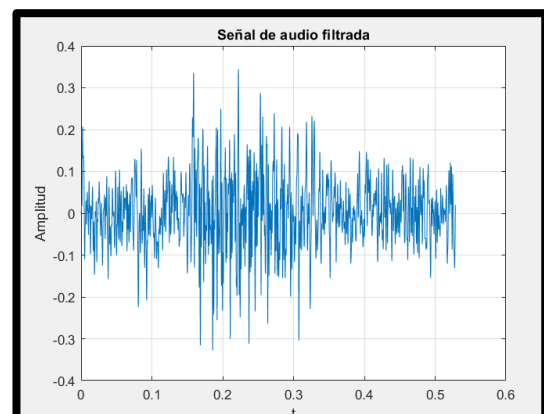


Figura 6.1

## 7.Llevar la señal original y la señal contaminada con ruido al dominio de las frecuencias.

En las siguientes dos líneas de Código se calcula la transformada rápida de Fourier (FFT) de las señales `y` (original) y `y_noise` (contaminada con ruido), respectivamente. La FFT se utiliza para transformar las señales del dominio del tiempo al dominio de la frecuencia, lo que nos permite analizar las componentes de frecuencia de las señales (Figura 7).

```
%% Procesamiento en frecuencia  
  
Y = fft(y);  
Y_noise = fft(y_noise);
```

Figura 7

Se observa en la figura que se está creando un eje de frecuencia que permite analizar las características de la señal de audio en el dominio de la frecuencia. Esto implica determinar qué rangos de frecuencia están presentes en la señal y con qué intensidad. Esto es crucial para entender cómo se distribuyen las diferentes frecuencias en la señal de audio, lo que puede ser útil para identificar patrones específicos, en este caso la presencia de ruido o la modulación de la señal.

```
%% Construyendo eje de frecuencias  
f_max = Fs/2;  
eje_f = (linspace (0,f_max*2,numel(y)))';
```

Figura 7.1

El siguiente bloque de código (Figura 7.2) crea una figura con dos gráficos. En el primer gráfico, muestra cómo están distribuidas las diferentes frecuencias en la señal de audio original. En el segundo gráfico, hace lo mismo, pero para la señal de audio con ruido añadido. Esto ayuda a comparar cómo el ruido afecta la distribución de frecuencias en la señal.

```
figure
subplot(1,2,1)
plot(eje_f,abs(Y))
xlabel("Hertz");
ylabel("|F|")
title("Y")
subplot(1,2,2)
plot(eje_f,abs(Y_noise))
title("Y\_noise")
xlabel("Hertz");
ylabel("|F|")
```

Figura 7.2

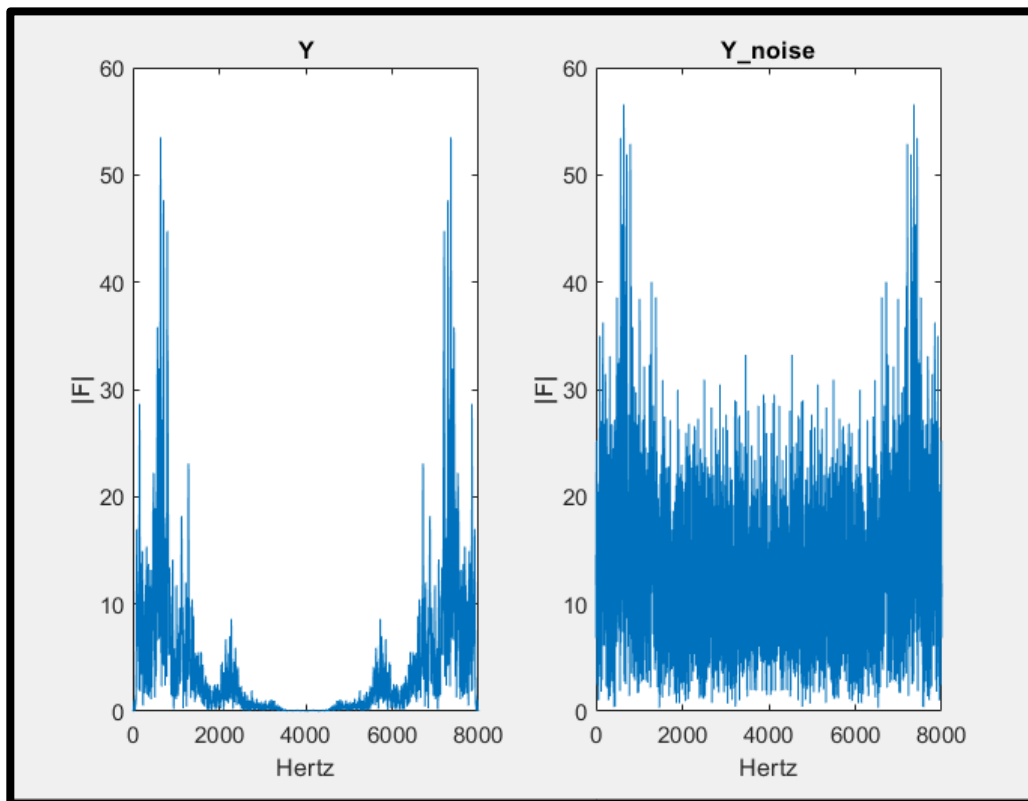


Figura 7.3. Gráfica de las frecuencias.

En la siguiente imagen podemos observar como en el primer subgráfico, se traza la señal de audio original contaminada con ruido ( $y_{\text{noise}}$ ) en función del tiempo (ejex).

En el segundo subgráfico, se traza la señal obtenida al realizar la transformada inversa de Fourier de la señal contaminada con ruido ( $y_{\text{IFFT\_noise}}$ ) en función del tiempo (ejex).

Esto permite comparar visualmente la señal original con la señal obtenida (Figura 7.5) mediante la transformada inversa de Fourier, lo que ayuda a comprender cómo se restaura la señal después de filtrarla en el dominio de la frecuencia.

```
%% Revisa que se logra con la transformada inversa
figure
% Regresamos al tiempo. Comparemos y_noise y y_IFFT_noise
y_IFFT_noise = ifft(Y_noise);
subplot(1,2,1)
plot(ejex, y_noise)
xlabel('t')
ylabel('Amplitud')
subplot(1,2,2)
plot(ejex, y_IFFT_noise)
xlabel('t')
ylabel('Amplitud')
```

Figura 7.4

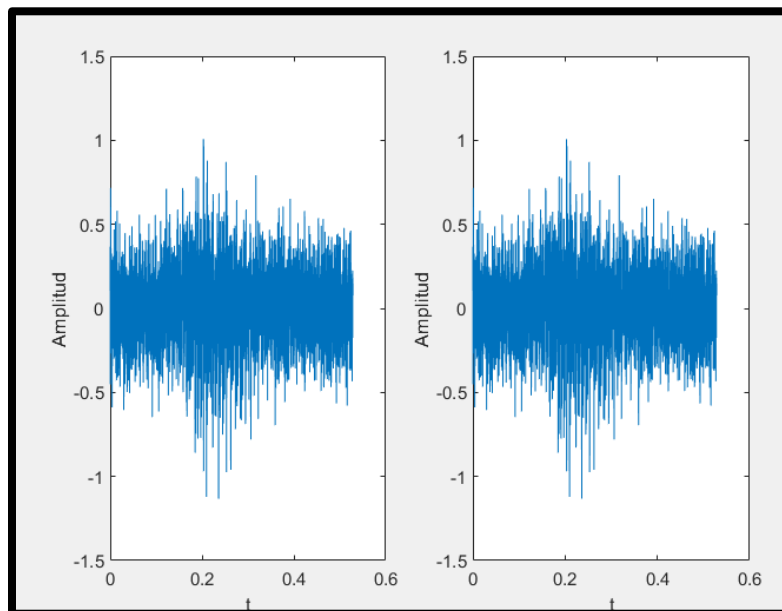


Figura 7.5

## **8. Eliminación de las frecuencias asociadas al ruido y regreso de la señal al dominio del tiempo. ¿Qué sucedió?**

En la figura 8 se muestra el código, filtra una señal de audio para eliminar parte del ruido. Luego se compara la señal filtrada con la original para ver cómo cambia después del proceso de filtrado.

### **Filtra la señal de audio en el dominio de la frecuencia:**

- Se crea un filtro en el dominio de Fourier que elimina ciertas frecuencias no deseadas.
- Se aplica este filtro a la transformada de Fourier de la señal de audio contaminada con ruido.
- Luego, se realiza una transformada inversa de Fourier para recuperar la señal filtrada en el dominio del tiempo.

### **Compara la señal filtrada con la original:**

- Se visualiza la señal filtrada en el dominio del tiempo ( $y_{\text{IFFT}}$ ) junto con la señal filtrada previamente en el dominio del tiempo ( $y_{\text{filtered}}$ ).
- Esto permite comparar cómo cambia la señal después del proceso de filtrado en los dominios de la frecuencia y del tiempo.

En la Figura 8.1, se puede apreciar visualmente el resultado de la eliminación del ruido de manera gráfica. Este análisis gráfico ofrece una representación clara de cómo el proceso de eliminación de ruido ha afectado la señal original, resaltando la mejora en la calidad y la claridad de la señal tras la eliminación del ruido.

```

%% Filtrar la señal en el espacio de Fourier.
filter_F = ( [ ones(1,100) , zeros(1,numel(y)-100)] )';
filtered_signal_F = Y_noise .* filter_F;
y_IFFT = ifft(filtered_signal_F);
% Visualiza la señal filtradas en Fourier y compara con la
% señal filtrada en el tiempo y el proceso que conllevó.
figure
subplot(1,2,1)
plot(ejex, y_IFFT)
xlabel('t')
ylabel('Amplitud')
title('y\_{IFFT}')
subplot(1,2,2)
plot(ejex, y_filtered)
xlabel('t')
ylabel('Amplitud')
title('y\_{filtered}')

```

Figura 8

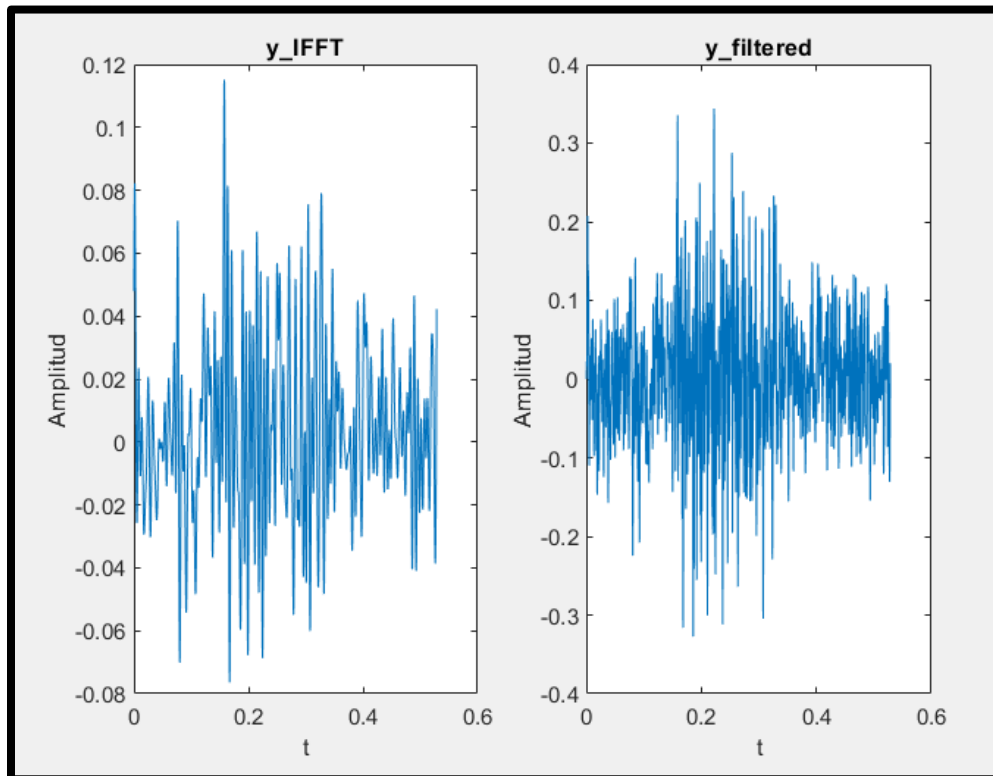


Figura 8.1

## 5. Conclusiones

En conclusión, este experimento demostró la aplicación práctica de técnicas de procesamiento de señales para analizar, manipular y restaurar señales de audio. Inicialmente, se calculó la amplitud máxima y mínima de la señal seleccionada, lo que proporcionó una comprensión básica de sus características. Luego, se visualizó la señal original junto con los valores de amplitud máxima y mínima.

Al reproducir la señal seleccionada, se pudo apreciar su contenido auditivo. Posteriormente, al contaminar la señal con ruido uniforme, se simularon condiciones realistas de transmisión o grabación. La visualización de la señal contaminada reveló cómo el ruido afecta la señal original, introduciendo perturbaciones no deseadas.

El siguiente paso fue filtrar la señal ruidosa en el dominio del espacio, para reducir el ruido y mejorar la calidad de la señal. Luego, se llevó tanto la señal original como la contaminada al dominio de las frecuencias, donde se identificaron y eliminaron las frecuencias asociadas al ruido.

Sin embargo, al regresar la señal filtrada al dominio del tiempo, se observó que la restauración completa de la señal original no siempre fue posible. Esto sugiere que el proceso de filtrado puede introducir artefactos o distorsiones en la señal, especialmente si se eliminan frecuencias importantes junto con el ruido.

En resumen, este experimento resalta la importancia del procesamiento de señales en la manipulación y restauración de señales de audio. Si bien las técnicas de filtrado pueden ayudar a reducir el ruido y mejorar la calidad de la señal, es fundamental seleccionar parámetros y técnicas adecuadas para minimizar las distorsiones y preservar el contenido relevante de la señal original.



## Bibliografía

César, V., & Lorenzana. (n.d.). Transformada de Fourier P R E S E N T A  
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA  
DIVISIÓN DE CIENCIAS BÁSICAS COORDINACIÓN DE CIENCIAS APLICADAS  
Ciudad Universitaria, Cd. Mx., noviembre de 2022

Hasso, A. (2021). Signal System Analysis using MATLAB. UNIVERSITY OF ZAKHO.

Mata González, G., Sánchez Esquivel, V. M., & Gómez González, J. M. (2017). Análisis de sistemas y señales con cómputo avanzado. DGAPA-PAPIME, UNAM. Recuperado de <http://www.librosoa.unam.mx/handle/123456789/293>

Proakis, J. G., & Manolakis, D. G. (2007). Tratamiento digital de señales. Pearson/Prentice Hall.

Chaparro, L. F. (2018). Signals and Systems using MATLAB. Elsevier. Recuperado de <https://infonics.files.wordpress.com/2015/03/signals-systems-using-matlab-luis-f-chaparro.pdf>