

# Gas sensor array under dynamic gas mixtures Dataset analysis

Alexis Horacio López Fragoso

Universidad Autónoma de México,  
Facultad de Estudios Superiores Cuautitlán,  
Cuautitlán, Estado de México, México  
guados123@gmail.com

**Abstract.** El avance en el procesamiento de datos ha impulsado el desarrollo de nuevas herramientas para el análisis de diversos fenómenos físicos, lo que plantea la necesidad de encontrar técnicas más efectivas para la representación de señales continuas en el tiempo y datos estocásticos. Este proyecto tiene como objetivo analizar las señales continuas provenientes de cuatro sensores de gas, que detectarán una mezcla dinámica de etanol y dióxido de carbono. A través de un análisis detallado de estas señales, se busca no solo caracterizar las propiedades físicas y químicas de la mezcla de gases, sino también desarrollar un modelo preciso para su interpretación. Los resultados obtenidos podrían contribuir a optimizar los sistemas de detección de gases en tiempo real y mejorar la precisión de dispositivos en aplicaciones industriales y ambientales.

**Keywords:** procesamiento de datos, análisis de fenómenos físicos, señales continuas, sensores de gas, mezcla dinámica, etanol, dióxido de carbono

## 1 Introducción

En las últimas décadas, la recopilación y el almacenamiento de datos sobre distintos fenómenos físicos han cobrado una importancia crucial en numerosos campos científicos y tecnológicos. La capacidad de registrar y analizar con precisión estos fenómenos no solo permite una mejor comprensión, sino también un control más eficiente de sistemas muy complejos, lo cual resulta particularmente relevante en entornos industriales donde los procesos requieren altos estándares de seguridad y precisión. Uno de los campos que se han visto en un crecimiento significativo es el de la detección de gases, donde se emplean sensores especializados para monitorear la calidad del aire y prevenir riesgos asociados con la presencia de gases nocivos. Estos sensores juegan un papel vital en la identificación de concentraciones peligrosas de sustancias que, de no ser controladas, podrían generar riesgos tanto para la salud humana como para el medio ambiente. Los datos recogidos por estos sensores se almacenan en bases de datos, proporcionando información importante sobre las características específicas de los gases detectados, lo que facilita el análisis posterior para la toma de decisiones.

En este contexto, la capacidad de detectar y analizar en tiempo real la presencia de gases como el etanol y el dióxido de carbono es fundamental para garantizar la seguridad en una amplia variedad de entornos industriales. El monitoreo de estas sustancias es particularmente importante en industrias donde la manipulación de productos químicos es rutinaria, como en los laboratorios farmacéuticos, donde las mezclas de gases pueden afectar tanto la calidad del producto como la seguridad del personal. Además, la detección precisa y rápida de estos gases puede reducir significativamente los riesgos de contaminación ambiental, permitiendo a las empresas mejorar sus protocolos de seguridad y minimizar la exposición a gases peligrosos, que podrían resultar dañinos.

Desde una perspectiva tecnológica, este proyecto tiene como objetivo contribuir al análisis y la detección precisa de los datos provenientes de sensores de gas. Esto permitirá no solo adquirir una comprensión más profunda en el ámbito científico, sino también optimizar los sistemas de control en tiempo real en industrias donde la detección temprana de rarezas en las concentraciones de gases es fundamental. La correcta interpretación de los datos recogidos por los sensores no solo optimiza los procesos industriales, sino que también mejora la capacidad para gestionar situaciones críticas.

A pesar de los avances en inteligencia artificial y las técnicas de análisis de datos, aún hay importantes limitaciones en el análisis de mezclas dinámicas de gases. Estas limitaciones hacen difícil la interpretación correcta de los datos, lo que puede deberse a factores como la adquisición no supervisada de los datos de los sensores, la falta de control en las condiciones ambientales o el descuido de parámetros externos que influyen directamente en las mediciones. Estos problemas son comunes en escenarios industriales, donde múltiples variables pueden afectar la calidad de las señales de los sensores. Por lo tanto, es fundamental desarrollar métodos de procesamiento de datos más eficientes que permitan superar estos problemas y asegurar una mayor confiabilidad en los resultados.

Este proyecto se centrará en el análisis de señales continuas provenientes de cuatro sensores de gas usados para detectar mezclas dinámicas de etanol y dióxido de carbono en tiempo real. Estas señales, capturadas directamente de los sensores, serán procesadas utilizando Matlab, que cuenta con funciones para el procesamiento de señales que permitirán mejorar la calidad de los datos. Matlab se usará para aplicar procesos de suavizado y normalización para eliminar el ruido de las señales y ajustar cada una de las mediciones, lo que facilitará el análisis. Estos pasos iniciales son esenciales para garantizar que las señales sean legibles y de muy buena calidad.

Además, se realizará un análisis estadístico de los datos recolectados, lo que permitirá identificar patrones y tendencias clave de las señales. Posteriormente, se implementarán técnicas como la Transformada de Fourier, que permite descomponer la señal en sus componentes frecuenciales, proporcionando información sobre las frecuencias dominantes presentes en las señales capturadas.

Adicionalmente, se aplicará la Transformada Wavelet, una herramienta que facilita el análisis simultáneo en los dominios del tiempo y la frecuencia, lo que permite una evaluación más precisa de las señales. Por último, se empleará un

tercer método de procesamiento de señales, el cual es la Transformada de Hilbert. Esta herramienta nos permite obtener la respuesta de fase mínima a partir del análisis del espectro de la señal, garantizando un correcto análisis de la señal. Con ello se demostrará la capacidad que tienen estos métodos de procesamiento de señales para brindar características importantes de las señales.

## **2 Estado del Arte**

### **2.1 Sensores de Detección de Gases**

Los sensores de detección de gases han sido ampliamente utilizados en diversas aplicaciones industriales, ambientales y de seguridad. Estos sensores detectan la presencia de gases específicos en el entorno, proporcionando datos en tiempo real que permiten un monitoreo eficaz de la calidad del aire. Existen diferentes tipos de sensores de gases, como los sensores electroquímicos, los sensores infrarrojos y los sensores semiconductores de óxidos metálicos (MOS). Los sensores MOS han demostrado ser efectivos en la detección de gases como el etanol y el dióxido de carbono debido a su alta sensibilidad y bajo costo. Sin embargo, presentan desafíos como el desgaste del material y la inestabilidad en condiciones ambientales cambiantes [1]. Además, los avances en la miniaturización de sensores y la integración con sistemas de adquisición de datos en tiempo real han permitido que estos dispositivos se utilicen en aplicaciones portátiles y móviles [2].

### **2.2 Procesamiento de Señales en Tiempo Real**

El procesamiento de señales en tiempo real es crucial en aplicaciones donde se requiere una detección instantánea y precisa de cambios en las condiciones ambientales o industriales, como la monitorización de gases. Técnicas de procesamiento como el suavizado, el filtrado de Kalman, y el filtrado adaptativo son ampliamente utilizadas para reducir el ruido y mejorar la precisión de los datos capturados por los sensores [3]. Estas técnicas permiten eliminar la variabilidad indeseada en las señales, ofreciendo una mayor claridad en la representación de los fenómenos detectados. Además, la incorporación de sistemas embebidos con capacidades de procesamiento en tiempo real ha mejorado la eficiencia de los sistemas de detección de gases, permitiendo respuestas rápidas ante condiciones peligrosas [4].

### **2.3 Análisis de Fourier**

La Transformada de Fourier (FT) es una herramienta estándar utilizada en el procesamiento de señales para descomponer una señal en sus componentes frecuenciales. En el contexto de la detección de gases, la FT permite identificar las frecuencias dominantes en las señales generadas por los sensores, lo que es útil para detectar patrones repetitivos o anomalías en las concentraciones de gases. Sin embargo, la Transformada Rápida de Fourier (FFT), una versión más

eficiente de la FT, es la técnica preferida cuando se requiere un procesamiento en tiempo real debido a su menor complejidad computacional [5]. A pesar de su utilidad, la FT presenta limitaciones en el análisis de señales no estacionarias, ya que no proporciona información sobre cómo cambian las frecuencias en el tiempo, lo que puede ser crítico en la detección de eventos transitorios [6].

## **2.4 Análisis Wavelet**

La Transformada Wavelet ha surgido como una alternativa poderosa a la FT para el análisis de señales no estacionarias, como las que se generan en la detección de gases en mezclas dinámicas. A diferencia de la FT, que solo proporciona información en el dominio de la frecuencia, la Wavelet ofrece una resolución simultánea en el dominio del tiempo y la frecuencia, lo que permite un análisis más detallado de los cambios abruptos en las señales [7]. Esto es especialmente útil en aplicaciones donde se requiere detectar eventos transitorios o cambios repentinos en la concentración de gases. Las Wavelets Discretas y Continuas han sido aplicadas con éxito en la detección de gases, mejorando la resolución temporal y la precisión de la identificación de picos de concentración [8].

## **2.5 Transformada Hilbert**

La Transformada de Hilbert es otra técnica que se utiliza en el análisis de señales, proporcionando información sobre la envolvente instantánea y la frecuencia instantánea de una señal. En el contexto de la detección de gases, esta transformada es útil para analizar señales no estacionarias y para estudiar las modulaciones presentes en las señales capturadas por los sensores. Aunque la Transformada de Hilbert se ha utilizado menos que las técnicas de Fourier y Wavelet, estudios recientes han demostrado su utilidad en aplicaciones donde es necesario caracterizar señales moduladas y detectar cambios rápidos en la frecuencia [9]. Esta transformada complementa el análisis de Fourier al proporcionar una visión diferente sobre cómo varían las señales en función del tiempo.

## **2.6 Plataformas de Software para el Procesamiento de Señales**

Entre las plataformas más populares para el procesamiento de señales destaca MATLAB, que ofrece una amplia gama de herramientas para la implementación de transformadas de Fourier, Wavelet, y Hilbert, así como técnicas de filtrado y análisis estadístico. MATLAB es ampliamente utilizado en la academia y la industria debido a su versatilidad y facilidad de uso. Además, la Signal Processing Toolbox y la Wavelet Toolbox proporcionan una integración fluida de estas técnicas para el análisis de señales en tiempo real [10]. Alternativamente, Python ha ganado popularidad en los últimos años debido a sus librerías gratuitas como SciPy y PyWavelets, que ofrecen funcionalidades similares a MATLAB para el procesamiento de señales, con la ventaja de ser de código abierto [11]. Estas plataformas permiten realizar un análisis exhaustivo de las señales obtenidas de los sensores de gas, mejorando la precisión y la capacidad de detección en tiempo real.

### 3 Conocimientos Previos

#### 3.1 Transformada de Fourier

La Transformada de Fourier es una técnica fundamental en el análisis de señales, usada para descomponer una señal temporal en sus componentes de frecuencia. Esta herramienta permite estudiar cómo varían las frecuencias en una señal, lo cual es esencial para comprender los patrones periódicos en los datos. En el caso de señales que representan fenómenos físicos, como las señales de sensores de gases, la Transformada de Fourier permite identificar las frecuencias dominantes en la mezcla de gases.

Para una señal continua  $x(t)$ , la Transformada de Fourier está dada por:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt$$

donde  $X(f)$  es la función transformada que describe cómo las distintas frecuencias contribuyen a la señal original, y  $f$  es la frecuencia de la señal.

La Transformada Inversa de Fourier permite reconstruir la señal en el dominio del tiempo mediante la siguiente expresión:

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft} df$$

Esto es útil cuando se necesita reconstruir la señal en el dominio temporal después de haber sido analizada en el dominio frecuencial.

En el contexto de sensores de gases, la Transformada de Fourier es útil para identificar patrones repetitivos en la mezcla de gases, detectar señales periódicas que puedan indicar la presencia de ciertos gases y separar el ruido de la señal útil en el análisis de las concentraciones de gases. La Transformada Rápida de Fourier (FFT), una versión más eficiente para señales discretas, es ampliamente utilizada en el procesamiento de señales digitales porque reduce la complejidad computacional de  $O(N^2)$  a  $O(N \log N)$ , lo que la hace ideal para el procesamiento en tiempo real en sistemas de detección de gases [5].

#### 3.2 Elementos Estadísticos para el Análisis de Señales

El análisis estadístico es esencial para interpretar señales en ambientes ruidosos o en presencia de múltiples factores que afectan la calidad de los datos. Entre los elementos estadísticos clave que se utilizan para describir y analizar una señal se incluyen la media, la varianza, la desviación estándar, la correlación y la autocorrelación.

La media de una señal  $x(t)$  proporciona una medida de tendencia central, que puede indicar el valor promedio de la señal. Matemáticamente, para una señal continua:

$$\mu = \frac{1}{T} \int_0^T x(t) dt$$

En el caso de una señal discreta:

$$\mu = \frac{1}{N} \sum_{n=0}^{N-1} x[n]$$

La varianza  $\sigma^2$  mide la dispersión de los valores de la señal respecto a su media, una varianza alta indica que la señal tiene una gran fluctuación en torno a su media. Para una señal continua:

$$\sigma^2 = \frac{1}{T} \int_0^T (x(t) - \mu)^2 dt$$

La desviación estándar es simplemente la raíz cuadrada de la varianza:

$$\sigma = \sqrt{\sigma^2}$$

La correlación mide la similitud entre dos señales. Si se tienen dos señales  $x(t)$  y  $y(t)$ , la correlación entre ellas en un retardo  $\tau$  se define como:

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} x(t)y(t + \tau)dt$$

La autocorrelación es un caso especial donde se mide la similitud de una señal consigo misma, lo cual es útil para identificar patrones repetitivos o detectar ruido en la señal [12].

### 3.3 Transformada Wavelet

La Transformada Wavelet es una herramienta que permite analizar señales no estacionarias, es decir, aquellas cuya frecuencia cambia en el tiempo. Mientras que la Transformada de Fourier solo proporciona información sobre las frecuencias presentes en una señal, la Wavelet ofrece información tanto en el dominio de la frecuencia como en el del tiempo.

La Transformada Wavelet Continua (CWT) de una señal  $x(t)$  se define como:

$$W_x(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t)\psi^*\left(\frac{t-b}{a}\right) dt$$

donde  $\psi$  es la wavelet madre,  $a$  es el factor de escala, que ajusta la dilatación o compresión de la wavelet, y  $b$  es el parámetro de traslación en el tiempo. La Transformada Wavelet Discreta (DWT) es una versión simplificada y computacionalmente eficiente, que se aplica a señales discretas. Permite descomponer la señal en distintos niveles de resolución, facilitando la captura de detalles finos y tendencias de largo plazo [7].

La principal ventaja de las Wavelets es su capacidad para capturar tanto detalles locales (alta frecuencia) como tendencias globales (baja frecuencia) de una señal. Además, las Wavelets permiten analizar la señal a diferentes escalas, proporcionando una visión más completa de la dinámica del sistema.

En el contexto de la detección de gases, la Transformada Wavelet se utiliza para detectar cambios abruptos en las concentraciones de gases, identificar eventos transitorios que no serían visibles con la Transformada de Fourier y proporcionar un análisis multiescala que revela tanto detalles de alta frecuencia (ruido o fluctuaciones rápidas) como tendencias de baja frecuencia (cambios generales en la concentración de gases) [8].

### 3.4 Transformada de Hilbert

La Transformada de Hilbert proporciona una forma de analizar las características instantáneas de una señal, como su envolvente y fase instantánea. La transformada de Hilbert es útil para el análisis de señales moduladas y no estacionarias, como aquellas que se observan en muchos sistemas de sensores.

La Transformada de Hilbert de una señal  $x(t)$  está dada por:

$$\hat{x}(t) = \frac{1}{\pi} P.V. \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau$$

donde P.V. denota el valor principal de Cauchy, que asegura que la integral converge. La señal analítica asociada  $z(t)$  se define como:

$$z(t) = x(t) + j\hat{x}(t)$$

La magnitud de esta señal analítica  $z(t)$  proporciona la envolvente de la señal, mientras que el argumento de  $z(t)$  da la fase instantánea:

$$A(t) = |z(t)| = \sqrt{x^2(t) + \hat{x}^2(t)}$$

$$\phi(t) = \arg(z(t)) = \tan^{-1} \left( \frac{\hat{x}(t)}{x(t)} \right)$$

La Transformada de Hilbert es especialmente útil en el análisis de señales moduladas, obtener la envolvente instantánea que es útil para detectar cambios graduales o bruscos en las concentraciones de gases, y proporcionar una representación instantánea de la fase de una señal, lo cual es crucial en sistemas de detección donde los cambios de fase son indicadores de la presencia de ciertos gases [13].

## 4 Metodología

El proyecto se basará en la selección y análisis de una o dos bases de datos (BD) provenientes de fuentes accesibles en internet o de páginas específicas mencionadas en clase. La elección de la base de datos será libre, permitiendo flexibilidad en función de los intereses y preferencias del investigador. Se recomienda optar por bases de datos en las que cada caso represente una señal continua, como pueden ser lecturas de radiación, ondas cerebrales, señales satelitales, entre otras.

La metodología se desarrollará en dos fases principales: preprocesamiento de datos y análisis avanzado de señales, utilizando MATLAB como herramienta clave para la implementación de los algoritmos de procesamiento de señales.

#### 4.1 Preprocesamiento de los Datos

En la primera fase, el enfoque estará en la preparación y limpieza de la base de datos seleccionada. Esto incluirá las siguientes tareas:

**Limpieza de los datos:** Se procederá a la identificación y corrección de datos erróneos o inconsistentes, así como a la definición de valores faltantes. Se podrán utilizar técnicas de imputación de valores o eliminar los casos que no cumplan con los requisitos de calidad. Esta etapa será asistida por herramientas de análisis de datos en MATLAB, que ofrecen funciones para detección de valores atípicos y tratamiento de datos faltantes.

**Normalización:** Cada caso de la base de datos será normalizado, ajustando los valores de las señales a un rango entre 0 y 1. MATLAB será utilizado para realizar este proceso de normalización mediante las funciones de escalamiento y transformación de datos, lo cual garantizará la homogeneidad de las señales y permitirá un análisis más preciso.

**Análisis Estadístico:** Se determinarán estadísticas descriptivas de los datos, tales como media, mediana, moda, desviación estándar, entre otras, utilizando funciones integradas de MATLAB como mean, median, mode y std. Si los casos de la base de datos representan señales continuas, se procederá a calcular el número de cruces por cero y los cambios de signo en las señales, lo cual proporcionará información relevante sobre la dinámica de las señales.

**Visualización de las señales:** Para facilitar la comprensión de los datos, se procederá a graficar las señales en su estado original, así como las señales después del proceso de normalización. MATLAB, a través de sus herramientas de visualización como plot y subplot, permitirá la generación de gráficos que comparen las señales antes y después del preprocesamiento, destacando cualquier cambio en su comportamiento.

#### 4.2 Análisis de las Señales

La segunda fase del proyecto estará enfocada en el análisis de las señales continuas de la base de datos seleccionada. Se aplicarán diversas técnicas de procesamiento de señales con el fin de extraer características clave de las mismas, utilizando MATLAB para la implementación de los scripts y funciones.

**Aplicación de la Transformada de Fourier:** Las señales continuas serán descompuestas en sus componentes frecuenciales utilizando la Transformada de Fourier, implementada mediante la función fft de MATLAB. Esto permitirá identificar las frecuencias dominantes y patrones periódicos presentes en los datos.

**Aplicación de la Transformada Wavelet:** Posteriormente, se aplicará la Transformada Wavelet a las señales utilizando la Wavelet Toolbox de MATLAB. Esta transformada permitirá un análisis multiresolución en los dominios del tiempo y la frecuencia, facilitando la identificación de eventos transitorios o cambios bruscos en las señales. La función wavedec será clave para este análisis.

**Tercer Algoritmo de Procesamiento:** Se implementará un tercer algoritmo de procesamiento de señales a elección del investigador, que podrá ser seleccionado

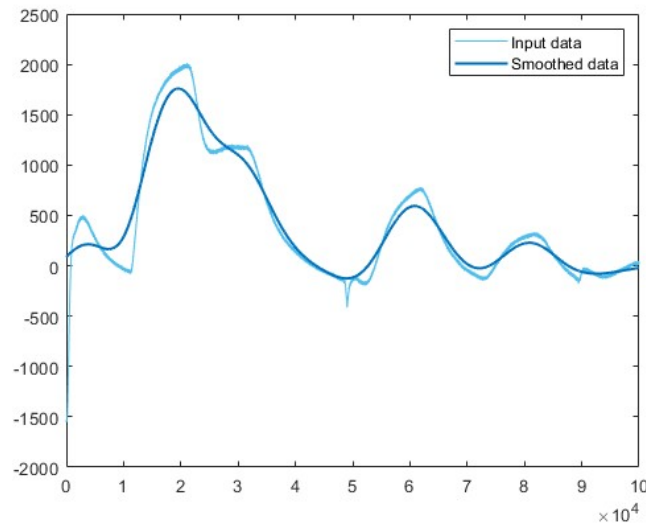


en función de las características particulares de la base de datos. Entre las opciones sugeridas se encuentran la Transformada de Hilbert (utilizando la función hilbert de MATLAB), filtrado de Kalman o cualquier otro método adecuado para la naturaleza de las señales analizadas.

## 5 Resultados de las Señales Procesadas

### 5.1 Señal Original y Suavizada

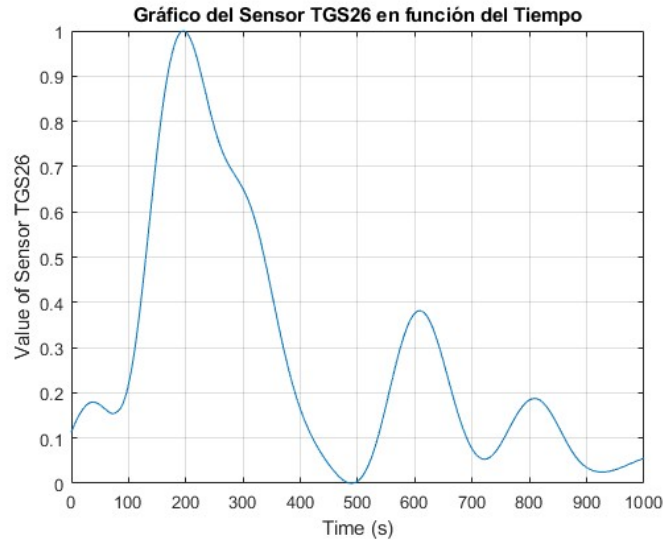
La siguiente imagen muestra la señal original proveniente del sensor TGS2602, junto con la señal suavizada usando el suavizado gaussiano con un factor de suavización de 0.25. Como se observa en la gráfica, la señal suavizada ha reducido las oscilaciones rápidas de la señal original, eliminando el ruido de alta frecuencia mientras preserva las tendencias generales de la señal.



**Fig. 1.** Señal original y suavizada del sensor TGS2602.

### 5.2 Señal Normalizada

En esta imagen se presenta la señal del sensor TGS2602 después del proceso de normalización. La normalización ajusta los valores de la señal entre 0 y 1, lo que permite una comparación adecuada con otras señales procesadas de forma similar.



**Fig. 2.** Señal normalizada del sensor TGS2602.

### 5.3 Transformada de Fourier

La Transformada de Fourier descompone la señal del sensor en sus componentes de frecuencia. La gráfica resultante muestra el espectro de frecuencias, donde se destacan las frecuencias dominantes. Esto es útil para identificar patrones periódicos o la presencia de ruido en frecuencias específicas.

### 5.4 Transformada Wavelet

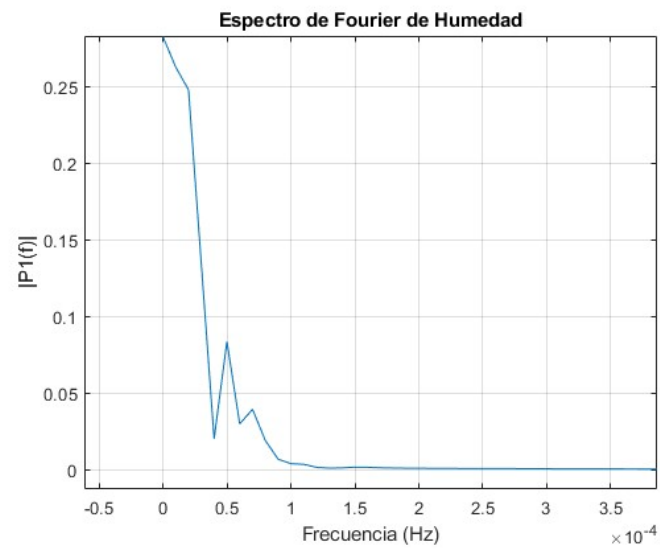
La Transformada Wavelet permite una representación multiescalar de la señal del sensor. A diferencia de la Transformada de Fourier, que analiza solo las frecuencias, la Transformada Wavelet proporciona información simultánea sobre el tiempo y la frecuencia, permitiendo observar cambios abruptos o transitorios en la señal.

### 5.5 Transformada de Hilbert

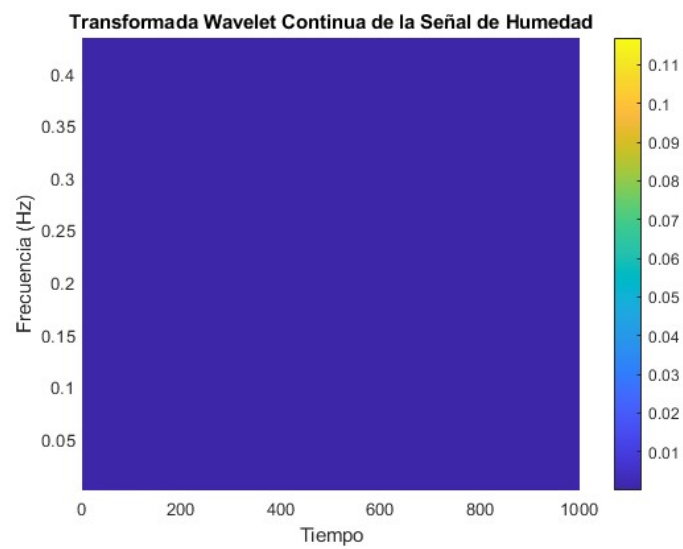
La Transformada de Hilbert se utiliza para analizar la envolvente de la señal y su fase. En la siguiente imagen se puede observar la envolvente de la señal, lo cual permite estudiar modulaciones presentes y cambios en la amplitud a lo largo del tiempo.

### 5.6 Comparación Final de Procesamientos

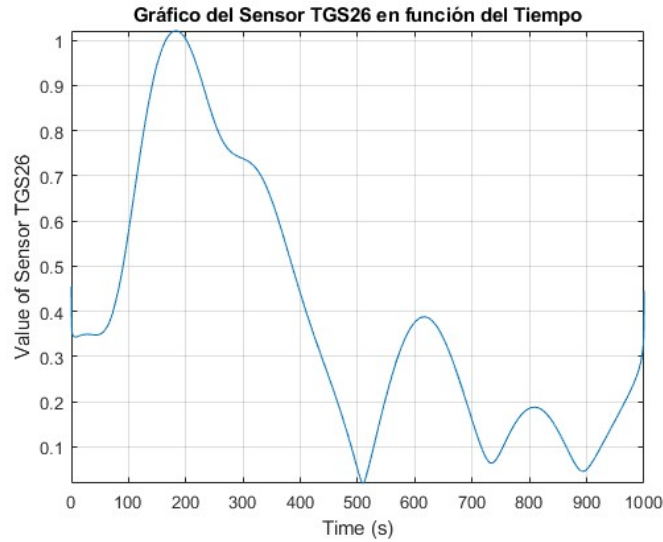
Finalmente, se comparan los diferentes métodos aplicados (Transformada de Fourier, Transformada Wavelet y Transformada de Hilbert) para resaltar las



**Fig. 3.** Transformada de Fourier de la señal del sensor TGS2602.

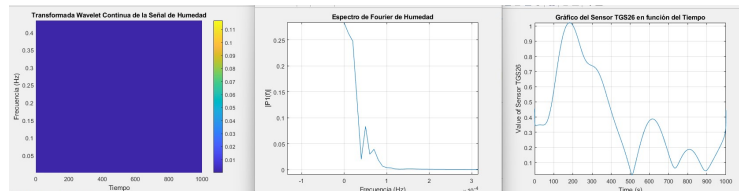


**Fig. 4.** Transformada Wavelet de la señal del sensor TGS2602.



**Fig. 5.** Transformada de Hilbert de la señal del sensor TGS2602.

diferencias en la información obtenida a través de cada técnica. Mientras que la Transformada de Fourier revela información sobre las frecuencias dominantes, la Transformada Wavelet captura eventos transitorios, y la Transformada de Hilbert ofrece detalles sobre la modulación de la señal.



**Fig. 6.** Comparación de diferentes técnicas de procesamiento de la señal del sensor TGS2602.

## 5.7 Análisis Estadístico de la Señal

En esta tabla se presentan los resultados del análisis estadístico de la señal proveniente del sensor TGS2602. Los parámetros estadísticos incluyen la media, mediana, moda, desviación estándar, varianza, rango, valor máximo, valor mínimo y el rango intercuartílico (IQR). Estos valores ofrecen una visión detallada del comportamiento de la señal, permitiendo identificar tendencias, dispersión y posibles valores atípicos.

**Table 1.** Análisis Estadístico de la Señal del Sensor TGS2602

Parámetro Estadístico	Valor
Media	406.8689
Mediana	196.9601
Moda	-125.3737
Desviación Estándar	530.7438
Varianza	281688.9748
Rango	1885.2982
Valor Máximo	1759.9244
Valor Mínimo	-125.3737
IQR (Rango Intercuartílico)	586.4340

## 6 Conclusión

En este trabajo se ha presentado un análisis exhaustivo de las señales obtenidas de los sensores de gases, específicamente del sensor TGS2602, utilizando diversas técnicas de preprocesamiento y procesamiento de señales. A través del proceso de normalización y suavizado, se logró mejorar la calidad de los datos, permitiendo una representación más clara de las tendencias subyacentes en la señal. Las transformadas de Fourier, Wavelet y Hilbert permitieron un análisis detallado de las características frecuenciales y temporales de la señal, brindando una visión profunda de los eventos transitorios y las modulaciones presentes.

El análisis estadístico reveló información clave sobre la distribución de la señal, destacando la variabilidad y la dispersión de los datos a través de parámetros como la media, mediana, moda, desviación estándar y varianza. Estos resultados proporcionan una base sólida para futuros estudios, ya que permiten identificar patrones y anomalías en las señales que pueden estar vinculadas con la detección de gases en tiempo real.

## A Anexos

En esta sección se incluyen los códigos utilizados en el desarrollo del proyecto, tanto en Python como en MATLAB.

### A.1 Código en Python

El siguiente código en Python se utilizó para el preprocesamiento de las imágenes, convirtiéndolas a escala de grises, aplicando un filtro Gaussiano y binarizando las imágenes utilizando el método de Otsu.

[language=Python, caption=Código Python para el preprocesamiento de imágenes]

```
import cv2
import os
import numpy as np
```

```
Lista de subcarpetas (una por cada verdura) subfolders = ['Bean', 'BitterGourd', 'BottleGourd', 'Brinjal', 'Esrequerdo', 'Estare', 'Enelpathdondeseencuentranlas carpetas de las imágenes', 'Rutadelacarpetadondeestánlas imágenes', 'train_img', 'Rutadelacarpetadonde se guardarán las imágenes binarizadas', 'output_root_folder']
bin_train_img =
```

```

Procesar cada carpeta for subfolder in subfolders: Definir la ruta de la car-
peta de entrada y salida  $input\_folder = os.path.join(input\_root\_folder, subfolder)$   $output\_folder =$ 
 $os.path.join(output\_root\_folder, subfolder)$ 
Obtener la lista de archivos de imagen en la carpeta de entrada  $image\_files =$ 
 $[f \text{ for } f \text{ in } os.listdir(input\_folder) \text{ if } f.endswith('.jpg') \text{ or } f.endswith('.png')]$ 
Procesar cada imagen en la carpeta for i,  $image\_file \text{ in enumerate}(image\_files)$  :
Leer la imagen  $image\_path = os.path.join(input\_folder, image\_file)$   $image = cv2.imread(image\_path)$ 
Convertir la imagen a escala de grises  $gray\_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)$ 
Aplicar filtro Gaussiano para eliminar ruido  $blurred\_image = cv2.GaussianBlur(gray\_image, (5, 5), 0)$ 
Binarizar la imagen utilizando el método de Otsu  $binary\_image = cv2.threshold(blurred\_image, 0, 255, cv2.$ 
 $cv2.THRESH_OTSU)$ 
Crear el nombre del archivo de salida  $output\_image\_path = os.path.join(output\_folder, f'_{subfolder.lower()}_{i}.png')$ 
Guardar la imagen binarizada  $cv2.imwrite(output\_image\_path, binary\_image)$ 
Mostrar progreso  $print(f'Imagen {i+1}/{len(image\_files)} de la carpeta {subfolder} procesada y guardada en {output\_folder}')$ 

```

## A.2 Código en MATLAB

El siguiente código en MATLAB se utilizó para redimensionar las imágenes, convertirlas en vectores fila, entrenar el perceptrón y controlar el Arduino mediante el paquete de soporte de MATLAB para hardware Arduino.

```

[language=Matlab, caption=Código MATLAB para el entrenamiento del per-
ceptrón y control del Arduino] cd('C:NN')
root_folder = 'bin_train_img'; subfolders = 'Bean', 'Bitter_Gourd', 'Bottle_Gourd', 'Brinjal', 'Broccoli', 'Ca-
numel(subfolders);
X_train = []; y_train = []; image_size = 64 * 64;
for class = 1:num_classes
    train_folder = fullfile(root_folder, subfolders{class}); image_files =
    dir(fullfile(train_folder, '*.png')); num_images = numel(image_files);
    for i = 1:num_images
        image_path = fullfile(train_folder, image_files(i).name); img =
        imread(image_path); img = imresize(img, [64, 64]); img = double(img);
        img_vector = (img(:))';
        img_vector = img_vector/255;
        X_train = [X_train; img_vector];
        y_train = [y_train; class]; end
    num_inputs = image_size; num_outputs = 4;
    net = perceptron; net = configure(net, ones(num_inputs, 1), ones(num_outputs, 1));
    y_train_bin = double(dec2bin(y_train - 1, num_outputs) - '0');
    net.trainParam.epochs = 10000; net = train(net, X_train', y_train_bin');
    test_folder = 'bin_img_val/'; X_test = []; y_test = [];
    for class = 1:num_classes
        test_subfolder = fullfile(test_folder, subfolders{class}); test_files =
        dir(fullfile(test_subfolder, '*.png')); num_test_images = numel(test_files);
        for i = 1:num_test_images
            test_image_path = fullfile(test_subfolder, test_files(i).name); img_test =
            imread(test_image_path); img_test = imresize(img_test, [64, 64]); img_test_vector =
            img_test(:)'; X_test = [X_test; img_test_vector];
            y_test = [y_test; class]; end
        y_pred_bin = net(X_test');

```

```

y_pred = bin2dec(num2str(y_pred_bin));
y_pred = y_pred + 1;
disp('Etiquetas verdaderas:'); disp(y_test); disp('Predicciones del Perceptrón : '
); disp(y_pred);
accuracy = sum(y_pred == y_test)/numel(y_test); fprintf('Precisión del Perceptrón :
clear a; a = arduino('COM8','Uno');
outputPins = 'D7','D6','D5','D4';
for i = 1:length(outputPins) configurePin(a, outputPins(i), 'DigitalOutput');
end
disp('Listo para cargar y clasificar una imagen. ');
[file, path] = uigetfile('*.*png;*.jpg;*.jpeg', 'Archivos de Imagen (*.png, *.jpg,
*.jpeg)');
if isequal(file, 0) disp('No se seleccionó ninguna imagen'); return; end
imagen_path = fullfile(path, file); img = imread(imagen_path);
img_resized = imresize(img, [64, 64]);
if isa(img_resized, 'uint8') img_normalized = double(img_resized)/255; else img_normalized =
img_resized; end
img_vector = img_normalized(:)';
y_pred_onehot = net(img_vector');
[~, y_pred_class] = max(y_pred_onehot);
if y_pred_class > 0 y_pred_class <= numel(subfolders) figure, imshow(img_resized), title(['Clase predicha : '
, subfolders{y_pred_class}]); else disp('Predicción fuera de rango. Verifica los datos de salida. '); end
class_bin = dec2bin(y_pred_class - 1, 4);
disp(['Clase predicha en binario (4 bits): ', class_bin]);
for i = 1:4 bit_value = str2double(class_bin(i)); writeDigitalPin(a, outputPins(i), bit_value); end
pause(10);
for i = 1:length(outputPins) writeDigitalPin(a, outputPins(i), 0); end

```

## Referencias

## References

1. L. Ma, Z. Zhang, Y. Zhang, and C. Jia, "A review of gas sensor technologies for environmental monitoring," *Sensors*, vol. 19, no. 23, pp. 1–17, Dec. 2019.
2. S. K. Sharma, J. C. Prasad, and P. Singh, "Development of real-time gas sensing system using miniature gas sensors," *IEEE Sensors Journal*, vol. 19, no. 15, pp. 6068–6075, Aug. 2019.
3. P. Shah, S. B. Patil, and V. M. Wankhede, "Real-time gas sensor signal processing using kalman filter and sensor fusion," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 8, no. 6, pp. 74–80, Jun. 2020.
4. B. Kouvarakis, A. Simitsis, and C. Arvanitakis, "Embedded real-time signal processing system for industrial gas monitoring," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8493–8501, Oct. 2018.
5. A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd ed. New York, NY: Pearson, 2010.

6. M. Prasad and R. K. Tripathi, "Comparison of fourier and wavelet transforms for non-stationary signal analysis," *International Journal of Computer Applications*, vol. 118, no. 8, pp. 24–28, May 2015.
7. I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.
8. M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
9. A. P. Liavas, P. Papakonstantinou, and G. B. Giannakis, "A hilbert transform approach to signal analysis and its applications in communications and biomedical engineering," *IEEE Transactions on Signal Processing*, vol. 45, no. 12, pp. 3084–3093, Dec. 1997.
10. MathWorks, "Signal processing toolbox: User's guide," 2023, [Online]. Available: <https://www.mathworks.com/help/signal/>.
11. E. Jones, T. Oliphant, P. Peterson *et al.*, "Scipy: Open source scientific tools for python," 2023, [Online]. Available: <https://scipy.org/>.
12. S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Upper Saddle River, NJ: Prentice-Hall, 1993.
13. D. Gabor, "Theory of communication," *Journal of the Institution of Electrical Engineers*.