UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escuela Técnica Superior de Ingeniería de Telecomunicación

# Implementación de un sistema de Control Activo de Ruido Multicanal usando redes neuronales con entrenamiento on-line basado en la técnica de retro-propagación.

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

AUTOR/A: Lozano Marin, Lola

Tutor/a: Diego Antón, María de

Cotutor/a: Ferrer Contreras, Miguel

CURSO ACADÉMICO: 2023/2024

**NO PONER PORTADA**

## Resumen

Este TFG tiene como objetivo analizar y valorar estrategias de aprendizaje automático frente a los métodos clásicos basados en el uso de algoritmos adaptativos convencionales para resolver el problema de control activo de ruido multicanal.

Se propone una implementación basada en redes neuronales para actualizar los coeficientes del filtro, tratándolos como los pesos de una red neuronal. Esta aproximación modifica el algoritmo de filtrado-x LMS (FxLMS) para adaptarlo a una red neuronal. La adaptación de los coeficientes se logrará mediante técnicas de aprendizaje automático utilizando algoritmos de entrenamiento de retropropagación.

Este enfoque se implementará en MATLAB, aprovechando sus herramientas de aprendizaje automático, para explorar cómo las redes neuronales pueden conseguir resultados de cancelación de ruido en un sistema multicanal de forma similar a los clásicos algoritmos adaptativos.

## Resum

El TFG té com a objectiu analitzar i valorar estratègies d'aprenentatge automàtic enfront dels mètodes clàssics basats en l'ús d'algoritmes adaptatius convencionals per a resoldre el problema de control actiu de soroll multicanal.

Es proposa una implementació basada en xarxes neuronals per a actualitzar els coeficients del filtre, tractant-los com els pesos d'una xarxa neuronal. Aquesta aproximació modifica l'algoritme de filtratge-x LMS (FxLMS) per a adaptar-lo a una xarxa neuronal. L'adaptació dels coeficients s'aconseguirà mitjançant tècniques d'aprenentatge automàtic utilitzant algoritmes d'entrenament de retropropagació.

Aquest enfocament s'implementarà en MATLAB, aprofitant les seues eines d'aprenentatge automàtic, per a explorar com les xarxes neuronals poden aconseguir resultats de cancel·lació de soroll en un sistema multicanal de forma similar als clàssics algoritmes adaptatius.

## Abstract

The thesis aims to analyze and evaluate machine learning strategies against classical methods based on the use of conventional adaptive algorithms to solve the problem of multichannel active noise control.

An implementation based on neural networks is proposed to update the filter coefficients, treating them as the weights of a neural network. This approach modifies the filtered-x LMS algorithm (FxLMS) to adapt it to a neural network. The adaptation of the coefficients will be achieved through machine learning techniques using back-propagation training algorithms.

This approach will be implemented in MATLAB, leveraging its machine learning tools, to explore how neural networks can achieve noise cancellation results in a multichannel system similarly to classical adaptive algorithms.

# RESUMEN EJECUTIVO

La memoria del TFG de 'Implementación de un sistema de Control Activo de Ruido Multicanal usando redes neuronales con entrenamiento on-line basado en la técnica de retro-propagación' debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la Ingeniería de Telecomunicación

| **CONCEPT (ABET)** | **CONCEPTO (traducción)** | **¿Cumple? (S/N)** | **¿Dónde? (páginas)** |
|---|---|---|---|
| 1. IDENTIFY: | 1. IDENTIFICAR: | S | 1-19 |
| 1.1. Problem statement and opportunity | 1.1. Planteamiento del problema y oportunidad | S | 1-20 |
| 1.2. Constraints (standards, codes, needs, requirements & specifications) | 1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones) | S | 19,23 |
| 1.3. Setting of goals | 1.3. Establecimiento de objetivos | S | 1,2 |
| 2. FORMULATE: | 2. FORMULAR: | S | 19-23 |
| 2.1. Creative solution generation (analysis) | 2.1. Generación de soluciones creativas (análisis) | S | 23-33 |
| 2.2. Evaluation of multiple solutions and decision-making (synthesis) | 2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis) | S | 24,25 |
| 3. SOLVE: | 3. RESOLVER: | S | 24-30 |
| 3.1. Fulfilment of goals | 3.1. Evaluación del cumplimiento de objetivos | S | 27-30 |
| 3.2. Overall impact and significance (contributions and practical recommendations) | 3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas) | S | 30 |

# Índice

# Índice de ilustraciones

# Capítulo 1. Introducción al proyecto

El control activo de ruido (ANC) [1] representa una de las soluciones más innovadoras para mitigar el ruido indeseado, utilizando tecnologías que involucran la generación de una señal acústica que "cancela" el ruido original mediante interferencia destructiva. Tradicionalmente, los sistemas de ANC han dependido de algoritmos adaptativos basados en técnicas como el algoritmo de filtrado-x LMS (FxLMS) para ajustar los coeficientes de los filtros encargados de generar las señales antirruido. Sin embargo, los avances de la inteligencia artificial y el aprendizaje automático abren nuevas vías para explorar en el ámbito del ANC.

Este trabajo propone el desarrollo de un sistema de control activo de ruido multicanal que utiliza redes neuronales. La propuesta reinterpreta el algoritmo de FxLMS en el contexto de una red neuronal, tratando los coeficientes del filtro como si fueran pesos neuronales. Esta integración permite emplear técnicas de aprendizaje automático avanzadas, como la retropropagación, para ajustar los coeficientes de manera eficiente en un entorno multicanal. En este enfoque, se instruye a una red neuronal para que se adapte utilizando la misma lógica que el algoritmo de FxLMS modificado, esto es, resolviendo un problema de minimización mediante un algoritmo de gradiente.

El desafío principal reside en aprender a usar las herramientas proporcionadas para solucionar un problema que tradicionalmente se ha solucionado mediante la implementación de filtros adaptativos, empleando redes neuronales. En una primera etapa se ha llevado a cabo un estudio previo, que ha servido para definir adecuadamente la arquitectura de la red en el contexto de un problema de ANC y para determinar qué datos deben ser suministrados como predictores y objetivo de manera que la red emule de manera efectiva las tareas que haría un filtro adaptativo.

## 1.1 Motivación

La motivación principal para la elección de este proyecto ha sido la de obtener conocimientos sobre las redes neuronales, materia que no se ha estudiado durante el grado cursado, pero accesible con los conocimientos adquiridos en éste, especialmente en el campo del *Machine Learning* y la inteligencia artificial. Por otro lado, se plantea analizar las posibilidades que las técnicas de aprendizaje automático brindan en la resolución de un problema clásico como es el ANC, que tradicionalmente se resuelve mediante algoritmos adaptativos.

## 1.2 Objetivos

El objetivo principal de este proyecto es desarrollar y validar un sistema de control activo de ruido multicanal mediante el uso de redes neuronales para adaptar los coeficientes de los filtros, con el fin de estudiar el posible uso de técnicas de aprendizaje automático para la cancelación de ruido en comparación con los algoritmos adaptativos tradicionales.

Para lograr este objetivo, primero se realizará un estudio exhaustivo de los sistemas clásicos de control de ruido, centrándose particularmente en aquellos que utilizan configuraciones multicanal y algoritmos como el FxLMS. Se definirá un entorno experimental que incluya múltiples altavoces y sensores para simular condiciones realistas de ruido y evaluar la efectividad de la cancelación.

A continuación, se diseñará y desarrollará una variante del algoritmo de FxLMS adaptado a las condiciones del problema, al que se le integrará una red neuronal que permita la adaptación automática de los coeficientes del filtro. Este novedoso sistema será comparado con los métodos clásicos para verificar su eficiencia en la reducción de ruido. Este enfoque integral no solo busca reinventar técnicas para la tecnología de control de ruido sino también explorar aplicaciones prácticas de las redes neuronales en sistemas de control en tiempo real.

## 1.3 Metodología

La metodología para desarrollar el sistema de control activo de ruido se centra en el diseño, implementación y validación de una red neuronal entrenada en tiempo real. Para alcanzar los objetivos propuestos, ha sido imprescindible realizar un estudio previo sobre la cancelación de ruido y sobre las redes neuronales. Este estudio se detalla en los primeros capítulos de esta memoria y sirve como base teórica del trabajo.

La herramienta seleccionada para diseñar el sistema de control y manejar las redes neuronales es MATLAB, debido a sus toolboxes y librerías especializadas en redes neuronales que permiten al usuario configurar y personalizar los parámetros del sistema de forma eficiente.

La efectividad del sistema se mide por la disminución de la potencia de las señales de error, entendida como su valor cuadrático medio instantáneo, que debería aproximarse a cero.

# Capítulo 2. Introducción al problema de Cancelación de Ruido

La tecnología de cancelación de ruido ha transformado nuestra experiencia auditiva al permitirnos sumergirnos en un entorno más silencioso al bloquear los sonidos no deseados del entorno.

El ANC involucra un sistema electroacústico o electromecánico [2] que elimina el ruido indeseado empleando el principio de superposición acústica. En particular, esto es generando una señal antirruido con la misma amplitud pero en fase opuesta, que se combina con la señal de ruido original para anularse ambas señales entre sí.

Por otro lado, el control pasivo de ruido (PNC) se centra en es aislar acústicamente la fuente de ruido, mediante el aislamiento del entorno o el uso de materiales absorbentes de ondas sonoras.

## 2.1 Cancelación de ruido Pasiva

La PNC se fundamenta en los principios físicos que rigen la interacción entre los sonidos y los materiales. Los sistemas que utilizan esta tecnología están diseñados de manera que las señales de ruido externas son bloqueadas físicamente, disminuyendo así el nivel de ruido que llega a los usuarios. Esto se logra mediante la construcción de barreras físicas que emplean materiales densos, dispuestos de manera adecuada para absorber y bloquear el sonido no deseado.

Sin embargo, las limitaciones de los métodos pasivos en bajas frecuencias son un impedimento [3], ya que los materiales aislantes, las barreras antirruido, los resonadores de Helmholtz y los encapsulamientos de fuentes de ruido suelen requerir dimensiones y pesos que resultan impracticables por debajo de los 500 Hz, es decir, aunque el PNC es efectivo en bajas frecuencias, el problema radica en la magnitud de las soluciones requeridas. A estas desventajas se le suman otros problemas como la falta de control sobre el campo acústico residual final y la poca flexibilidad de estos métodos para adaptarse a diferentes entornos o condiciones variantes en el tiempo.

La cancelación pasiva de ruido se utiliza ampliamente en una gran variedad de aplicaciones. Los auriculares comerciales son los dispositivos más comunes [4] que incorporan esta tecnología para mejorar la calidad del audio en entornos ruidosos. Además, algunos equipos de protección auditiva utilizan características de cancelación pasiva de ruido para reducir el ruido ambiental y mejorar la audición en entornos laborales con mucha contaminación acústica.

## 2.2 Cancelación de ruido Activa

El proceso de ANC se basa en el principio de superposición reflejado en la figura 1, donde se crea una señal antirruido con la misma amplitud pero fase inversa a la señal de ruido principal, neutralizándolo eficazmente. La ANC es particularmente efectivo para reducir ruidos de baja frecuencia. Gracias a estas ventajas, la ANC ha evolucionado rápidamente, ofreciendo reducciones significativas en tamaño, peso y coste de las soluciones.

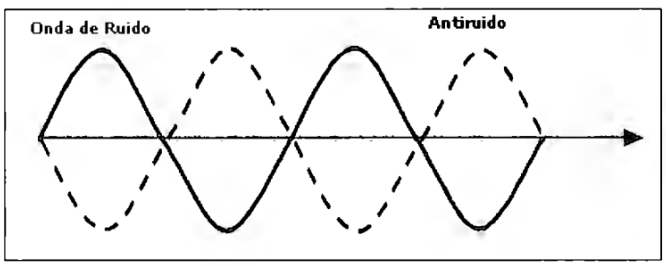


**Figura 1. Principio de superposición**

El concepto de ANC acústico se introdujo por primera vez en una patente de 1936 por Paul Lueg [5], que propuso el uso de un micrófono y un altavoz controlado eléctricamente para generar un sonido de cancelación.

Los sistemas ANC adaptativos son preferibles si las propiedades del ruido, como la frecuencia, amplitud, fase y velocidad del sonido, varían con el tiempo. Los filtros adaptativos, como los de respuesta al impulso finita (FIR) y los de respuesta al impulso infinita (IIR), ajustan continuamente sus coeficientes para minimizar parámetros como la potencia media, la potencia instantánea o la diferencia entre la señal generada y la señal objetivo.

La implementación de técnicas de procesado digital de señal en los sistemas de ANC ha transformado su efectividad y eficiencia. [6] Gracias a los avances tecnológicos en microchips durante la década de 1980, fue posible adoptar algoritmos adaptativos avanzados de manera económica. Estos algoritmos permiten que las señales captadas por transductores se procesen digitalmente, mejorando significativamente la capacidad de los sistemas ANC para ajustarse dinámicamente a diferentes condiciones de ruido.

En el control activo, se pueden distinguir cuatro elementos principales [7] :

- La planta, se refiere al sistema físico o canal que se desea controlar, como pueden ser el ruido de una maquina industrial.
- Los sensores, pueden ser micrófonos u otros sensores, tienen la función de supervisar el comportamiento del sistema de ANC.
- Los actuadores, que son dispositivos como altavoces o generadores de vibraciones, que intervienen directamente para modificar la respuesta de la planta.
- El controlador, se trata de un procesador de señales, generalmente digital, encargado de coordinar las acciones de los actuadores en función de la información proporcionada por los sensores en todo momento.

En la figura 2 el ruido original de la fuente es recogido por un micrófono que actúa como una referencia de ruido. Esta señal es luego procesada por un filtro que, basado en los parámetros que recibe del algoritmo adaptativo, genera un "ruido cancelador" (señal antirruido). Este ruido es emitido en todo el espacio y tiene la característica especial de estar en fase opuesta al ruido original, lo que provoca que, al encontrarse ambas señales, se cancelen mutuamente, resultando una reducción significativa del ruido percibido. [8]

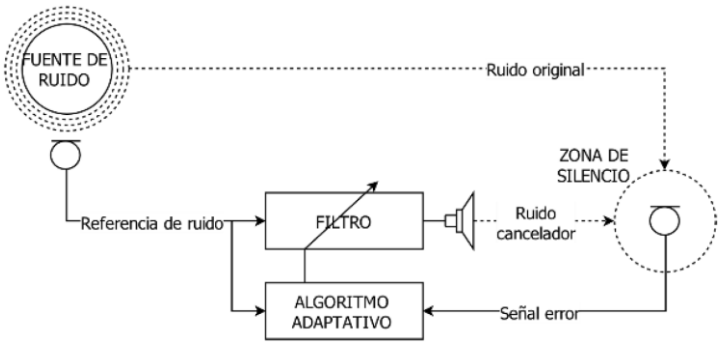


**Figura 2. Esquema general de control activo de ruido**

El algoritmo adaptativo juega un papel crucial en la cancelación activa de ruido, permitiendo un ajuste del filtro en tiempo real. Este proceso dinámico asegura que el sistema de cancelación se mantenga efectivo incluso cuando las características del ruido ambiental cambian.

### *2.2.1 Algoritmo adaptativo LMS*

El algoritmo Least-Mean-Square (LMS) fue propuesto por Widrow y Hoff [9] en 1960 como un filtro adaptativo diseñado para encontrar la solución de mínimos cuadrados de forma iterativa y estocástica, enfocado en optimizar cierta función de error cuadrático. Este algoritmo pertenece a los algoritmos de gradiente estocástico, lo que significa que adapta sus coeficientes basándose solo en el error del momento, sin necesidad de calcular funciones de correlación ni invertir matrices de correlación, simplificando así su implementación.

El filtro adaptativo LMS es una herramienta clave en procesamiento de señales, usada especialmente para identificar sistemas y cancelar ruido no deseado. Este tipo de filtro se ajusta de manera continua a las características cambiantes de la señal de entrada, mejorando la precisión de la cancelación de ruido a lo largo del tiempo. Funciona ajustando los coeficientes del filtro de forma iterativa para minimizar el error cuadrático instantáneo entre la señal deseada y la salida del filtro. En cada momento, el filtro actualiza sus coeficientes según la ecuación 1, de manera que el error entre la salida real del filtro y la señal deseada se reduzca al mínimo.

$$w(n+1) = w(n) + \mu \cdot e(n) \cdot x(n) \tag{1}$$

Donde $w(n)$ representa los coeficientes del filtro, $\mu$ es el paso adaptativo, y las señales $x(n)$ y $e(n)$ son la señal de entrada al filtro y la señal de error respectivamente.

Sin embargo, es importante señalar que el algoritmo LMS no puede emplearse directamente en sistemas de control de ruido activo debido a la presencia de la planta acústica entre el actuador y el punto de control, ya que el algoritmo LMS no tiene esta en cuenta. Por ello, para adaptarse a las necesidades específicas del control de ruido activo, se desarrollan alternativas como el algoritmo Filtered-x LMS, que está específicamente diseñado para manejar la interacción dinámica entre los actuadores y los sensores en un entorno acústico más complejo.

### *2.2.2 Algoritmo Multichannel Filtered-x LMS*

Para poder solucionar un caso de control activo de ruido, se debe hacer una modificación en el algoritmo tradicional, ya que este solo tiene en cuenta un camino acústico que introduce ruido, pero en un entorno real existiría otra perturbación acústica entre la salida del filtro y el micrófono donde se cancela el ruido. Una variante del algoritmo LMS es el algoritmo Filtered-x LMS, que se ilustra en la figura 3. Este algoritmo en su versión multicanal puede observarse en un contexto donde interactúan múltiples señales de referencia y varios filtros de control en un entorno de control activo de ruido. En este sistema se puede observar que contribuyen a la introducción de ruido 2 caminos acústicos, el camino primario y el camino secundario, denotado como $H_{jk}$.

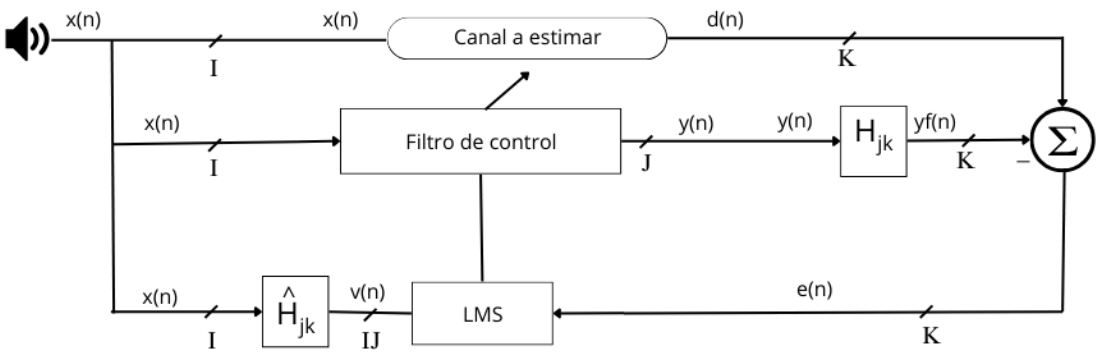


**Figura 3. Esquema de un sistema ANC multicanal IxJxK mediante el algoritmo FxLMS**

En este enfoque, se parte de I señales de referencia $x_i(n)$ que circulan a través del sistema de control de ruido activo. Estas señales se propagan a través de un camino primario hacia el área donde se quiere suprimir el ruido, representado como K señales de ruido primario $d_k(n)$.

Paralelamente, las I señales $x_i(n)$ son filtradas por IxJ filtros de control, dando lugar a las J señales de salida de los filtros $y_j(n)$.

K señales de error $e_k(n)$ son el resultado de la superposición de las K señales de ruido primario $d_k(n)$ con las J señales de salida de los filtros que se propagan a través de los (JxK) caminos secundarios, denominadas como $y_{f_k}(n)$.

[10] Durante el proceso de adaptación, el algoritmo FxLMS ajusta los coeficientes de los filtros para minimizar la potencia del error instantáneo de las señales de error. Este ajuste se realiza mediante un enfoque de descenso de gradiente. El algoritmo se basa en el supuesto de que todas las respuestas impulsivas están modeladas por filtros FIR (Respuesta Finita al Impulso).

En sistemas de tiempo discreto, las señales de salida $y_j(n)$ pueden expresarse como una convolución entre los coeficientes del filtro $w_{ij}(n)$ y las señales de entrada $x_i(n)$, dadas por la ecuación:

$$y_j(n) = \sum_i w_{ij}(n) * x_i(n), \qquad \text{j} = 1,\dots,\text{J} \;\; \text{y} \;\; \text{i} = 1,\dots,\text{I} \tag{2}$$

Teniendo en cuenta que $w_{ij}(n)$ representa el filtro que alimenta al altavoz j-ésimo a partir de la señal de referencia i-ésima.

La ecuación para actualizar los pesos del filtro es fundamental en el algoritmo FxLMS, donde $\mu$ es el paso de convergencia que controla la velocidad de convergencia del algoritmo.

La actualización de los pesos tiene esta forma:

$$w_{ij}(n+1) = w_{ij}(n) + \mu \sum_k v_{ijk}(n) \cdot e_k(n), \qquad \text{k} = 1,\dots,\text{K} \tag{3}$$

Donde en el instante n-ésimo,

- $w_{ij}$(n) es el peso que une la señal i-ésima con el actuador j-ésimo
- $e_k(n)$ es el error medido en el micrófono de error k-ésimo
- $v_{ijk}(n)$ representa la muestra de la señal $x_i(n)$, filtrada por la estimación del camino secundario $\widehat{H}_{jk}$ :

$$v_{ijk}(n) = \widehat{H}_{jk} * x_i(n) \tag{4}$$

La k-ésima señal de error $e(n)$ según se define en la ecuación 5. Este error se utiliza para ajustar los coeficientes de los filtros de la ecuación 3, de forma que se minimice este error en el siguiente paso o iteración.

$$e_k(n) = d_k(n) + \sum_j H_{jk} * y_j(n) \tag{5}$$

Este proceso de adaptación se repite con cada muestra en el tiempo, ajustando continuamente los pesos de los filtros para alcanzar la minimización del ruido.

### *2.2.3 Algoritmo Modified Multichannel Filtered x-LMS*

La elección del algoritmo FxLMS modificado es fundamental en el contexto de este proyecto para acelerar la convergencia hacia una solución óptima. Al simular la adaptación como si no existieran los caminos secundarios, el FxLMS modificado simplifica el problema de control de ruido activo. Esta simplificación es lo que va a permitir integrar redes que no están intrínsecamente diseñadas para tratar con las peculiaridades introducidas por los caminos secundarios de manera directa. Así, el uso del FxLMS modificado nos permite aprovechar la eficiencia de las redes neuronales en un entorno controlado, centrando el aprendizaje en la minimización del error de forma más eficiente y efectiva.

La diferencia con el algoritmo FxLMS tradicional es la manera en la que se calculan las señales de error, tal y como se puede observar en la figura 4. Para calcular las señales de error en el algoritmo modificado, se hace una estima de la señal primaria de ruido $d'_k(n)$, mediante el uso de otra nueva estimación del camino secundario:

$$d'_k(n) = e_k(n) - \sum_j \widehat{H}_{jk} * y_j(n) \quad (6)$$

Y a partir de la ecuación 6 se calcula el nuevo error, a partir de la estimación de la señal primaria de ruido:

$$e''_k(n) = d'_k(n) + \sum_i \sum_j w_{ij}(n) \cdot v_{ijk}(n) \quad (7)$$

Con esta señal de error estimada, se llega al ajuste de pesos del algoritmo modificado:

$$w_{ij}(n+1) = w_{ij}(n) - \mu \sum_k v_{ijk}(n) \cdot e''_k(n) \quad (8)$$

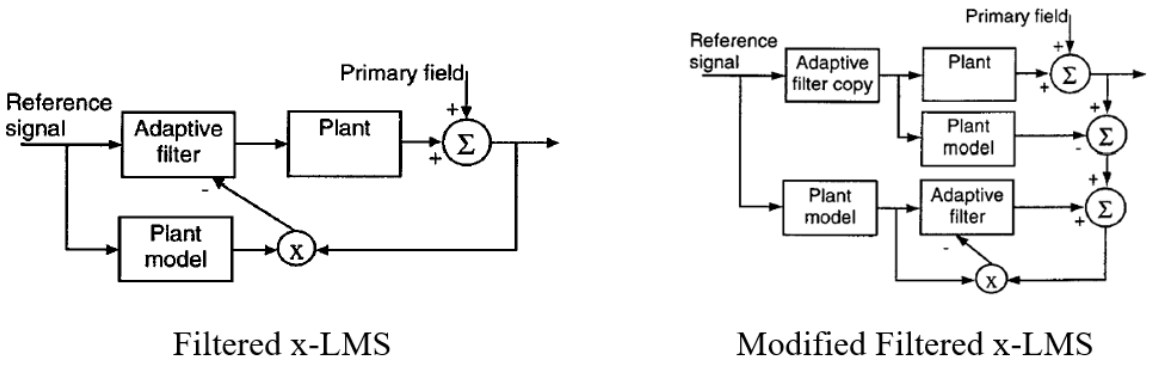


**Figura 4. Comparación esquemas FxLMS y Modified FxLMS [11]**

# Capítulo 3. Introducción a las redes neuronales

## 3.1 Historia

La historia de las redes neuronales se remonta a la década de 1950 con la creación del Perceptrón por Frank Rosenblat [11], concebido como una simple unidad de procesamiento que utilizaba entradas binarias y pesos asignados para producir una salida binaria basada en un umbral predeterminado. Este modelo se utilizaba principalmente para tareas de clasificación binaria simple y para implementar funciones lógicas como puertas AND y OR. Con el tiempo, surgieron avances más complejos, como el *Multilayer Perceptron* en 1965, que incluía capas de entrada, capas ocultas y capas de salida.

En los años 80, el surgen las Neuronas Sigmoides y las Redes *Feedforward*, a partir de este momento, las redes neuronales comenzaron a aprender de forma autónoma y podían ser utilizadas para llevar a cabo tareas más sofisticadas. El algoritmo de *Backpropagation*, desarrollado en 1986, surgió para entrenar redes neuronales de múltiples capas de forma supervisada, lo que expandió enormemente sus capacidades.

En 1989, las *Convolutional Neural Networks* (CNN) revolucionaron el procesamiento de imágenes al imitar el funcionamiento del cortex visual de los animales. Este avance permitió una extracción de características más eficiente y una clasificación más precisa en aplicaciones de reconocimiento de imágenes y procesamiento del lenguaje natural.

En 2006, las *Deep Belief Networks* (DBN) utilizaron una combinación de máquinas de Boltzmann y *autoencoders* para un preentrenamiento, este paso estableció las bases para los avances modernos en aprendizaje profundo.

Más tarde surgieron las *Generative Adversarial Networks* (GAN), [12] donde dos redes compiten entre sí para generar y discriminar datos, mejorando continuamente la calidad de la generación de datos sintéticos.

Desde 2014, ha habido avances significativos en las redes neuronales que han ampliado su aplicabilidad y eficacia en diversos campos [13]. Los desarrollos recientes en inteligencia artificial generativa, como GPT (*Generative Pre-trained Transformer*), han revolucionado áreas que van desde la composición de música y generación de texto hasta la creación de imágenes y producción de video. Herramientas como *Stable Diffusion*, *DALL-E*, y *Jukebox* han demostrado capacidades notables, apoyándose en modelos avanzados que incluyen *autoencoders* y redes neuronales adversarias generativas.

Además, las innovaciones en hardware neuromórfico han facilitado simulaciones más eficientes y económicas de redes neuronales [14], inspirándose en los mecanismos del cerebro biológico para mejorar el rendimiento computacional. Mejorando así la eficiencia energética y la capacidad de procesamiento de las redes neuronales artificiales, particularmente en aplicaciones como la conducción autónoma y la visión por computadora para drones.

## 3.2 Funcionamiento general de una red neuronal

En las Redes Neuronales Artificiales (ANN), los elementos procesadores funcionan de manera similar a las neuronas biológicas. Cada neurona recibe múltiples entradas que sumadas forman una señal combinada. Esta señal combinada es luego transformada por una función de transferencia, cuyo resultado se envía directamente a la salida de la neurona. Las salidas de estas neuronas pueden estar conectadas a las entradas de otras neuronas a través de conexiones ponderadas que simulan la eficacia sináptica en las neuronas biológicas.

Normalmente, las neuronas se organizan en grupos conocidos como capas.[15] Una red típica está formada por varias capas en cascada, con conexiones que solo se establecen entre capas contiguas.

Existen dos capas especiales que interactúan directamente con el entorno exterior, que se pueden ver en la Figura 5, la capa de entrada, donde se introducen los datos a la red, y la capa de salida que muestra la respuesta de la red ante los datos ingresados. Las capas que se encuentran entre medias de la capa de entrada y la capa de salida se conocen como capas ocultas.

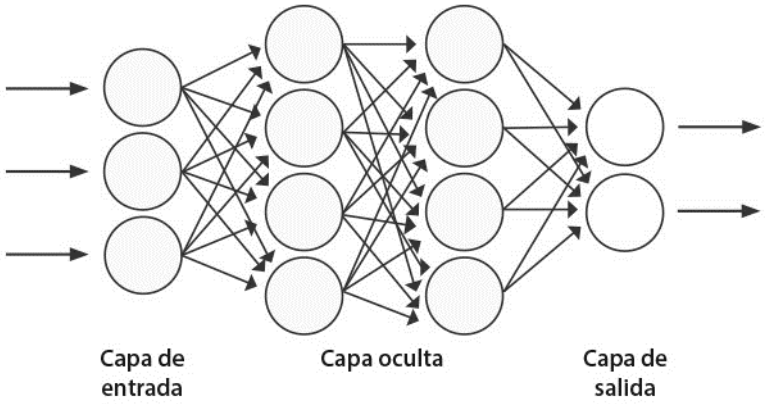


**Figura 5. Estructura común de una red neuronal**

- **Red neuronal monocapa**

Una red neuronal monocapa o perceptrón simple, consta de una única capa de neuronas de salida directamente conectadas a las entradas.

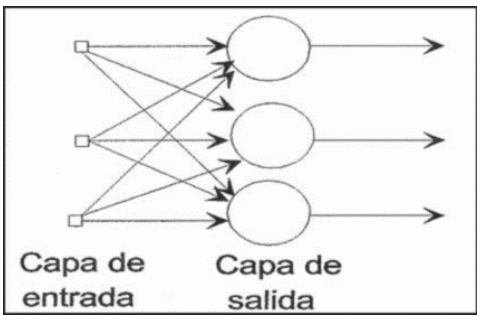


**Figura 6. Red neuronal monocapa**

- **Red neuronal multicapa**

Las redes neuronales multicapa, también conocidas como perceptrones multicapa, incorporan múltiples capas ocultas entre la entrada y la salida, que permiten modelar complejidades y abstracciones de los datos.

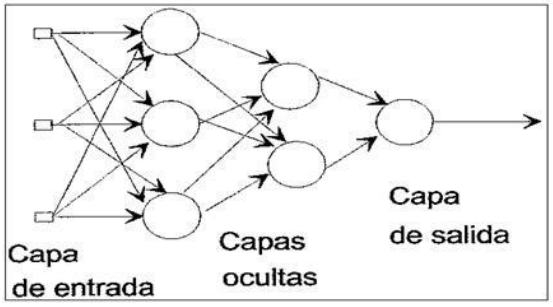

**Figura 7. Red neuronal multicapa**

- **Red neuronal recurrente**

Las redes neuronales recurrentes se caracterizan por tener conexiones que forman bucles, permitiendo el procesamiento de secuencias de datos y la persistencia de información a través del tiempo. Esto las hace especialmente aptas para tareas donde el contexto y el orden temporal son cruciales, como el procesamiento de lenguaje natural y las series temporales.

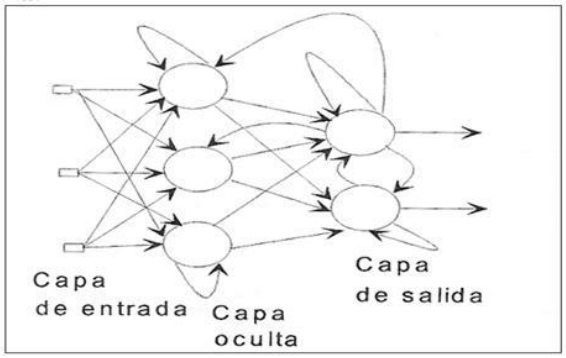


**Figura 8. Red neuronal recurrente**

## 3.3 Modelo de una neurona convencional

En el interior de una neurona artificial, representado en la figura 9, el proceso comienza con la recepción de señales procedentes de las entradas de la red o de las salidas de otras neuronas $x_j$. Cada entrada viene acompañada de un peso sináptico $w_{ij}$ , que indica la importancia o influencia de la señal de entrada correspondiente en la neurona.

Las señales de entrada multiplicadas por sus respectivos pesos se suman en un proceso que imita la integración de las señales en el cuerpo celular de una neurona biológica. En esta etapa también se incluye un término llamado umbral o sesgo $\theta_i$ , que se sustrae de la suma ponderada de las entradas. Este umbral determina la facilidad con la que la neurona puede activarse, dependiendo del resultado del sumatorio.

Finalmente, el resultado de esta suma se pasa a través de una función de activación $f(\cdot)$ , que determina la salida $y_i$ de la neurona. [16]

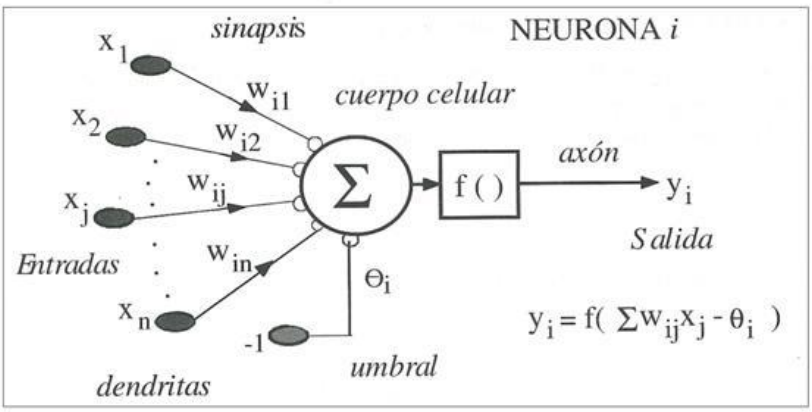


**Figura 9. Modelo de una neurona convencional**

En resumen, la operación de una neurona artificial en una red neuronal se puede describir por la ecuación,

$$y_i = f(\sum \omega_{ij} \cdot x_i - \theta_i)$$

**( 9 )**

que subraya el proceso de ponderación de las entradas $x_i$ con su peso sináptico correspondiente, su integración y posterior transformación a través de la función de activación $f(\cdot)$ para producir una salida $y_i$ , que será utilizada ya sea como una respuesta final de la red o como entrada a otra neurona en una capa siguiente.

## 3.4 Funciones de activación

Las funciones de activación transforman la entrada ponderada de un nodo en su salida correspondiente. [17] Sin estas funciones, una red neuronal se limitaría a comportarse como un modelo de regresión lineal, lo que restringiría su capacidad para aprender y abordar tareas más complejas.

Las funciones de activación se dividen en dos tipos, lineales y no lineales:

- **Función lineal:** Conocida también como función identidad, esta función de activación tiene la característica de que la salida replicará exactamente la entrada [18]. Así, si en una red neuronal multicapa se aplican exclusivamente funciones lineales de activación, el comportamiento resultante es análogo al de una regresión lineal. La función de activación lineal se utiliza típicamente cuando el resultado deseado es llevar a cabo una regresión lineal, facilitando así que la red neuronal produzca un valor específico y determinado, esta función se utilizará más adelante en la propuesta de cancelación de ruido de este trabajo.

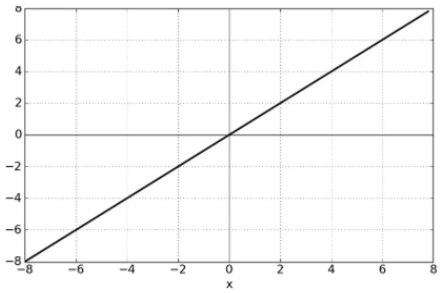


**Figura 10. Función lineal**

Entre las funciones no lineales más usadas se encuentran:

- **Función Sigmoide:** Transforma los valores de entrada para que estén entre 0 y 1, lo cual es óptimo para problemas de clasificación binaria. La función sigmoide se define como:

$$\theta(z) = \frac{1}{1 + e^{-z}}$$

**( 10 )**

Donde la variable z equivale a la suma ponderada de la entrada de la neurona:

$$z = b + \sum w_i \cdot x_i$$

**( 11 )**

Sin embargo, esta función ha perdido popularidad en ciertos campos debido a que puede provocar un estancamiento en la convergencia debido a la aparición de valores de gradiente próximos a cero durante el entrenamiento de redes neuronales profundas. Si se traza el valor de $\theta(z)$ como función de la entrada z, se obtiene la Figura 11.

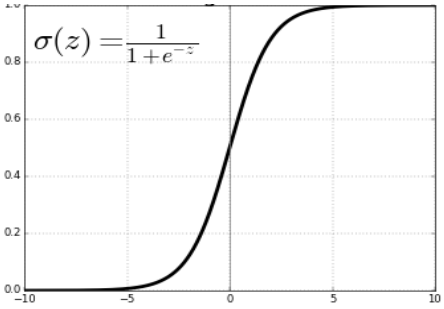


**Figura 11. Función sigmoid**

- **Función Tangente Hiperbólica:** Opera de manera similar a la sigmoide, pero su salida varía entre -1 y 1. También sufre del problema de estancamiento en la convergencia en contextos específicos.

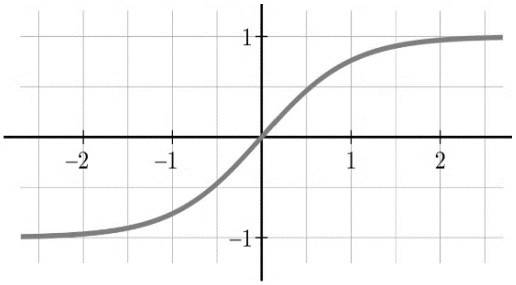


**Figura 12. Función tanh**

- **Función Unidad Lineal Rectificada:** También conocida con ReLU, devuelve el máximo entre cero y el valor de entrada, facilitando un entrenamiento más eficiente y evitando ciertos problemas en redes profundas. Una variante, la ReLU con fugas, ayuda a mitigar el problema conocido como "muerte de neuronas".

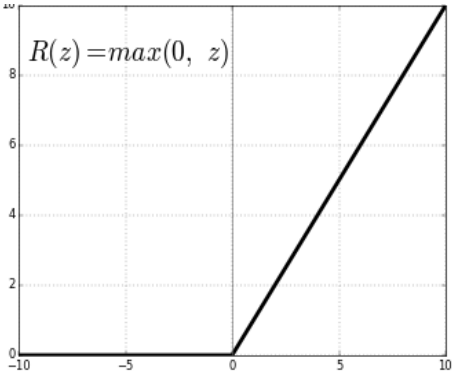


**Figura 13. Función ReLU**

- **Función Softmax:** Comúnmente usada en la capa de salida para clasificación multiclase, convierte las salidas en un conjunto de probabilidades que suman uno, simplificando la interpretación de las predicciones. [19]

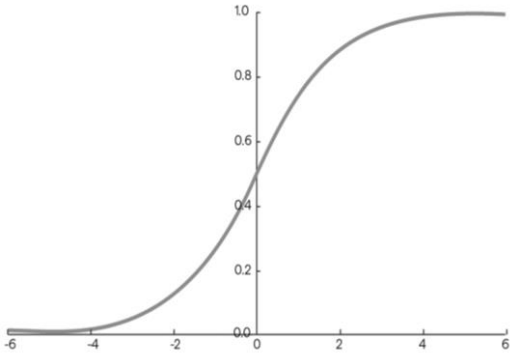


**Figura 14. Función Softmax**

## 3.5 Función de salida

La función de salida define el valor que se transfiere de una neurona a su capa de salida, ya sea bien conectada a siguientes neuronas, o sea esta misma la salida de la red [20]. Esta función determina si el valor generado por la función de activación alcanza un cierto umbral para ser transmitido a la siguiente neurona.

Generalmente, no cualquier valor puede servir como entrada para una neurona, por lo que los valores de salida suelen estar limitados a rangos como [0, 1] o [-1, 1], o pueden ser binarios, es decir, {0, 1} o {-1, 1}.

Las dos funciones de salida más utilizadas son:

- **Identidad:** Es la forma más simple de función de salida, donde el valor de salida es exactamente igual al valor de entrada. Es útil en casos donde se desea mantener la linealidad entre la entrada y la salida.
- **Binaria:** Asigna un 1 si la activación es mayor o igual que el umbral $\theta_i$ , y un 0 en caso contrario. Esta disposición asegura que solo los valores que superan un cierto límite son transmitidos, simplificando la salida a formas más manejables para ciertos tipos de procesamiento.

## 3.6 Aprendizaje de la red

El *learning rate* o paso de aprendizaje de una red neuronal ajusta los pesos y sesgos de las conexiones entre neuronas para minimizar el error entre las predicciones de la red y los valores reales. Se realiza a través de un algoritmo de optimización, como el descenso de gradiente, que modifica los parámetros de la red en cada ciclo de entrenamiento. El tamaño del paso de aprendizaje es crucial, ya que influye directamente en la rapidez con la que la red converge hacia una solución óptima y en la calidad del resultado final.

La función de aprendizaje, o función de pérdidas, es la que cuantifica la diferencia entre las salidas calculadas por la red y los valores reales. La selección de esta función depende del tipo de tarea que la red está realizando [21].

Entre las más usadas encontramos:

- **Error Cuadrático Medio**: Muy utilizada en problemas de regresión. Calcula el promedio de los cuadrados de las diferencias entre las predicciones de la red y los valores reales. La fórmula para el MSE es:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 , \tag{12}$$

  donde $(n)$ es el número total de muestras, $(y_i)$ es el valor real de la muestra $(i)$, y $(\hat{y}_i)$ es la predicción de la red para la misma muestra.

- **Entropía Cruzada:** La entropía cruzada es una función de aprendizaje comúnmente utilizada en problemas de clasificación binaria y multiclase. Mide la diferencia entre la distribución de probabilidad predicha por la red y la distribución de probabilidad real. La fórmula para la entropía cruzada es:

$$-\frac{1}{n}\sum_{i=1}^{n} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i) \tag{13}$$

  donde $(y_i)$ es el valor real de la clase para la muestra $(i)$, y $(\hat{y}_i)$ es la probabilidad predicha por la red de que la muestra pertenezca a la clase $(i)$.

- **Entropía Cruzada Binaria:** Similar a la entropía cruzada, pero diseñada específicamente para problemas de clasificación binaria. Esta función mide la diferencia entre la distribución de probabilidad predicha por la red y la distribución de probabilidad real en un problema de dos clases. [22]

### *3.6.1 Algoritmo de descenso de gradiente*

En la búsqueda de la configuración óptima de los parámetros de un modelo que minimice la función de pérdidas, un enfoque teórico sería evaluar la pérdida para cada combinación posible de parámetros y seleccionar aquellos que produzcan el valor mínimo. Sin embargo, no es posible llevar este método a la práctica, ya que tiene un enorme costo computacional y requiere un largo tiempo de procesamiento, especialmente en modelos con una gran cantidad de parámetros como son las redes neuronales [23].

Para superar esta limitación, se utiliza el algoritmo de descenso de gradientes, que es más eficiente y práctico. En este algoritmo, se selecciona un punto inicial arbitrario en el espacio de parámetros y se calcula el gradiente de la función de pérdida en este punto. El gradiente indica la dirección en la que la función de pérdida aumenta más rápidamente.

Para avanzar hacia el mínimo de la función de pérdida, el próximo paso se determina moviéndose en la dirección opuesta al gradiente. Esto se logra multiplicando el gradiente por el paso de aprendizaje, que generalmente varía entre 0 y 1. Un paso de aprendizaje típico podría ser 0.05, aunque el valor óptimo puede variar significativamente dependiendo del problema específico y la configuración del modelo.

Este método iterativo de ajuste de parámetros permite que el modelo se adapte gradualmente a los datos, optimizando la función de pérdida de manera más eficiente que los métodos exhaustivos, lo que facilita la construcción de modelos predictivos precisos y robustos en un tiempo computacionalmente razonable.

## 3.7 Técnica de Back-propagation

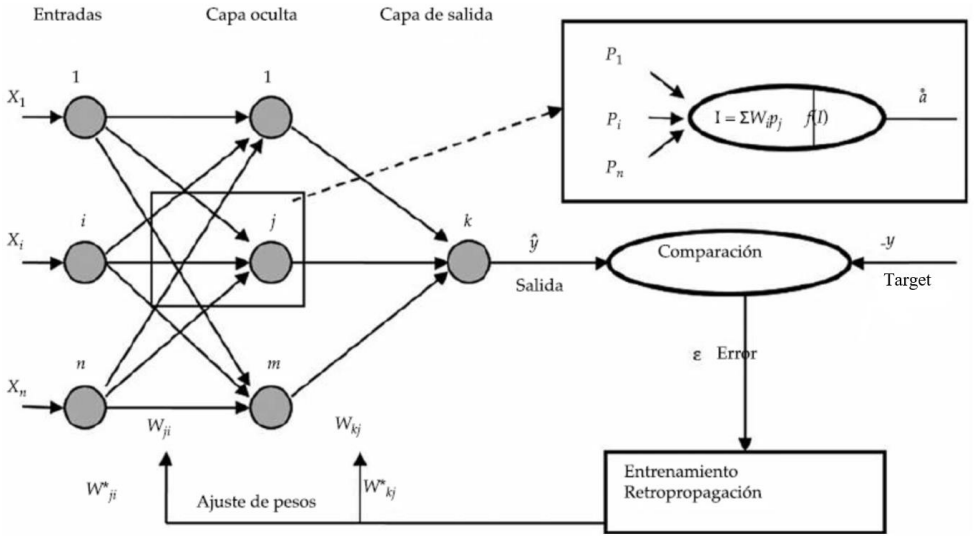


**Figura 15. Proceso de back-propagation**

La técnica de retropropagación, o *back-propagation*, es un método supervisado de entrenamiento de redes neuronales, que permite ajustar los pesos de la red para minimizar la discrepancia entre las respuestas deseadas (*targets*) y las obtenidas, como se puede observar en la figura 15.

El proceso de aprendizaje de retropropagación se desarrolla en dos fases principales: propagación hacia adelante y propagación hacia atrás. [24]

### *3.7.1 Propagación hacia adelante:*

En esta etapa, cuando se presenta una señal de entrada, se activa la capa de entrada de la red y se calcula la salida de la primera capa oculta. Este proceso se repite sucesivamente a través de las capas, donde la salida de cada neurona se calcula como una suma ponderada de sus entradas, modificada por una función de activación. Estos valores de salida se transmiten a lo largo de las conexiones de salida hacia la siguientes capas, hasta llegar a la última.

### *3.7.2 Propagación hacia atrás:*

Una vez completada la propagación hacia adelante y obtenida la respuesta de la red, comienza la fase de corrección o propagación hacia atrás. En esta fase se ajustan los pesos, iniciando por la capa de salida y retrocediendo hacia las capas anteriores. El ajuste se basa en el cálculo del error, que es la diferencia entre la salida deseada (target) y la salida obtenida.

Para las neuronas de la capa de salida, este ajuste es más directo, dado que se conoce la salida deseada. Sin embargo, para las capas ocultas, donde no hay un valor deseado claro, el error se propaga hacia atrás desde la capa de salida, ajustando los pesos en función de su contribución al error total, esto se lleva a cabo mediante la regla de la cadena.

Es decir, durante la propagación hacia atrás, el error calculado en la salida se utiliza para ajustar los pesos. Para cada neurona, el impacto de sus pesos en el error final es determinado por derivadas parciales, que indican cómo cambia el error con pequeñas modificaciones en los pesos. La regla de la cadena se aplica aquí para descomponer estas derivadas en términos más simples, que se pueden calcular paso a paso desde la salida hacia la entrada de la red.

Se puede observar un ejemplo de este procedimiento en la Figura 16.

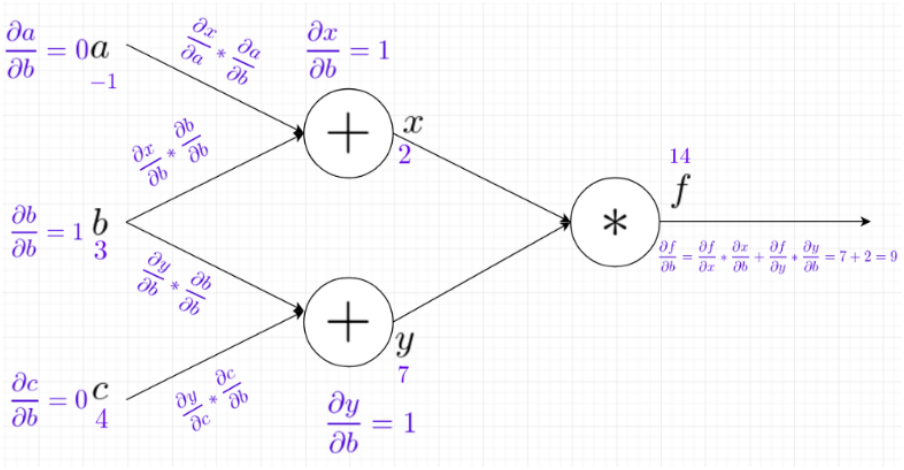


**Figura 16. Regla de la cadena**

Este proceso iterativo de ajuste de pesos mediante la retropropagación del error permite que la red aprenda a mapear de manera eficaz las entradas con las salidas deseadas. Este ciclo se repite hasta alcanzar un rendimiento satisfactorio de la red o hasta que los cambios en el error sean mínimos, facilitando así la creación de modelos predictivos robustos y precisos.

# Capítulo 4. Resolución del problema de identificación de canal

En los próximos capítulos, se va a abordar la solución propuesta a la cancelación de ruido activa mediante el uso de redes neuronales. Como base de esta solución el primer problema a considerar es la estimación de un canal, y sus posibles soluciones.

La identificación de un canal consiste en estimar las características o la respuesta de un sistema de transmisión, que puede ser un medio físico por el que se propaga una señal, [25] como cables en telecomunicaciones, el aire en sistemas acústicos, o cualquier otro medio conductor. Este proceso permite entender cómo una señal se ve afectada por su trayecto desde el emisor hasta el receptor, identificando efectos como la distorsión, el retardo, la atenuación y otros tipos de interferencias que puedan alterar la señal original. Al comprender y modelar correctamente estas características del canal, es posible diseñar sistemas que compensen activamente estas alteraciones, mejorando así la integridad y la calidad de las señales transmitidas. La identificación de un canal puede emplear diversas técnicas dependiendo del tipo de señal y del sistema específico. A continuación se verán las técnicas de identificación de canal más relevantes para este proyecto.

## 4.1 Identificación de canal mediante LMS

La identificación del canal utilizando el algoritmo adaptativo LMS es una técnica básica en el campo del procesamiento de señales y las comunicaciones. El algoritmo LMS es un método adaptativo simple y eficaz para estimar los coeficientes de un filtro que representan el canal por el cual una señal ha sido propagada.

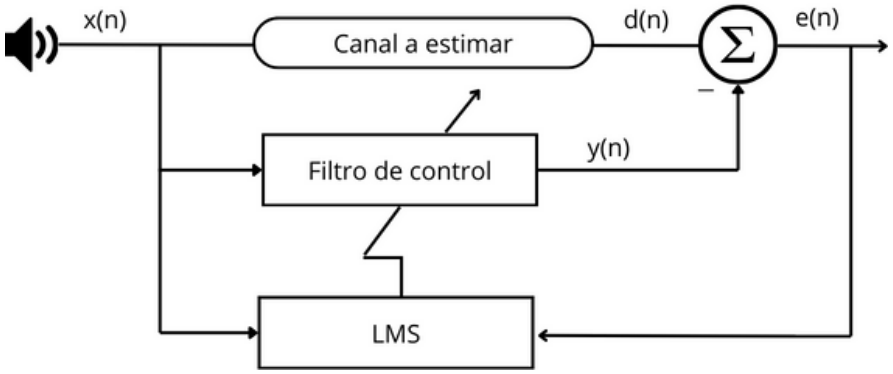


**Figura 17. Identificación de canal mediante LMS**

El proceso de identificación de canal mediante el algoritmo LMS representado en la figura 17, comienza con una inicialización aleatoria o nula de los coeficientes del filtro que representan el canal a estimar. Este algoritmo ajusta iterativamente los coeficientes del filtro de control basándose en el error entre la señal primaria de ruido (salida original del canal a estimar, $d(n)$) y la señal procesada por los coeficientes estimados actualmente (salida del filtro, $y(n)$). En cada paso, los coeficientes del filtro se actualizan para minimizar el error cuadrático instantaneo. Este proceso de ajuste se repite hasta que los cambios en los coeficientes sean mínimamente pequeños, indicando que el filtro ha convergido a una buena representación del canal.

## 4.2 Identificación de canal mediante red neuronal

La identificación de canal puede abordarse mediante una red neuronal de manera similar a la identificación mediante el algoritmo LMS, ya que ambos métodos se centran en optimizar los pesos o coeficientes para minimizar la diferencia entre la señal deseada (salida del canal) y la señal de salida producida por el modelo, bien sea el algoritmo LMS o la red neuronal.

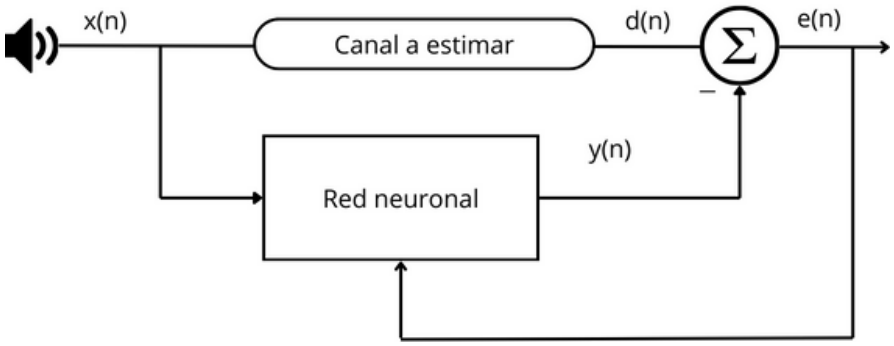


**Figura 18. Identificación de canal mediante red neuronal**

En el caso del uso de una red neuronal para realizar una estimación del canal como la representada en la figura 18, si esta red está configurada con una función de activación lineal y entrenada con un algoritmo de descenso de gradiente, el proceso es notablemente similar al descrito anteriormente. Las señales en juego son las mismas que en el caso anterior, la red también recibe una señal de entrada (señal de referencia $x(n)$) y ajusta sus pesos sinápticos (equivalentes a los coeficientes del filtro en LMS) para minimizar la diferencia entre la señal primaria de ruido $d(n)$ y la señal que sale de la misma red $y(n)$. Por lo que la red puede ser entrenada con el mismo objetivo que el algoritmo LMS: minimizar el error entre la salida calculada y la salida deseada.

En el caso de identificación de canal mediante el algoritmo LMS, los coeficientes del filtro se actualizan utilizando la fórmula de la ecuación 1. Esta actualización de coeficientes intenta minimizar el error cuadrático medio entre la señal de ruido primaria y la salida del sistema.

Si este filtro adaptativo, es sustituido por una red neuronal con activación lineal, los pesos de la red se actualizan de manera similar, a través del gradiente del error, que también se basa en la minimización del error cuadrático medio. La regla de actualización puede escribirse como:

$$w = w - \mu \cdot \nabla E \tag{14}$$

donde $\nabla E$ es el gradiente del error cuadrático, calculado como el producto de la señal de error y la entrada.

Tanto en el algoritmo LMS como en la red neuronal con activación lineal, los pesos se ajustan en función del error producido en la salida. La diferencia principal radica en la formulación del gradiente: mientras que en el algoritmo LMS el ajuste es directamente proporcional al error y la entrada; en la red neuronal, el ajuste también considera la derivada de la función de activación. Sin embargo, cuando esta función es lineal, la derivada es 1, por lo tanto, el ajuste se simplifica a ser directamente proporcional al error y la entrada, al igual que en el algoritmo LMS.

En práctica, esto significa que en el caso de estimación de un canal, una red neuronal que se ajusta (entrena) muestra a muestra con una función de activación puramente lineal se comporta de manera equivalente a un filtro adaptativo LMS en términos de cómo se ajustan los pesos. Ambos sistemas adaptan sus coeficientes para minimizar de forma iterativa el error cuadrático instantáneo, aprovechando la correlación entre la entrada y el error para dirigir la adaptación.

# Capítulo 5. Resolución del problema de control activo de ruido

Si nos centramos en el problema de ANC, resulta fundamenta manipular adecuadamente las señales de entrada y adaptar los coeficientes de los filtros con un modelo de procesado adecuado. Al contrario que en el caso de la simple identificación de un canal donde se caracteriza un medio estático a través del cual se propaga una señal, el ANC enfrenta el desafío adicional de lidiar con señales de ruido que varían en el tiempo y en su naturaleza. Esto exige no solo identificar el canal por el que se transmite el ruido sino también generar una señal opuesta que efectivamente neutralice este ruido en tiempo real. Este proceso requiere no solo un ajuste inicial basado en un modelo estadístico o histórico de la señal de ruido, sino una adaptación constante a las variaciones instantáneas del ruido, lo que implica una manipulación compleja y continua de las entradas y ajustes de los modelos en uso.

Esto quiere decir, que si se quiere solucionar el problema de ANC, no se puede usar la identificación del canal descrita anteriormente, sin una manipulación previa de las entradas tanto de la red neuronal, como del algoritmo LMS.

El esquema de la Figura 19, muestra el diagrama de bloques de la solución propuesta para ANC. En el caso de cancelación de ruido activa mediante algoritmos tradicionales**,** el bloque de procesado de señal implicaría la manipulación de las señales de entrada indicadas, así como el ajuste de los coeficientes del filtro adaptativo, o el ajuste de los pesos de la red neuronal, dependiendo de que solución se escoja.

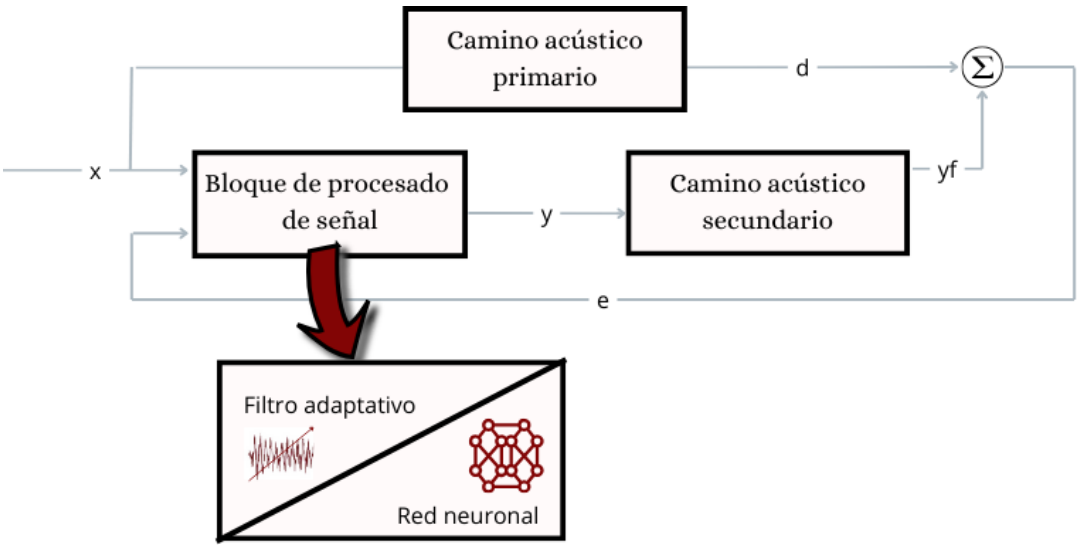


**Figura 19. Sistema adaptativo para control activo de ruido**

El sistema que se propone para la ANC incorpora dos caminos acústicos como se puede observar en la figura 19, el desafío principal radica en gestionar las señales indeseadas introducidas por ambos de ellos. El camino primario es el canal por el cual la señal de referencia se propaga y se convierte en la señal de ruido primaria, por otro lado, existe un segundo camino (camino secundario) que se encuentra a la salida del bloque de procesado de señal.

Como se ha descrito en el capítulo 1, el ANC se basa en el principio de superposición por lo que el objetivo de este sistema, será generar una señal ($y$) mediante el modelo adaptativo, que después de ser propagada por el camino secundario mencionado se convierta en la señal ($y_f$) y sea opuesta en fase a la señal de ruido primaria ($d$) que sale del otro camino acústico, con el objetivo de anularla.

Los pasos seguidos para desarrollar la solución propuesta mediante una red neuronal incluyen: la creación de un algoritmo LMS para identificación de dos caminos acústicos desconocidos, su

modificación para solucionar un problema de ANC y finalmente el cambio del filtro del algoritmo modificado por una red neuronal personalizada. El siguiente paso ha sido la extensión de todos estos algoritmos para conseguir solucionar un problema multicanal.

## 5.1 ANC mediante algoritmo FxLMS modificado

En la figura 20 se pueden ver los cambios realizados sobre la identificación de canal tradicional mediante el algoritmo LMS, para poder solucionar un problema de ANC, esto se ha resuelto mediante el FxLMS modificado.

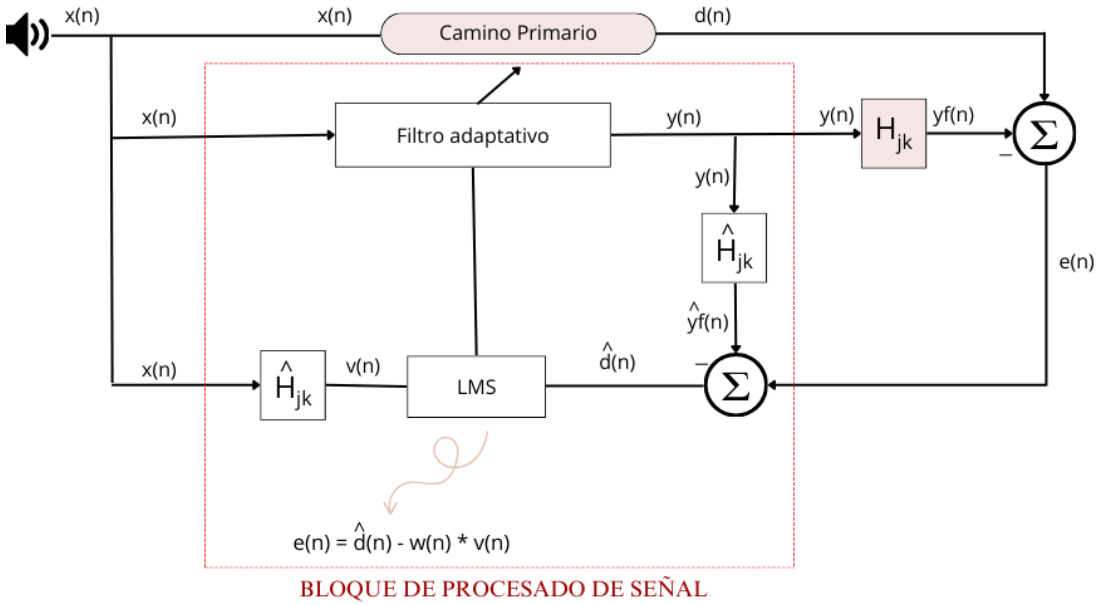


**Figura 20. Identificación de canales mediante algoritmo adaptativo**

En el sistema que se considera, existen el camino primario que propaga las señales de ruido hasta los micrófonos de error y el canal que introduce ruido a la señal canceladora de la salida del filtro (camino secundario, representado mediante $H_{jk}$). Si se considera un caso monocanal, tal y como se ha visto en los anteriores capítulos, el ajuste de pesos en el algoritmo FxLMS modificado propuesto, sigue la siguiente ecuación,

$$w(n+1) = w(n) + \mu \cdot v(n) \cdot e''(n) \quad \textbf{(15)}$$

haciendo uso de la señal de error:

$$e''(n) = \hat{d}(n) - w(n) \cdot v(n), \quad \textbf{(16)}$$

donde,

- $w(n)$ hace referencia al vector de coeficientes del filtro adaptativo
- $v(n)$ hace referencia a la señal $x(n)$ filtrada por la estima del camino secundario $\hat{H}_{jk}$
- $\mu$ hace referencia al paso de convergencia
- Siendo $\hat{d}(n)$ una estimación de la señal primaria de ruido y $e(n)$ la señal de error recogida por el micrófono:

$$\hat{d}(n) = e(n) - \hat{H}_{jk} * y(n) \quad \textbf{(17)}$$

Estas ecuaciones se pueden ver reflejadas en la estructura modificada de la figura 20.

El motivo principal de modificar las entradas que se usan normalmente en una estimación de canal mediante el algoritmo LMS, se debe a la necesidad de proporcionar en el caso de la sustitución del filtro adaptativo por una red neuronal, un objetivo o target para el entrenamiento de la red, esto se explica más en detalle en el siguiente apartado.

## 5.2 ANC mediante red neuronal

En este apartado se propone resolver mediante una red neuronal, el ajuste de los pesos de la red de manera análoga al ajuste de los coeficientes del filtro de la ecuación 15.

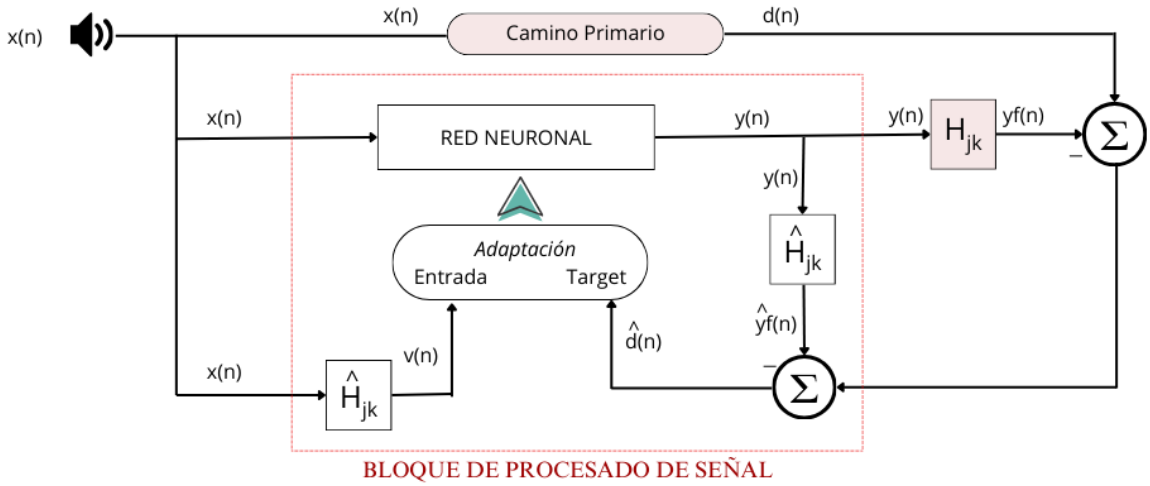


**Figura 21. Identificación de canales mediante red neuronal**

Esto se lleva a cabo mediante la adaptación de la red, que como se puede observar en la figura 21, requiere de una señal de entrada a la red que actúe como predictor y un target al que adaptar esta entrada. La estima de la señal primaria de ruido $\hat{d}(n)$ actúa como el objetivo (target) que se quiere obtener a la salida de la red, y la señal de referencia $x(n)$ filtrada por la estimación del camino secundario actúa como el predictor.

En el desarrollo de la solución de ANC mediante redes neuronales, se ha empleado MATLAB como herramienta de simulación para validar y optimizar el enfoque propuesto en diferentes configuraciones acústicas. Inicialmente, se abordó el problema en un entorno monocanal, lo que permitió concentrarse en la capacidad de la red neuronal para aprender y adaptarse a una única señal de referencia y generar una única señal que es recogida por un único micrófono de error. Este escenario simplificado ayudó a establecer una base sólida para los experimentos y a ajustar los parámetros críticos de la red, como la tasa de aprendizaje y la estructura de las capas.

El siguiente ha sido la expansión de esta estructura para abordar un escenario multicanal más complejo, ya que los entornos reales de ruido a menudo involucran múltiples fuentes y caminos de propagación del sonido, haciendo que el control de ruido sea más desafiante. La simulación multicanal permitió probar la eficacia de la red neuronal para manejar y cancelar ruido en varios puntos simultáneamente, como se ha propuesto en el objetivo de este proyecto.

### *5.2.1 Caso monocanal*

En el entorno de MATLAB se ha creado una red haciendo uso de la función *network*, compuesta por una sola capa oculta que contiene una única neurona. Esta configuración sugiere un enfoque centrado en capturar linealidades simples dentro de los datos, ideal para tareas de filtrado donde las relaciones entre las entradas y las salidas pueden ser directas y no excesivamente complejas.

La estructura de la red propuesta tiene una entrada y una salida como se puede ver reflejado en la figura 22, con la capa oculta conectada directamente a ambas. Esto significa que cada entrada se procesa a través de la neurona en la capa oculta, y el resultado es emitido directamente como la salida de la red.

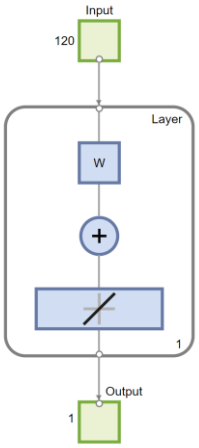


**Figura 22. Diagrama de red monocanal**

La red diseñada no tiene conexiones de sesgo, para para replicar el comportamiento de la ecuación de actualización de los pesos en el algoritmo adaptativo. Se ha utilizado una función de aprendizaje de descenso de gradiente '*learngd'*.

Además, se adaptan los pesos y sesgos usando la función de adaptación '*adaptwb'*, que es crucial para ajustar dinámicamente la red durante el entrenamiento en respuesta a los errores observados. Como función de rendimiento se utiliza el error cuadrático medio '*mse*', donde el objetivo es minimizar la diferencia cuadrada entre las salidas predichas y las reales.

Como se ha explicado anteriormente, una red neuronal con una función de activación puramente lineal se comporta de manera equivalente a un filtro adaptativo LMS modificado en términos de cómo se ajustan los pesos. Ambos sistemas adaptan sus coeficientes para minimizar de forma iterativa el error cuadrático instantáneo, aprovechando la correlación entre la entrada y el error para dirigir la adaptación.

Para comparar el comportamiento del algoritmo FxLMS modificado propuesto con el de la red neuronal diseñada, se han llevado a cabo sus respectivas simulaciones en MATLAB.

La eficiencia de estos algoritmos ha sido probada bajo las mismas condiciones, es decir, se ha realizado una simulación de ambos algoritmos, siendo estos alimentados con la misma señal de referencia, que ha sido propagada por el mismo camino primario, dando lugar a la misma señal primaria de ruido. El número de coeficientes del filtro equivale al número de pesos de la red neuronal, y el camino secundario por el que ha sido propagada tanto la salida del filtro como la salida de la red, es el mismo.

Los parámetro escogidos para esta simulación han sido:

| | |
|---|---|
| N | 3000 muestras |
| L | 120 coeficientes |
| $\mu$ | 0.5 |

**Tabla 1. Parámetros simulación monocanal**

Los resultados de la simulación de la del algoritmo FxLMS modificado se presentan en la figura 23, donde se observa la señal primaria de ruido (en azul) y la señal de error (en rojo), mostrando como la señal de error se reduce progresivamente, debido al ajuste de pesos del filtro adaptativo.

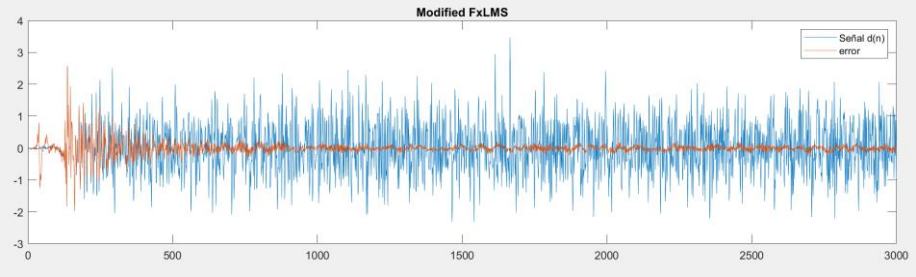


**Figura 23. Simulación algoritmo FxLMS Modificado monocanal**

Por otra parte, la figura 24 ilustra las mismas señales pero reemplazando el filtro adaptativo por la red neuronal diseñada.

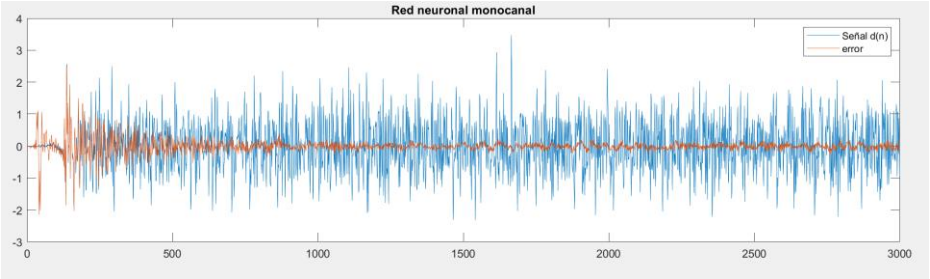


**Figura 24. Simulación red neuronal monocanal**

Como se puede observar, ambos modelos de ANC tienen el mismo resultado a la hora de solucionar el problema de ANC en un mismo sistema, aunque se puede observar que hay leves diferencias en el transitorio, debido a la diferencia en la inicialización de cada estrategia.

### *5.2.2 Caso multicanal*

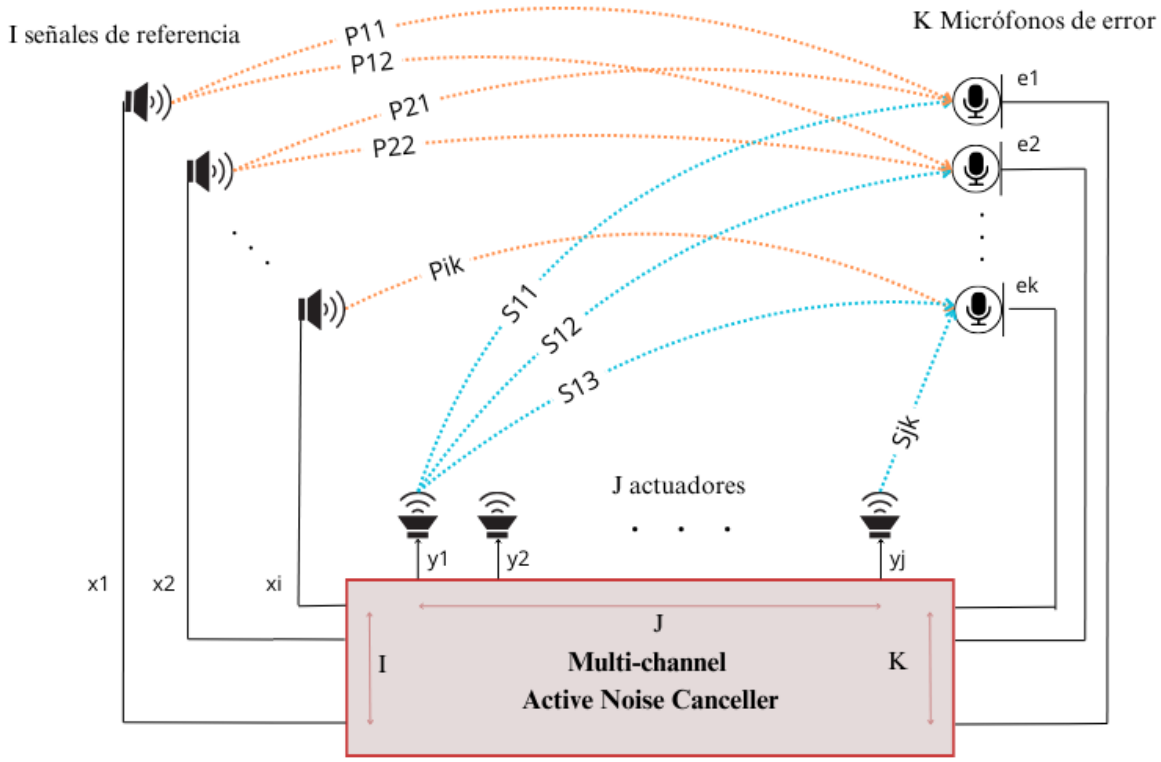


**Figura 25. Diagrama solución ANC multicanal**

Tras abordar con éxito el caso de ANC monocanal, se ha desarrollado un enfoque para sistemas multicanal como se ilustra en la Figura 25, el sistema propuesto es el siguiente:

Se tienen $I$ señales de referencia, que son propagadas por sus respectivos caminos primarios ($P_{ik}$) hasta los $K$ micrófonos de error, donde se recogen las K señales de ruido primarias. En el cancelador de ruido activo, se utiliza una red neuronal con $I \times J$ conexiones de pesos sinápticos, cada uno de tamaño $L$ (número de coeficientes a estimar de cada filtro adaptativo equivalente) para procesar y transformar las I señales de referencia en J señales de cancelación de ruido. Estas señales de cancelación de ruido serán entonces propagadas a través de sus respectivos caminos secundarios ($S_{jk}$) hacia los $K$ micrófonos de error. En los K micrófonos de error, las señales de cancelación de ruido generadas por el sistema ANC se emplean para contrarrestar o mitigar las señales de ruido primarias que llegan desde los caminos primarios, buscando así neutralizar el efecto acústico indeseado introducido mediante el principio de superposición.

La estructura de la red dentro del sistema cancelador de ruido que se presenta en la figura 26 está diseñada con J entradas, cada una de dimensiones IxL. En esta configuración, cada entrada está directamente conectada a una única capa que simula una salida, resultando así en un total de J salidas, que serían en efecto las salidas de los J actuadores. Las conexiones están organizadas de manera que cada señal i-ésima en cada entrada tiene una única conexión con la neurona de su respectiva capa oculta j-ésima, esta conexión está definida por un peso específico $w_{ij}$, lo que garantiza que cada neurona procese exclusivamente la información de su entrada asignada para generar la salida deseada.

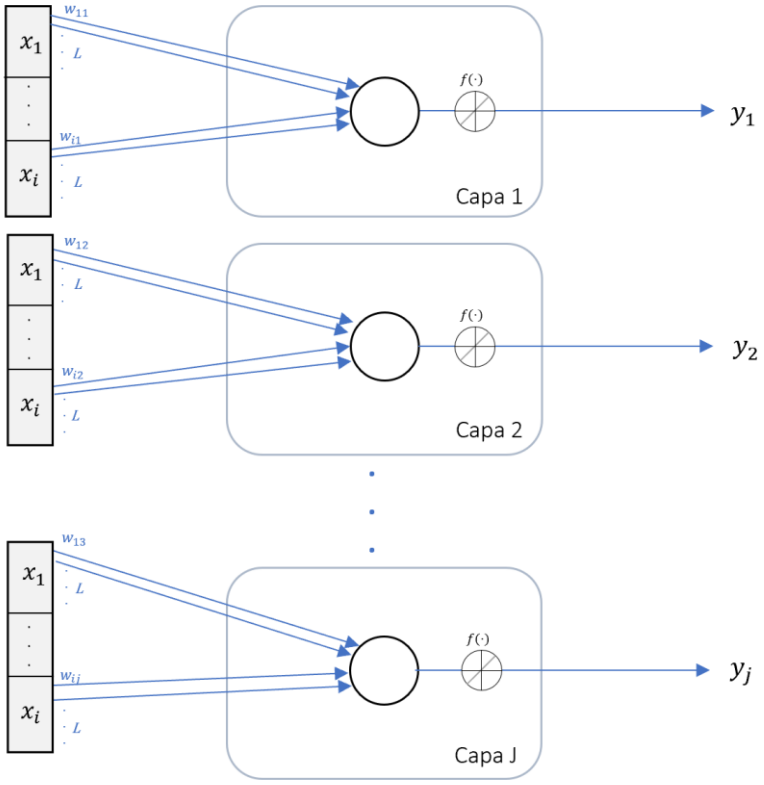


**Figura 26. Red neuronal personalizada**

La mayor dificultad encontrada al plantear la solución multicanal en el entorno de MATLAB ha sido la organización de los datos, estos se detallan en la Tabla 1.

| | |
|---|---|
| I | Número de señales de referencia |
| J | Número de actuadores |
| K | Número de micrófonos de error |
| L | Número de coeficientes del canal a estimar |
| $\mu$ | Paso adaptativo |
| $\hat{h}_{jk}$ | Estimación del camino secundario que vincula el j-ésimo actuador con el k-ésimo micrófono de error |
| $x_i(n)$ | Valor en el instante n de la i-ésima señal de referencia |
| $d_k(n)$ | Valor en el instante n de la señal de ruido primario recogida en el k-ésimo micrófono de error |
| $e_k(n)$ | Valor en el instante n de la señal de error recogida en el k-ésimo micrófono de error |
| $y_j(n)$ | Valor en el instante n de la señal que genera el actuador j-ésimo |
| $v_{l\_ijk}(n)$ | Valor en el instante n del buffer con los últimos L valores de la señal de referencia $x_i$ después de ser propagada por la estimación del camino secundario $\widehat{\mathrm{H}}_{jk}$ |
| $w_{ij}(n)$ | Matriz de pesos sinápticos de la red que unen la i-ésima señal de referencia con el j-ésimo actuador |

**Tabla 2. Notación algoritmo de ANC mediante red neuronal**

A continuación, se muestran los pasos que se han llevado a cabo para la ANC, mediante la red neuronal presentada en la figura 26:

* Para fines descriptivos, las ecuaciones y estructuras usadas en esta descripción simulan el caso de un cancelador activo de ruido de 2x3x4, que se refiere a 2 señales de referencia, 3 actuadores y 4 micrófonos de error *

El primer paso consiste en estimar cómo el camino secundario influirá en las señales que salen de la red. Utilizando las J señales de la salida de la red del procesamiento previo mostradas en la ecuación 19, estas señales se propagan a través de la estimación del camino secundario, resultando en una matriz que representa la convolución de cada señal $y_j$ con la estima del camino secundario $\hat{h}_{jk}$ correspondiente, representada en la ecuación 20.

$$\mathrm{y} = [\mathrm{y}_1 \ \mathrm{y}_2 \ \mathrm{y}_3 \,] \tag{18}$$

$$\hat{\mathrm{y}}_\mathrm{f} = \begin{bmatrix} \mathrm{y}_1 * \hat{\mathrm{h}}_{11} & \mathrm{y}_1 * \hat{\mathrm{h}}_{12} & \mathrm{y}_1 * \hat{\mathrm{h}}_{13} & \mathrm{y}_1 * \hat{\mathrm{h}}_{14} \\ \mathrm{y}_2 * \hat{\mathrm{h}}_{21} & \mathrm{y}_2 * \hat{\mathrm{h}}_{22} & \mathrm{y}_2 * \hat{\mathrm{h}}_{23} & \mathrm{y}_2 * \hat{\mathrm{h}}_{24} \\ \mathrm{y}_3 * \hat{\mathrm{h}}_{31} & \mathrm{y}_3 * \hat{\mathrm{h}}_{32} & \mathrm{y}_3 * \hat{\mathrm{h}}_{33} & \mathrm{y}_3 * \hat{\mathrm{h}}_{34} \end{bmatrix} \tag{19}$$

Así cada elemento de esta matriz representa la estimación de la salida j-ésima del filtro al viajar por el camino secundario que la lleva al k-ésimo micrófono de error.

El siguiente paso sería calcular los targets que se van a usar en la adaptación de la red. La red se va a adaptar K veces para cada muestra de señal, ya que en cada iteración se va a buscar minimizar el error en el k-ésimo micrófono.

Al tener la red J salidas, cada k-ésima adaptación tiene J targets. Estos targets se definen como la diferencia entre la señal de error recogida en el micrófono k-ésimo y la salida de la red filtrada por la estimación del camino secundario correspondiente, que sería el elemento correspondiente a la posición $j, k$ de la matriz $\hat{\mathrm{y}}_\mathrm{f}$.

Si se recuerda la figura 21, el otro elemento necesario para realizar la adaptación de la red serían las J entradas, que se nombran como predictores.
En cada k-ésima adaptación, cada j-ésimo predictor se obtiene mediante la propagación de la i-ésima señal de referencia a través de la estimación del camino secundario correspondiente, esta operación se refleja en la ecuación 25.

$$v_{ijk}(n) = \hat{h}_{jk} * x_i \tag{20}$$

Hay que tener en cuenta que en cada j-ésimo predictor solo se usan las últimas L entradas anteriores, para ello se usa el buffer de la figura 27.

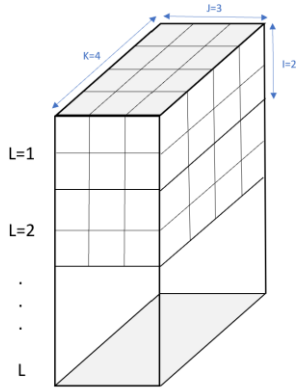


**Figura 27. Buffer de L muestras para predictores**

El learning rate se ajusta en cada adaptación en función al paso adaptativo $\mu$ y a los J predictores. Para el ajuste de este, se considera tanto la energía de los predictores como la correlación entre ellos. Como se ha explicado en capítulos anteriores, el learning rate, es el parámetro que determina la velocidad con la que los pesos de la red se ajustan durante el entrenamiento. Un learning rate inapropiado puede llevar a una convergencia muy lenta o a oscilaciones inestables que nunca convergen.

Después de la fase de adaptación de la red, el siguiente paso es la simulación de la salida real de la red. Este proceso se lleva a cabo mediante la función *'sim'*, que implica evaluar cómo la red adaptada responde a las entradas existentes bajo las mismas condiciones de entrenamiento. En esta etapa, se emplea otro buffer de tamaño L que almacena las ultimas muestras de las I señales de referencia. Si se lleva a cabo la simulación de la red con estas entradas, se obtendrán entonces las J salidas de la red que se propagarán por los caminos secundarios y llegarán a los K micrófonos de error.

Así se comenzaría de nuevo el proceso de adaptación de la red para las siguientes muestras de las señales de referencia con las nuevas salidas de los J actuadores y las K señales de error recogidas en los micrófonos.

## 5.3 Resultados simulaciones

Una vez creadas ambas soluciones al problema cancelación de ruido activa, se han simulado para el mismo entorno. La comparación entre ambos algoritmos propuestos para la ANC resulta en diferencias significativas en cuanto a eficiencia temporal y complejidad computacional.

Para realizar la simulación, se han tomado los parámetros de la tabla 2.

| | |
|---|---|
| N | 3000 muestras |
| I | 2 señales de referencia |
| J | 3 micrófonos actuadores |
| K | 4 micrófonos de error |
| L | 120 coeficientes |
| $\mu$ | 0.1< $\mu$ <1 |

**Tabla 3. Parámetros simulación**

### *5.3.1 Simulación ANC multicanal mediante variante del algoritmo LMS*

El algoritmo FxLMS modificado propuesto, ha demostrado un tiempo de ejecución notablemente bajo, de solo 4.33 segundos. Este resultado destaca su idoneidad para situaciones donde la rapidez de respuesta es esencial, como en sistemas de cancelación de ruido en tiempo real. Los gráficos de la simulación muestran una reducción clara y efectiva del ruido en los 4 micrófonos de error, lo que indica que el algoritmo es capaz de ajustar eficazmente sus filtros para minimizar la presencia de ruido disruptivo en el ambiente.

La figura 28, muestra 4 gráficas que representan las señales recogidas en cada k-ésimo micrófono de error. En cada gráfica se puede observar la señal de ruido primaria $d(n)$ en azul y la señal de error $e(n)$ en rojo, que va disminuyendo a medida que avanza el tiempo de procesado.

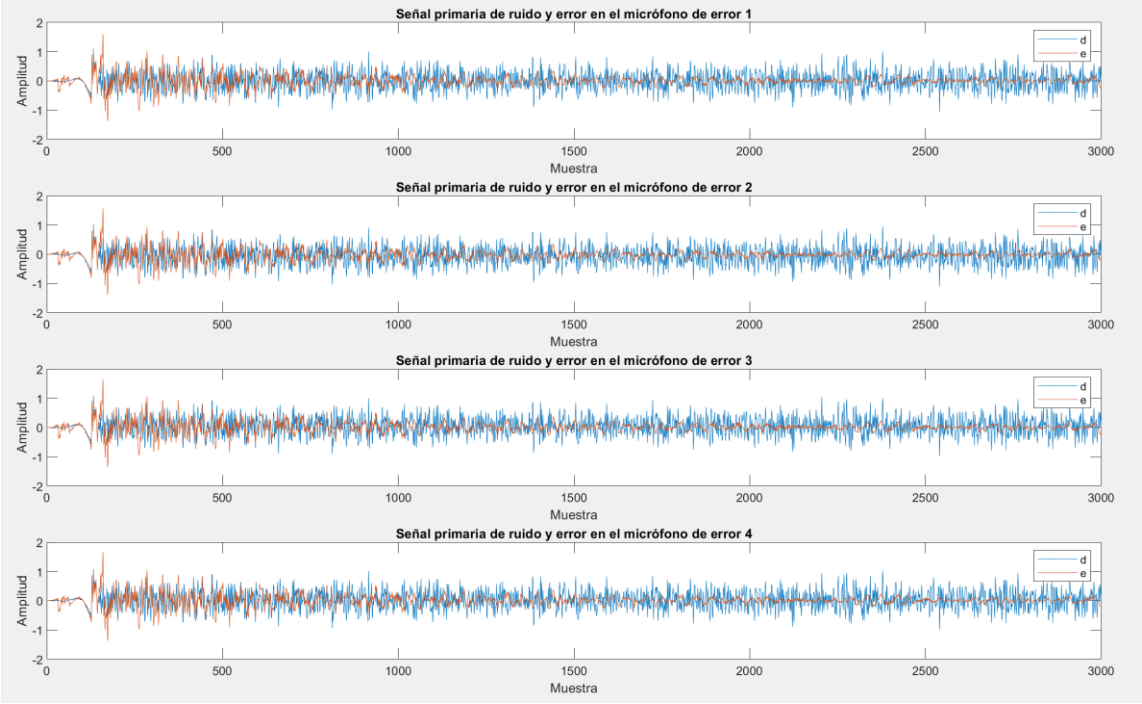


**Figura 28. Simulación FxLMS Modificado**

### *5.3.2 Simulación ANC multicanal mediante red neuronal*

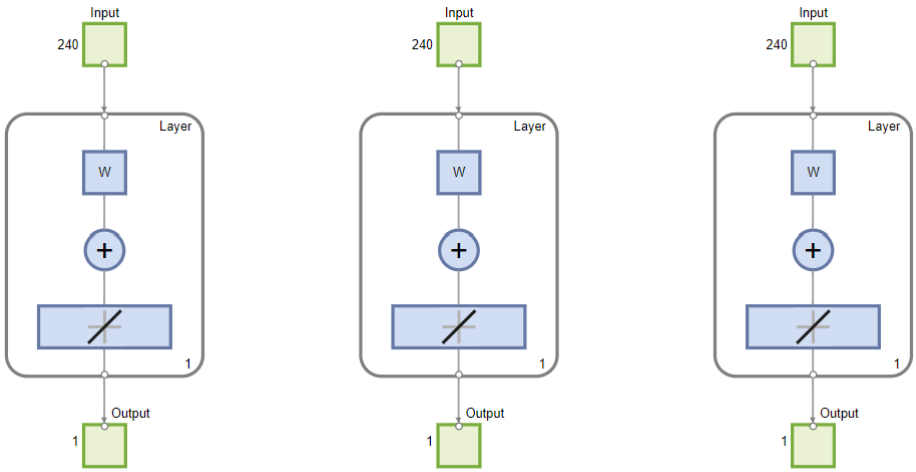


**Figura 29. Red neuronal para configuración 2x3x4**

La red neuronal diseñada, presentó un tiempo de ejecución mucho más largo, de 611.82 segundos, reflejando una complejidad computacional más alta. Aun así también logró una reducción efectiva del ruido mediante la implementación de un sistema ANC que utiliza redes neuronales como una alternativa a los métodos tradicionales.

La figura 30, muestra 4 gráficas que representan las señales recogidas en cada k-ésimo micrófono de error. En cada gráfica se puede observar la señal de ruido primaria $d(n)$ en azul y la señal de error $e(n)$ en rojo, que va disminuyendo a medida que avanza el tiempo de procesado.

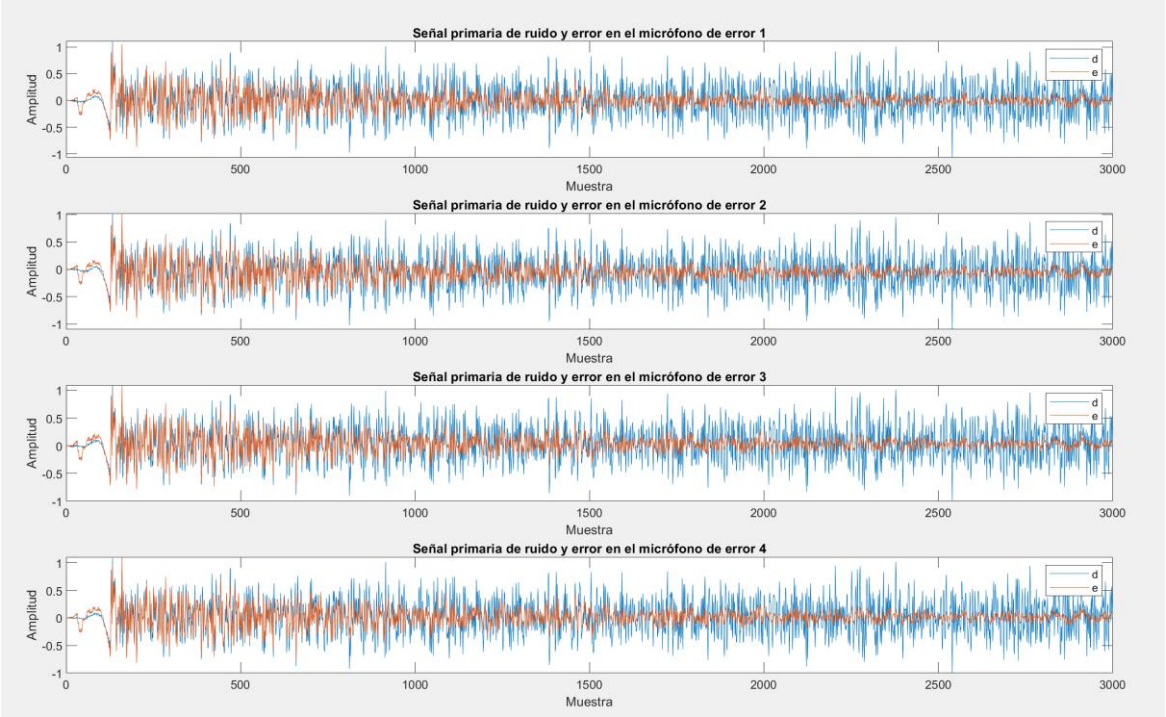


**Figura 30. Simulación red**

# Capítulo 6. Conclusiones

Este estudio ha explorado la implementación de un sistema de ANC multicanal utilizando técnicas avanzadas de aprendizaje automático. Se ha transformado el enfoque tradicional mediante la integración de redes neuronales, proporcionando una nueva perspectiva sobre los métodos adaptativos clásicos. Se ha adaptado una variación del algoritmo FxLMS para funcionar dentro del marco de una red neuronal, tratando los pesos de la red como si fueran los coeficientes de un filtro adaptativo convencional. Esta aproximación ha permitido la adaptación eficaz de estos pesos a través de técnicas de aprendizaje automático, específicamente a través de la retro-propagación.

Los resultados indican que las redes neuronales pueden ofrecer una alternativa viable a los sistemas adaptativos convencionales, tanto en configuraciones monocanal como multicanal. Los objetivos del proyecto se han cumplido, mostrando que el problema propuesto de control de ruido se puede abordar efectivamente con redes neuronales, pero el estudio de estas herramientas también ha revelado limitaciones significativas en términos de coste computacional y eficiencia temporal. Utilizar herramientas genéricas de aprendizaje automático como las de redes neuronales implica un mayor gasto computacional debido a su naturaleza diseñada para tareas más complejas y versátiles.

En comparación directa, el algoritmo LMS modificado propuesto ha demostrado ser más rápido que la solución basada en redes neuronales, aunque ambos métodos han sido efectivos en la reducción del ruido. Este hallazgo subraya la importancia de elegir la herramienta adecuada según el contexto específico y los requisitos de rendimiento del sistema.

Este trabajo de fin de grado no solo ha demostrado la viabilidad de aplicar técnicas de aprendizaje automático en el ANC, sino que también ha abierto caminos para futuras investigaciones, donde se podrían explorar configuraciones más optimizadas y específicas para aplicaciones de reducción de ruido en tiempo real.

# Capítulo 7. Bibliografía

[1] S. J. Elliott and P. A. Nelson, *Active noise control*, in IEEE Signal Processing Magazine, vol. 10, no. 4, pp. 12-35, Oct. 1993, doi: 10.1109/79.248551. keywords: {Active noise reduction; Acoustic noise; Frequency; Loudspeakers; Acoustic signal processing; Acoustic waves;Noise cancellation; Microphones; Low-frequency noise; Signal processing}

[2] Song, Yu. (2022). *Active Noise Cancellation and Its Applications*. Journal of Physics: Conference Series. 2386. 012042. 10.1088/1742-6596/2386/1/012042

[3] Kuo, S.M., & Morgan, D.R. (2000). *Review of DSP algorithms for active noise control*. Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162), 243-248

[4]https://www.shure.com/es-LATAM/audifonos/amplify/noise-cancellation-or-sound-isolation-whats-the-difference

[5] Lueg, P. (1936). *Process of silencing sound oscillations* (Patent No. 2043416). Retrieved from https://www.freepatentsonline.com/2043416.html

[6] Kuo, Sen & Morgan, Dennis. (1999). *Active noise control: A tutorial review*. Proceedings of the IEEE. 87. 943 - 973. 10.1109/5.763310]

[7] Aguayo, G., Enríquez, R., Chao, A., Méndez, Á., & Bustamante, R. *Sistema práctico de cancelación activa de ruido monocanal*. Tecnológico de Monterrey

[8] Swain, Anshuman. (2013). Active Noise Control: Basic Understanding.]

[9] B. Widrow, J. McCool and M. Ball, *The complex LMS algorithm*, in Proceedings of the IEEE, vol. 63, no. 4, pp. 719-720, April 1975, doi: 10.1109/PROC.1975.9807.

[10] M. Bouchard and S. Quednau, "Multichannel recursive-least-square algorithms and fast-transversal-filter algorithms for active noise control and sound reproduction systems," in IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 606-618, Sept. 2000, doi: 10.1109/89.861382

[11] F. Rosenblatt, *Perceptron Simulation Experiments*, in Proceedings of the IRE, vol. 48, no. 3, pp. 301-309, March 1960, doi: 10.1109/JRPROC.1960.287598

[12]https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/

[13] Bengesi, S (2023). *Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers*. Department of Computer Science, Bowie State University.

[14] Amirhossein Javanshir, Thanh Thi Nguyen, M. A. Parvez Mahmud, Abbas Z. Kouzani; *Advancements in Algorithms and Neuromorphic Hardware for Spiking Neural Networks*. Neural Comput 2022; 34 (6): 1289–1328.

[15] Basogain Olabe X. (1998). *Redes neuronales artificiales y sus aplicaciones*. Publicaciones de la Escuela de Ingenieros; Escuela Superior de Ingeniería de Bilbao

[16] CONCEPTOS BÁSICOS SOBRE REDES NEURONALES, Universidad de Sevilla

[17]https://aprendizajeprofundo.github.io/LibroFundamentos/Redes_Neuronales/Cuadernos/Activation_Functions.html

[18] Menacho Chiok, C. H. (2014). *Modelos de regresión lineal con redes neuronales*. Anales Científicos, 75(2), 253-260]

[19]https://ml4a.github.io/ml4a/es/neural_networks/

[20] *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Profesores: Carlos Alberto Ruiz Marta Susana Basualdo

[21] Terven, Juan & Cordova-Esparza, Diana-Margarita & Ramirez-Pedraza, Alfonzo & Chávez Urbiola, Edgar. (2023). *Loss Functions and Metrics in Deep Learning. A Review*.

[22]https://www.diegocalvo.es/funcion-de-coste-redes-neuronales/

[23] Pastor Ruiz, V. (2021). *Desarrollo de redes neuronales para resolución de problemas de estructura*s [Trabajo fin de grado, Universidad Politécnica de Madrid]. Escuela Técnica Superior de Ingenieros Industriales.

[24] Basogain Olabe X. (1998). *Redes neuronales artificiales y sus aplicaciones*, Publicaciones de la Escuela de Ingenieros; Escuela Superior de Ingeniería de Bilbao

[25] K. Burse, R. N. Yadav and S. C. Shrivastava, *Channel Equalization Using Neural Networks: A Review*, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 3, pp. 352-357, May 2010, doi: 10.1109/TSMCC.2009.2038279.