



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO APLICACIONES DEL SONIDO EN EL INTERIOR DE LOS AUTÓMOVILES

Autor: Álvaro González Sardiñas
Director: Manuel Gómez de la Calle

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Aplicaciones del sonido en el interior de los automóviles

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2023/2024 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Álvaro González Sardiñas

Fecha: ...27.../ ...08.../ ...2024...

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Manuel Gómez de la Calle

Fecha: ...27.../ ...08.../ ...2024...

RESUMEN

La innovación tecnológica en el sector automotriz no solo abarca aspectos como la eficiencia de los motores o la incorporación de dispositivos de seguridad avanzados, sino también el confort acústico y la comunicación dentro del vehículo. En este contexto, el siguiente documento de estudio aborda un tema de gran relevancia: las aplicaciones del sonido en el interior de los automóviles, centrándose en dos tecnologías emergentes y de gran impacto: la Cancelación Activa de Sonido, por sus siglas en inglés ANC (Active Noise Cancellation) y el Reconocimiento de Voz, por sus siglas en inglés ASR (Automatic Speech Recognition).

El desarrollo de sistemas avanzados de ANC y ASR en automóviles responde a una demanda creciente por parte de los consumidores, quienes valoran la tranquilidad y la seguridad en sus vehículos. Además, la implementación de estas tecnologías puede tener un impacto significativo en la industria automotriz, ya que permite prescindir de materiales aislantes tradicionales, reduciendo así el peso del vehículo y mejorando su eficiencia energética. De este modo, el proyecto también se alinea con objetivos de sostenibilidad de la agenda 2030, contribuyendo a la disminución de la huella de carbono en el transporte.

Esta investigación propone un análisis exhaustivo de los principios físicos, algoritmos y aplicaciones prácticas de ambas tecnologías, además de evaluar su implementación conjunta a través de un sistema prototipo. Este proyecto no solo busca mejorar la experiencia del usuario al proporcionar un entorno más silencioso y personalizado, sino que también pretende contribuir a la reducción de la fatiga del conductor y, en última instancia, a la disminución de accidentes de tráfico.

Por un lado, la tecnología de Cancelación Activa del Sonido se basa en el principio de interferencia destructiva, donde dos ondas de sonido de igual amplitud, pero fase opuesta se encuentran, resultando en la reducción del sonido percibido. Este fenómeno, demostrado por primera vez por Thomas Young, ha sido aplicado en diversos campos, desde auriculares hasta plantas industriales.

Por otro lado, la tecnología de Reconocimiento de Voz ha avanzado significativamente gracias a los algoritmos de aprendizaje automático y las redes neuronales. Esta tecnología permite la identificación de la voz humana como una señal biométrica, lo que facilita la interacción hombre-máquina y reduce la necesidad de botones físicos en el interior del vehículo.

La solución propuesta en este proyecto consiste en diseñar e implementar un sistema prototipo que combine las funcionalidades de ANC y ASR. Para ello, se han desarrollado varios programas en Python que simulan diferentes escenarios de cancelación de ruido y reconocimiento de voz, permitiendo una evaluación detallada de la calidad de la cancelación y la precisión del reconocimiento.

Los ensayos se centraron en la cancelación activa de los ruidos característicos del interior del vehículo, así como en la separación de voces y la transcripción de la voz en texto. Además, se realizó un análisis espectral de audio en tiempo real, lo que facilitó la identificación de frecuencias dominantes, parámetro fundamental a la hora de reconocer la voz y la visualización del espectro de frecuencia e intensidad del sonido.

Los resultados de las simulaciones indican que la combinación de ANC y ASR en un solo sistema prototipo es factible y puede mejorar significativamente la experiencia del usuario en el interior del vehículo. La tecnología de cancelación activa de sonido demostró ser eficaz en la atenuación de los ruidos de baja frecuencia, mientras que el sistema de reconocimiento de voz mostró una alta precisión en la identificación de comandos de voz en diferentes condiciones acústicas.

El estudio concluye que la integración de estas tecnologías puede abrir nuevas vías para el desarrollo de la automoción, permitiendo una mayor personalización y confort para los usuarios. Además, se destaca la importancia de seguir investigando en este campo para mejorar la eficacia de los sistemas y reducir aún más el peso y el consumo de combustible de los vehículos.

El análisis económico realizado en el proyecto revela que la implementación de sistemas avanzados de ANC y ASR en vehículos puede ser económicamente viable, siempre y cuando

los consumidores estén dispuestos a pagar un precio adicional por estos beneficios. Se estima que el costo de estas tecnologías podría ser compensado por la reducción en el consumo de materiales aislantes y la mejora en la eficiencia energética del vehículo.

De cara al futuro, se espera que las tecnologías de cancelación activa de sonido y reconocimiento de voz sigan evolucionando, incorporando algoritmos más sofisticados y adaptándose a las nuevas demandas del mercado. Además, se prevé que estas tecnologías puedan ser aplicadas en otros sectores, como el de la aviación o el ferroviario, ampliando así su impacto y beneficios.

ABSTRACT

Technological innovation in the automotive sector not only encompasses aspects such as engine efficiency or the incorporation of advanced safety devices but also includes acoustic comfort and communication within the vehicle. In this context, the following study document addresses a topic of great relevance: the applications of sound inside automobiles, focusing on two emerging and highly impactful technologies: Active Noise Cancellation (ANC) and Automatic Speech Recognition (ASR).

The development of advanced ANC and ASR systems in automobiles responds to a growing demand from consumers who value tranquility and safety in their vehicles. Moreover, the implementation of these technologies can have a significant impact on the automotive industry, as it allows for the elimination of traditional insulating materials, thereby reducing the vehicle's weight and improving its energy efficiency. In this way, the project also aligns with the sustainability goals of the 2030 agenda, contributing to the reduction of the carbon footprint in transportation.

This research proposes an in-depth analysis of the physical principles, algorithms, and practical applications of both technologies, in addition to evaluating their joint implementation through a prototype system. This project not only aims to enhance the user experience by providing a quieter and more personalized environment but also seeks to contribute to reducing driver fatigue and, ultimately, to decreasing traffic accidents.

On the one hand, Active Noise Cancellation technology is based on the principle of destructive interference, where two sound waves of equal amplitude but opposite phase meet, resulting in reduced perceived sound. This phenomenon, first demonstrated by Thomas Young, has been applied in various fields, from headphones to industrial plants.

On the other hand, Voice Recognition technology has advanced significantly thanks to machine learning algorithms and neural networks. This technology enables the identification of the human voice as a biometric signal, facilitating human-machine interaction and reducing the need for physical buttons inside the vehicle.

The proposed solution in this project involves designing and implementing a prototype system that combines the functionalities of ANC and ASR. To achieve this, several Python programs have been developed to simulate different scenarios of noise cancellation and voice recognition, allowing for a detailed evaluation of the quality of noise cancellation and the accuracy of recognition.

The trials focused on the active cancellation of characteristic noises inside the vehicle, as well as on voice separation and voice-to-text transcription. Additionally, a real-time spectral analysis of audio was conducted, facilitating the identification of dominant frequencies—a fundamental parameter for voice recognition—and the visualization of the sound's frequency spectrum and intensity.

The results of the simulations indicate that combining ANC and ASR into a single prototype system is feasible and can significantly improve the user experience inside the vehicle. Active noise cancellation technology proved effective in attenuating low-frequency noises, while the voice recognition system demonstrated high accuracy in identifying voice commands under different acoustic conditions.

The study concludes that integrating these technologies could open new avenues for automotive development, allowing for greater personalization and comfort for users.

Furthermore, it highlights the importance of continuing research in this field to improve system efficiency and further reduce vehicle weight and fuel consumption.

The economic analysis conducted in the project reveals that implementing advanced ANC and ASR systems in vehicles can be economically viable, as long as consumers are willing to pay a premium for these benefits. It is estimated that the cost of these technologies could be offset by reducing the use of insulating materials and improving the vehicle's energy efficiency.

Looking ahead, it is expected that active noise cancellation and voice recognition technologies will continue to evolve, incorporating more sophisticated algorithms and adapting to new market demands. Additionally, it is anticipated that these technologies could be applied in other sectors, such as aviation or rail, thereby expanding their impact and benefits.

AGRADECIMIENTOS

En primer lugar, quiero agradecer profundamente a mi familia, quienes han estado siempre a mi lado, aguantando mis altibajos y brindándome su apoyo incondicional durante todo este proceso. Gracias por ser mi pilar fundamental, por comprender mis ausencias y por apoyarme en cada momento.

A Antolín, quiero expresarle mi más sincero agradecimiento por la beca que me otorgó, la cual fue crucial para poder dedicarme de lleno a este proyecto. Gracias por confiar en mí y por brindarme esta oportunidad tan valiosa.

A mis amigos, quiero agradecerles por los ánimos constantes. Su apoyo y sus momentos de distracción fueron esenciales para mantener mi motivación y para recordar que, más allá del trabajo y el estudio, siempre hay espacio para la amistad y la risa.

No puedo dejar de agradecer a mis profesores, quienes con su dedicación y conocimiento me guiaron a lo largo de estos años. Gracias por compartir su sabiduría y por impulsarme a dar siempre lo mejor de mí.

Finalmente, quiero expresar mi más profundo agradecimiento a mi director de proyecto, quien, con su experiencia, orientación y paciencia, me guio en cada etapa de este trabajo. Gracias por su confianza, sus valiosas sugerencias y por inspirarme a alcanzar un nivel de excelencia que nunca pensé posible.

ÍNDICE DE LA MEMORIA

CAPÍTULO 1. INTRODUCCIÓN	13
1.1 MOTIVACIÓN DEL PROYECTO	13
1.2 OBJETIVOS.....	14
1.3 AGENDA 2030	16
1.4 ESTRUCTURA DEL DOCUMENTO.....	19
1.5 ACRÓNIMOS	21
CAPÍTULO 2. ESTADO DE LA CUESTIÓN	23
2.1 CANCELACIÓN ACTIVA DEL SONIDO	23
2.1.1 FUNCIONAMIENTO GENERAL	24
2.1.2 ELEMENTOS PARA EL CONTROL ACTIVO DEL RUIDO	26
2.1.3 ALGORITMOS DE CONTROL	29
2.1.4 FILTROS ADAPTATIVOS.....	30
2.1.5 ALGORITMOS ADAPTATIVOS.....	34
2.1.6 LA IA EN LA CANCELACIÓN ACTIVA DE SONIDO	39
2.1.7 EMPRESAS DEL SECTOR	40
2.1.8 PRÓXIMOS AVANCES	42
2.2 RECONOCIMIENTO DE LA VOZ	45
2.2.1 FUNCIONAMIENTO GENERAL.....	46
2.2.2 DIGITALIZACIÓN DE LA VOZ.....	48
2.2.3 EMPRESAS DEL SECTOR – ESTUDIO DEL MERCADO	53
CAPÍTULO 3. SOLUCIÓN PROPUESTA	59
3.1 PROGRAMA DE CANCELACIÓN ACTIVA DEL SONIDO	61
3.1.1 CANCELACIÓN ACTIVA DE LOS RUIDOS CARACTERÍSTICOS.....	61
3.2 EXPLICACIÓN PROGRAMA DE RECONOCIMIENTO DE SONIDO	67
3.2.1 EXPLICACIÓN PROGRAMA ESPECTRO DE AUDIO EN TIEMPO REAL	67
3.2.2 EXPLICACIÓN PROGRAMA DICTADO DE VOZ	71
3.2.3 EXPLICACIÓN DEL PROGRAMA DE RECONOCIMIENTO DE VOZ.....	72
3.2.4 EXPLICACIÓN PROGRAMA SEPARACION DE VOCES	74
CAPÍTULO 4. ENSAYOS Y RESULTADOS.....	76
4.1 CÓDIGOS	76

ICAI	ICADE	CIHS
------	-------	------

4.1.1	<i>CODIGOS PYTHON CANCELACIÓN ACTIVA DE SONIDO</i>	76
4.1.2	<i>CÓDIGOS DE PYTHON RECONOCIMIENTO DE SONIDO</i>	80
4.2	RESULTADOS SOBRE LA SIMULACIÓN DE LA CANCELACIÓN ACTIVA DEL SONIDO	88
4.3	RESULTADOS SOBRE LAS SIMULACIONES DEL RECONOCIMIENTO DE LA VOZ	91
CAPÍTULO 5.	<i>ESTUDIO ECONÓMICO</i>	94
CAPÍTULO 6.	<i>CONCLUSION</i>	96
CAPÍTULO 7.	<i>BIBLIOGRAFÍA</i>	98

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Experimento doble rendija Thomas Young (Beléndez, Thomas Young y la naturaleza ondulatoria de la luz, 2015).....	24
Ilustración 2 Interferencias constructivas y destructivas de ondas (Sánchez).....	25
Ilustración 3 Diagrama representativo del funcionamiento de un sistema ANC	26
Ilustración 4 Diagrama de bloques de filtro FIR (Chávez, 2005)	31
Ilustración 5 Diagrama de bloques de filtro IIR (POLARIDAD.ES).....	32
Ilustración 6 Diagrama de bloques algoritmo adaptativo	35
Ilustración 7 Algoritmos genéticos (Batista, 2009).....	38
Ilustración 8 IA en la cancelación activa del sonido (Alonso, 2021).....	39
Ilustración 9 Sistema cancelación de sonido Hyundai	41
Ilustración 10 Micrófonos colocados en un coche de prueba -Universidad de Le Mans (Romero, El algoritmo que ayudará a viajar en coche con niños: burbujas de sonido delimitadas a cada asiento, 2022).....	43
Ilustración 11 Recalm ancor infotainment-system (RECALM, 2024).....	43
Ilustración 12 Etapas de un sistema de ASR (Rufiner, 2004)	47
Ilustración 13 Transformada de Fourier	49
Ilustración 14 Escala de Mel	50
Ilustración 15 Esquema de una red neuronal.....	52
Ilustración 16 Estructura red neuronal	53
Ilustración 17 Diagrama R328.....	58
Ilustración 18 Diagrama disposición de elementos	60
Ilustración 19 Diagrama del programa de cancelación activa de ruidos	61
Ilustración 20 Diagrama del programa de espectro de audio en tiempo real	67
Ilustración 21 Diagrama del programa de dictado de voz.....	71
Ilustración 22 Diagrama del programa de reconocimiento de voz.....	72
Ilustración 23 Diagrama del programa de separación de voces	74
Ilustración 24 Atenuación ruido blanco	89
Ilustración 25 Atenuación ruido rosa	90

Ilustración 26 Atenuación ruido marrón.....	90
Ilustración 27 Espectro de frecuencias en tiempo real	91
Ilustración 28 Frecuencia dominante en tiempo real.....	92
Ilustración 29 Mando de opciones del programa.....	92

ÍNDICE DE TABLAS

Tabla 1 Tabla de características recogidas a través de un estudio de mercado	55
Tabla 2 Tabla de precios de instalación de los elementos necesarios	94
Tabla 3 Tabla de precios de licencias y desarrollo de software	95

CAPÍTULO 1. INTRODUCCIÓN

1.1 MOTIVACIÓN DEL PROYECTO

Los temas tratados en este proyecto, la cancelación activa del sonido y el reconocimiento de voz representan un avance significativo en la intersección de la innovación tecnológica y la mejora de la experiencia del usuario. Por un lado, la cancelación activa de sonido consiste en la atenuación de las ondas a través de una onda de mismas características, pero desfasada. En cuanto al reconocimiento de voz, consiste en una tecnología capaz de obtener nuestra voz como signo biométrico, que busca reducir el número de botones creando una comunicación usuario-computadora únicamente a través de la voz.

Las tecnologías de cancelación activa de sonido y reconocimiento de la voz tienen el potencial de transformar los viajes en automóviles, proporcionando un entorno más silencioso y cómodo que reduce el estrés y la fatiga del conductor y de los pasajeros. Se trata de un tema de gran importancia ya que, al aumentar la calidad de los viajes, reducir la fatiga del conductor y disminuir las distracciones, el número de accidentes de tráfico se puede ver reducido.

Desde el punto de vista económico, la implementación de sistemas avanzados de cancelación de ruido y reconocimiento de voz puede impulsar la competitividad de las marcas automotrices, atrayendo a consumidores que valoran la tranquilidad y la seguridad. Además, gracias a la implementación de sistemas de cancelación activa de sonido, se puede prescindir del uso de materiales aislantes encargados de cancelar el ruido lo máximo posible. Al poder prescindir de estos materiales, el vehículo tiene menos peso y consume menos combustible, por lo que puede llegar a ser más competitivo en precio.

Desde una perspectiva tecnológica, este sistema integra sofisticados algoritmos de procesamiento de señales y aprendizaje automático, posicionándose a la vanguardia de la inteligencia artificial y la ingeniería de sonido.

1.2 OBJETIVOS

En el presente proyecto se busca cumplir con los siguientes objetivos:

1. Explicación y análisis en profundidad de los sistemas de cancelación y reconocimiento de sonido:

El primer objetivo del proyecto se centra en realizar una descripción detallada del funcionamiento de los sistemas de cancelación y reconocimiento de sonido. Para ello, se analizarán los componentes de ambos sistemas, profundizando en sus características y funcionalidades específicas. Este Aobjetivo se desarrollará a lo largo del CAPÍTULO 2.

2. Análisis de proveedores para el desarrollo de los sistemas:

El segundo objetivo del proyecto consiste en identificar y analizar los distintos proveedores que se necesitan para el desarrollo de los sistemas de reconocimiento y cancelación activa del sonido. Se estudiarán las características y funcionalidades que ofrecen los diferentes proveedores, comparando sus ventajas e inconvenientes.

El objetivo de este análisis es seleccionar el sistema que mejor se adapten a las necesidades del proyecto, teniendo en cuenta factores como la calidad de los productos y servicios que ofrecen, el precio, la experiencia en el sector y la capacidad de soporte técnico.

Este objetivo se desarrollará a lo largo del CAPÍTULO 2.

3. Propuesta de una solución

En este objetivo se seleccionarán los sistemas previamente explicados en el Capítulo 2 para el diseño de un sistema de cancelación de sonido y reconocimiento de voz. Se detallará el funcionamiento de los programas necesarios para su implementación en el software, con un hardware ya elegido. Se programarán diferentes algoritmos de cancelación del sonido y se desarrollarán varios programas para el reconocimiento del hablante, con la posibilidad de ajustar diferentes parámetros. Estos programas podrían requerir un proceso de aprendizaje

previo, en el que se capturará la voz de cada persona junto con su nombre. Posteriormente, el programa será capaz de analizar la voz y reconocer a quien pertenece.

Este objetivo se desarrollará a lo largo del CAPÍTULO 3.

4. Comprobaciones de la eficacia de los sistemas diseñados

El cuarto objetivo del proyecto se centra en la definición de diferentes test, representativos de los ruidos presentes en el automóvil, así como aquellos de reconocimiento de la voz. Posteriormente, se realizarán una serie de simulaciones en Python para evaluar el rendimiento de los sistemas de cancelación y reconocimiento de sonido, con posterior análisis de los resultados.

Este objetivo se desarrollará a lo largo del **¡Error! No se encuentra el origen de la referencia.**

5. Análisis del mercado automovilístico y viabilidad económica:

El quinto y último objetivo del proyecto consiste en realizar un análisis del mercado automovilístico actual para evaluar la viabilidad económica de las nuevas tecnologías. Se estudiará el sobrecoste que estos sistemas tendría en el precio del automóvil. Para ello, se estimará el costo aproximado de las nuevas tecnologías y su implementación. En base a los resultados del análisis, se establecerá una conclusión sobre la rentabilidad de lanzar al mercado estas tecnologías en el contexto actual.

Este último objetivo se desarrollará en el CAPÍTULO 5.

1.3 AGENDA 2030

Este proyecto cumple con los objetivos propuestos en la agenda 2030.

Objetivo 3: Garantizar una vida sana y promover el bienestar para todos en todas las edades:

Al reducir significativamente el ruido ambiental en el interior del vehículo, este sistema contribuye a un ambiente más relajado y menos estresante para el conductor y los pasajeros. Un entorno acústico más agradable puede reducir la fatiga, mejorar la concentración y, en última instancia, promover una mejor salud mental.

Objetivo 7: Garantizar el acceso a una energía asequible, segura, sostenible y moderna:

Al reducir la necesidad de materiales aislantes tradicionales y optimizar el consumo de energía, este sistema promueve una producción y un consumo más responsables, garantizando una energía segura sostenible y más asequible. Además, al extender la vida útil de los vehículos gracias a un mayor confort y seguridad, se reduce la necesidad de reemplazarlos con frecuencia.

Objetivo 8: Promover el crecimiento económico inclusivo y sostenible, el empleo y el trabajo decente para todos:

El desarrollo y la implementación de esta tecnología requieren de una fuerza laboral calificada en áreas como ingeniería de sonido, programación y diseño de vehículos. Esto genera nuevos empleos y estimula la economía local. Además, la innovación en este sector puede atraer inversiones y fomentar el crecimiento económico a largo plazo.

Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación:

La creación de un sistema de cancelación activa de sonido y reconocimiento de voz implica una inversión significativa en investigación y desarrollo, lo que fomenta la innovación y la

creación de nuevas tecnologías. Además, la integración de esta tecnología en los vehículos requiere de una infraestructura de producción y distribución adecuada.

Objetivo 12: Garantizar modalidades de consumo y producción sostenibles:

Pretende construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación.

El crecimiento económico, el desarrollo social y la acción por el clima dependen en gran medida de las inversiones en infraestructuras, el desarrollo industrial sostenible y el progreso tecnológico. Ante la rápida evolución del panorama económico mundial y el aumento de las desigualdades, el crecimiento sostenido debe implicar una industrialización que, en primer lugar, haga accesibles las oportunidades a todas las personas y, en segundo lugar, se apoye en la innovación y en infraestructuras resistentes.

Los objetivos de este proyecto están en la línea del objetivo 9 de la agenda 2030. Buscan innovar en la comodidad en el transporte.

Objetivo 13: Adoptar medidas urgentes para combatir el cambio climático y sus efectos:

La crisis climática no va a parar, y la comunidad internacional rehúye el compromiso necesario para revertirla. El período comprendido entre 2010 y 2019 fue la década más calurosa que se haya registrado jamás y trajo consigo incendios forestales, huracanes, sequías, inundaciones y otros desastres naturales masivos en todos los continentes. Es necesario aumentar la ambición y actuar en todos los niveles para hacer frente al cambio climático. Están sucediendo muchos cambios en todo el mundo, y las inversiones en energías renovables se han disparado.

El hecho de conseguir una cancelación activa del sonido en el interior de la cabina afecta en el consumo de los materiales que hay que emplear. Al tener que emplear menos materiales

aislantes (cancelación pasiva del sonido) se consume menos combustible lo cual impacta positivamente en el treceavo objetivo de la agenda 2030.

Objetivo 17: Revitalizar la Alianza Mundial para el desarrollo sostenible:

Pretende revitalizar la alianza mundial para el desarrollo sostenible. La Agenda 2030 es universal y exige la implicación de todos los países, tanto desarrollados como en desarrollo, para garantizar que nadie se quede atrás. También requiere la colaboración entre los gobiernos, el sector privado y la sociedad civil.

Los Objetivos de Desarrollo Sostenible solo se pueden conseguir con asociaciones mundiales sólidas y cooperación para garantizar que nadie se quede atrás en nuestro camino hacia el desarrollo.

Hago referencia a este objetivo ya que es necesaria la colaboración del sector privado para conseguir lograr los objetivos, de lo contrario la dificultad crecería exponencialmente. (Gómara, 2024).

1.4 ESTRUCTURA DEL DOCUMENTO

El primer capítulo está dedicado a la motivación del proyecto donde se trata de representar la relevancia del proyecto. En este capítulo también se explicarán los objetivos que se trataran de desarrollar a lo largo del documento y los objetivos de la agenda 2030 ligados con este.

El segundo capítulo está dedicado al estado de la cuestión, en el cual se ofrecerá una revisión detallada de la tecnología de cancelación activa del sonido y del reconocimiento de la voz. Este capítulo proporcionará una base teórica mediante la definición de elementos clave, la explicación de los principios de funcionamiento de ambas tecnologías y un análisis de los proveedores. El objetivo es establecer un contexto claro y comprensible que permita una mejor comprensión de los componentes tecnológicos involucrados.

En el siguiente capítulo, se presentará la propuesta de solución, donde se delinearán cómo se integrarán las tecnologías y donde se dispondrán los sistemas para el correcto funcionamiento.

El capítulo 4 estará dedicado a la realización de ensayos, en los cuales se simulará el funcionamiento del software para evaluar su desempeño en condiciones controladas. Este capítulo se centrará en la metodología de prueba. Se presentarán los resultados de las simulaciones y se concluirá el grado en que el sistema cumple con los objetivos definidos

En el capítulo 5 se realizará una estimación económica del sistema, en la cual se analizarán los costos asociados con su desarrollo, implementación y mantenimiento. Esta sección incluirá una evaluación detallada de los recursos necesarios, los costos de los componentes tecnológicos, y los posibles gastos operativos. Posteriormente, se ofrecerán las conclusiones generales del estudio, donde se evaluará si se han cumplido los objetivos iniciales y se reflexionará sobre la viabilidad y los posibles impactos del sistema. Este análisis final permitirá una comprensión completa del éxito del proyecto y de las áreas que podrían requerir mejoras o ajustes.

Por último en el capítulo 6 se realizarán las conclusiones y se analizará si se han cumplido o no con los objetivos.

1.5 ACRÓNIMOS

ADC.....	Analog-to-Digital Converter
ANC.....	Active Noise Cancellation
ASR	Automatic Speech Recognition
CODEC.....	Coder-Decoder
CPU	Central Processing Unit
DAC.....	Digital-to-Analog Converter
DDR2.....	Double Data Rate 2
DMIC.....	Digital Microphone
DSP.....	Digital Signal Processor
ECM	Mean Squared Error
FFT	Fast Fourier Transform
FIR.....	Finite Impulse Response
I2S.....	Integrated Interchip Sound
IA.....	Artificial Intelligence
IIR.....	Infinite Impulse Response
LMS	Least Mean Squares
MB	Megabyte
MEMS	MicroElectroMechanical Systems

MFCCs Mel-Frequency Cepstral Coefficients

MSE Mean Squared Error

PDM Pulse Density Modulation

RANC Road Active Noise Control

VAD Voice Activity Detection

CAPÍTULO 2. ESTADO DE LA CUESTIÓN

2.1 CANCELACIÓN ACTIVA DEL SONIDO

En primer lugar, sobre el tema de cancelación de sonido es un tema que se está aplicando en varios campos. En el área en el que más presente se encuentra es en los auriculares ya que se trata de una tecnología muy demandada por los melómanos de la música. Ya son muchas de las grandes empresas tecnológicas como Apple o Samsung las que han incluido esta función en sus productos, sin embargo, la cancelación activa del sonido de los auriculares es mucho más sencilla que el sistema que se está analizando en este proyecto, esto es porque los auriculares se encuentran directamente en la oreja lo cual facilita mucho la tarea a la hora de atenuar las ondas.

La cancelación activa del sonido también se encuentra muy presente en las plantas industriales. Existen equipos rotantes que emiten elevados niveles de ruido, de frecuencias discretas relacionadas con su velocidad de rotación. Los elevados niveles de sonido causan daños al oído más o menos graves según la intensidad y el tiempo de exposición. En el interior de los vehículos, no existe un único foco emisor, sino varios. Los distintos sonidos ruido blanco, ruido rosa o ruido marrón penetran en el interior de la cabina del vehículo refractándose y reflejándose por las paredes del vehículo viéndose modificadas sus frecuencias dominantes y dificultando su atenuación. El ruido blanco es un tipo de señal con igual potencia en todas las frecuencias audibles, útil para probar la respuesta de un sistema a un amplio espectro de sonidos. El ruido rosa tiene más energía en frecuencias bajas y menos en altas, parecido al ruido de rodadura de los neumáticos, y se percibe más equilibrado por el oído humano. El ruido marrón, o browniano, cae más rápidamente en intensidad a medida que aumenta la frecuencia, predominando en frecuencias bajas, similar a los sonidos graves del motor o los ruidos estructurales del vehículo

2.1.1 FUNCIONAMIENTO GENERAL

La cancelación activa del sonido es un fenómeno explicado por Thomas Young a través del “*Experimento de interferencia de Young* “ (Beléndez, Thomas Young y la naturaleza ondulatoria de la luz, 2015) . En este experimento, el científico inglés demostró de forma convincente la naturaleza ondulatoria de la luz mediante el experimento de la doble rendija, desafiando las ideas de Newton. Este experimento, considerado uno de los más bellos de la física, mostró que, al superponer la luz de dos rendijas estrechas sobre una pantalla, se produce un patrón de franjas claras y oscuras (interferencia), lo que solo podía explicarse por el comportamiento ondulatorio de la luz.

De esta forma, se consigue en una zona determinada la denominada interferencia destructiva, esta zona recibe el nombre de zona silencio. La Ilustración 1 ejemplifica el experimento de Thomas Young. Se puede ver como las ondas de luz proyectadas a través de las rendijas se deconstruyen en las zonas negras de la placa (interferencia destructiva) y se superponen en las zonas luminosas (interferencia constructiva). Esto se debe a que en las zonas luminosas coinciden la misma onda sin desfase, sin embargo, en las zonas oscuras el fenómeno es al contrario: confluyen la misma onda, pero una de ellas se encuentra desfasada.

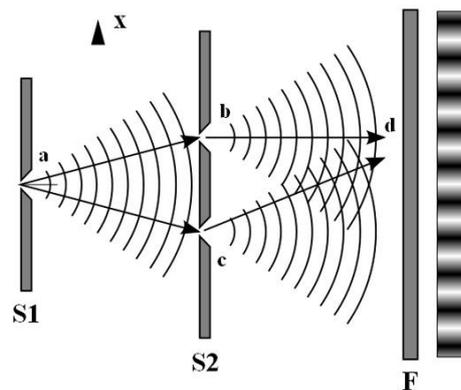


Ilustración 1 Experimento doble rendija Thomas Young (Beléndez, Thomas Young y la naturaleza ondulatoria de la luz, 2015)

El ANC en el interior de un vehículo se basa en el principio de interferencia destructiva:

Cuando dos ondas se encuentran en el mismo medio, no pueden evitar interactuar. Esta interacción, conocida como interferencia, se manifiesta como la suma de las perturbaciones individuales de cada onda, dando lugar a una onda resultante con características únicas.

En el caso de las ondas sonoras, la interferencia tiene un papel fundamental en la formación de ondas estacionarias resonantes. Estas ondas "atrapadas" en un espacio vibran con una frecuencia específica, creando patrones de sonido distintivos. La clave de la interferencia radica en la fase de las ondas. Si las crestas de dos ondas coinciden (interferencia constructiva), como en el primer caso de la Ilustración 2 Interferencias constructivas y destructivas de ondas se suman, creando una onda de mayor amplitud. En cambio, si una cresta se encuentra con un valle (interferencia destructiva), como en el segundo caso de la Ilustración 2 Interferencias constructivas y destructivas de ondas se restan, anulándose parcialmente o incluso por completo. (Sánchez)

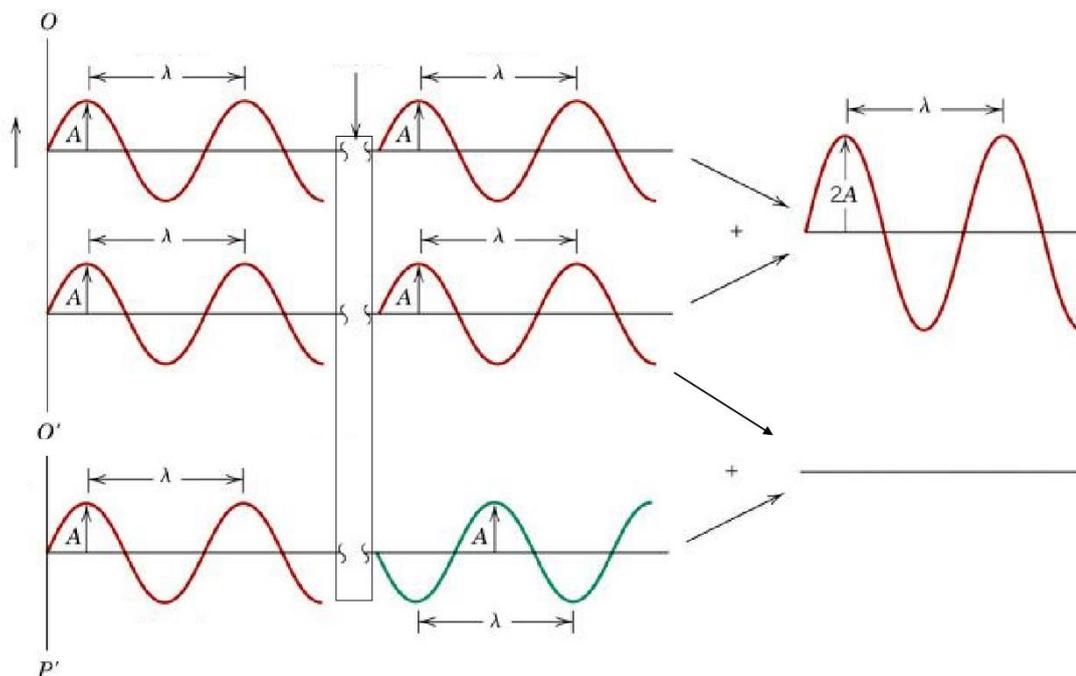


Ilustración 2 Interferencias constructivas y destructivas de ondas (Sánchez)

El funcionamiento del ANC, que está representado en la Ilustración 3, se puede resumir en los siguientes pasos:

1. En primer lugar, se emplean micrófonos para captar el ruido ambiental, es decir, se realiza la lectura de las ondas entrantes al vehículo.
2. Posteriormente, una vez el programa electrónico ha leído esta señal la envía a los altavoces, los cuales reproducen una señal de cancelación, se crea una señal idéntica a la recibida de igual amplitud, pero de fase opuesta.
3. Cuando las dos señales se superponen, se produce lo que se denomina una interferencia destructiva, cancelándose la onda y consiguiéndose eliminar el ruido en el interior del vehículo.

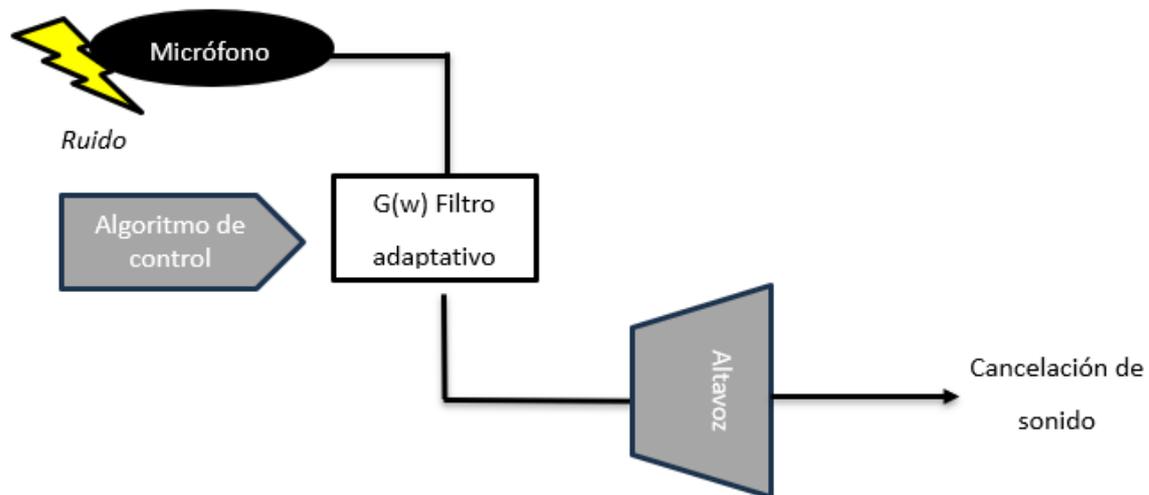


Ilustración 3 Diagrama representativo del funcionamiento de un sistema ANC

2.1.2 ELEMENTOS PARA EL CONTROL ACTIVO DEL RUIDO

Se utilizan múltiples micrófonos y altavoces estratégicamente ubicados en el interior del vehículo para captar el ruido ambiental. Por lo general, se colocan en puntos como en el reposacabezas, el revestimiento de las puertas, el techo y los pilares del automóvil. La cantidad de micrófonos y altavoces utilizados puede variar en función del modelo del vehículo para una cobertura adecuada y para obtener una señal precisa del ruido en diferentes áreas del vehículo. Estos micrófonos pueden ser de distintos tipos, a continuación, se detallan los diferentes tipos de micrófonos y altavoces empleados. La disposición de estos elementos

depende del tipo de vehículo, ya que su disposición varía dependiendo de las ondas reflejadas, que está directamente relacionado con la forma del vehículo.

2.1.2.1 MICRÓFONOS

2.1.2.1.1 MICRÓFONOS DE CONDENSADOR

Un micrófono de condensador convierte ondas sonoras en señales eléctricas a través y un condensador. Estos micrófonos emplean el principio de capacitancia. Esto es, la capacidad de un sistema para almacenar carga eléctrica cuando hay una diferencia de potencial entre dos conductores. Su funcionamiento se basa en un delgado diafragma conductor, usualmente de mylar, una lámina de poliéster recubierta de oro, colocado cerca de una placa posterior. Cuando las ondas sonoras hacen vibrar el diafragma, la distancia entre este y la placa posterior varía, alterando la capacitancia y produciendo fluctuaciones de voltaje que se amplifican para generar la señal de audio. La placa posterior prepolarizada es crucial, ya que mantiene una carga estática constante en el diafragma, permitiendo una rápida y precisa respuesta a las ondas sonoras.

2.1.2.1.2 MICRÓFONOS MEMS

Los MEMS (por sus siglas en inglés MicroElectro Mechanical Systems) tienen el mismo funcionamiento que el de un micrófono condensador. Sin embargo, son dispositivos que integran componentes electrónicos y mecánicos a escala microscópica, por lo que su tamaño es más reducido. La tendencia es utilizar este tipo de micrófonos para los ANC, ya que son de gran utilidad para aplicaciones de electrónica debido al bajo consumo de energía (su característica más destacable) lo cual es esencial para el desarrollo de los automóviles eléctricos. (Herrera, 2023)

Estos micrófonos son capaces de captar sonidos de frecuencias entre 20 Hz y 20 KHz.

Los micrófonos MEMS son una opción preferente debido a su alta eficiencia energética. Esta característica resulta especialmente valiosa en un contexto de creciente desarrollo de los vehículos eléctricos, donde la optimización del consumo energético es fundamental. En

cuanto al precio no existen grandes diferencias. Bien es cierto que los micrófonos condensadores son más robustos, sin embargo, esta característica no es tan importante en comparación con la eficiencia energética. En cuanto al rango dinámico de ambos micrófonos, existen diferencias. En el caso del micrófono condensador su rango es de unos 120 dB sin embargo en el caso de los micrófonos MEMS es de 100 dB, esto se debe a su menor tamaño. Pese a esta diferencia en el rango dinámico seguirán siendo preferibles los micrófonos MEMS.

2.1.2.2 ALTAVOCES

En un sistema de cancelación activa de ruido (ANC) utilizado en un automóvil, se pueden utilizar diferentes tipos de altavoces para reproducir la señal de cancelación y lograr la reducción del ruido no deseado en el interior del vehículo. A continuación, se mencionan algunos de los tipos de altavoces comúnmente utilizados en los sistemas ANC de automóviles:

2.1.2.2.1 ALTAVOCES DE RANGO COMPLETO

Los altavoces de rango completo están diseñados para reproducir todo el rango de frecuencias audibles para el ser humano, de 20 Hz a 20 KHz, aunque su rendimiento puede variar según las características físicas del diseño. Normalmente, estos altavoces utilizan un único cono para traducir los impulsos eléctricos de la bobina en sonido. Dado que los requisitos de un altavoz de rango completo incluyen una buena respuesta tanto en frecuencias bajas como altas, lo cual habitualmente es contradictorio en términos constructivos, generalmente los altavoces de rango completo se limitan a cubrir el espectro de audio por encima de 100 Hz, dejando que las frecuencias más bajas sean manejadas por un subwoofer especial para el refuerzo de bajas frecuencias (GemeratePress, 2024).

2.1.2.2.2 ALTAVOCES DE SUBGRAVES (SUBWOOFERS))

Estos altavoces se especializan en la reproducción de frecuencias de audio más bajas (por debajo de los 200 Hz). En el contexto de los sistemas ANC de automóviles, estos altavoces son utilizados para reproducir las frecuencias más bajas del ruido no deseado, como el ruido

del motor o las vibraciones. Estos altavoces suelen estar ubicados en el maletero o en áreas específicas del habitáculo del automóvil.

El hecho de conseguir una buena o mala atenuación del ruido depende de la distancia a la que se encuentren ambas fuentes. Cuanta mayor distancia entre la fuente receptora (el micrófono) y la emisora (el altavoz) peor será la cancelación del ruido. En el caso concreto de un automóvil, al tratarse de un recinto cerrado el campo acústico no está creado únicamente de la onda primaria (la que queremos eliminar) sino también por múltiples ondas procedentes del conjunto de reflexiones que se producen con las paredes del automóvil generándose un campo con cierta energía acústica potencial en el interior del recinto. Es por este aspecto por el que el desarrollo de esta tecnología es de gran dificultad.

2.1.3 ALGORITMOS DE CONTROL

El sistema consta de micrófonos que captan el ruido ambiental (onda de presión acústica inicial) y lo convierten en una señal eléctrica analógica. Esta señal se digitaliza y se envía a un controlador. El controlador utiliza una función de transferencia para analizar la señal y determinar las características del ruido (fase, frecuencia, amplitud). Basándose en esta información, el controlador genera una señal de control que modifica la señal original, cancelando las componentes no deseadas del ruido. La señal modificada se convierte de nuevo en analógica y se envía a los altavoces, generando una onda sonora que interfiere destructivamente con la onda de ruido original, eliminándola.

Los algoritmos deben de presentar dos fases:

En primer lugar, un proceso de filtrado de la señal de entrada mediante el cual se obtenga una señal de salida generada por los altavoces. Por otro lado, también debe de existir un proceso adaptativo, que proporcione al sistema la capacidad de poder adaptarse a distintas situaciones acústicas y de esta forma el error entre la señal de entrada y la de salida sea mínimo.

Ambas fases se explicaran en profundidad en el siguiente apartado 2.1.4.

2.1.4 FILTROS ADAPTATIVOS

2.1.4.1 FILTROS FIR (*FINITE IMPULSE RESPONSE*)

Este tipo de filtros su característica más destacable es que podemos diseñarlos con una fase perfectamente lineal, esto es de vital importancia para que no exista retardo entre las dos señales. Además, se trata de un filtro no recursivo, es decir, la ecuación de salida depende de las señales de entrada actuales y pasadas, pero no de las salidas anteriores, esto es debido a que no existe una realimentación entre la salida y la entrada. No existen polos en el semiplano estrictamente negativo, por lo que por el criterio de estabilidad de Nyquist no afectan a la estabilidad del sistema, siendo siempre estables.

El criterio de Nyquist es un método gráfico analítico que determina la estabilidad de un sistema en lazo cerrado, al investigar las propiedades de la traza de Nyquist en el dominio de la frecuencia de la función de transferencia del lazo. Un sistema en lazo cerrado es estable si y solo si la curva de Nyquist no encierra el punto (-1,0) (Martín, 2009). Por lo tanto, si no hay polos en el semiplano negativo, la curva de Nyquist no encerrará ese punto.

La posición de los polos de un filtro determina su respuesta en el tiempo. Si un polo se encuentra dentro del círculo unitario, la respuesta del filtro crecerá exponencialmente con el tiempo, lo que indica una condición inestable. Al no tener polos en el semiplano izquierdo estricto, los filtros FIR garantizan que su respuesta sea acotada y, por lo tanto, estables. (Mosquera, 2014)

La ecuación de los filtros FIR, denominada ecuación en diferencias es de la siguiente forma:

$$y(n) = \sum_{k=0}^M b * x(n - k)$$

El parámetro M es el número de coeficientes del filtro. Cuanto mayor sea este número, mejor será la aproximación. Este parámetro indica el retardo máximo que se le puede aplicar al filtro. Este retardo se obtiene mediante el cociente entre el parámetro M y la frecuencia de sonido que llega.

El diagrama de bloques de un filtro FIR se muestra en la Ilustración 4 Diagrama de bloques de filtro FIR

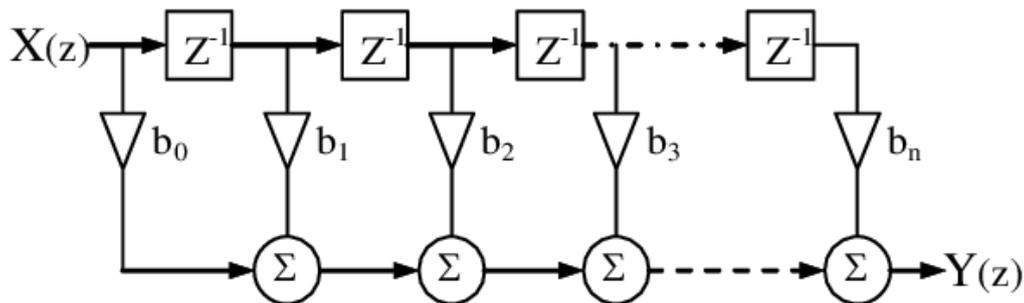


Ilustración 4 Diagrama de bloques de filtro FIR (Chávez, 2005)

Donde:

- $X(z)$ es la secuencia de entrada
- $b(n)$: valor de los coeficientes del filtro, representan la amplificación (son los coeficientes b de la ecuación que se mostraba previamente)
- Los bloques Z^{-1} son unidades de retardo, entregando un dato de salida un período de muestreo después de haberlo recibido en la entrada.
- Los sumatorios representan la suma de señales. En la práctica, estos filtros suelen implementarse mediante un acumulador que se inicializa en 0 y va sumando todas las multiplicaciones de cada muestra en la memoria
- $Y(z)$: secuencia de salida filtrada

2.1.4.2 IRR (INFINITE IMPULSE RESPONSE)

Un filtro de respuesta infinita al impulso (IIR, por sus siglas en inglés) es un tipo de filtro digital utilizado en el procesamiento de señales para modificar una señal de entrada y obtener una señal de salida con características específicas.

La característica principal de los filtros IIR es que su respuesta al impulso se mantiene indefinidamente en el tiempo. Esto permite que los filtros IIR tengan una respuesta en frecuencia más compleja en comparación con los filtros FIR (Finite Impulse Response) y pueden ser diseñados con un menor número de coeficientes. Esto es interesante ya que menos coeficientes significan menos operaciones de multiplicación y suma, y esto lo hace energéticamente más eficiente. El funcionamiento de un filtro IIR se basa en la

retroalimentación de su salida a su entrada, lo que permite que la señal de entrada sea filtrada repetidamente durante su procesamiento. Esto es posible gracias a la presencia de polos y ceros en la función de transferencia del filtro, que determinan su respuesta en frecuencia y su comportamiento en el dominio del tiempo. (POLARIDAD.ES). Aplicando el criterio de estabilidad de Nyquist, este filtro es inestable y su respuesta crecerá exponencialmente en el tiempo.

El diagrama de bloques de un filtro FIR se muestra en la Ilustración 4 Diagrama de bloques de filtro FIR

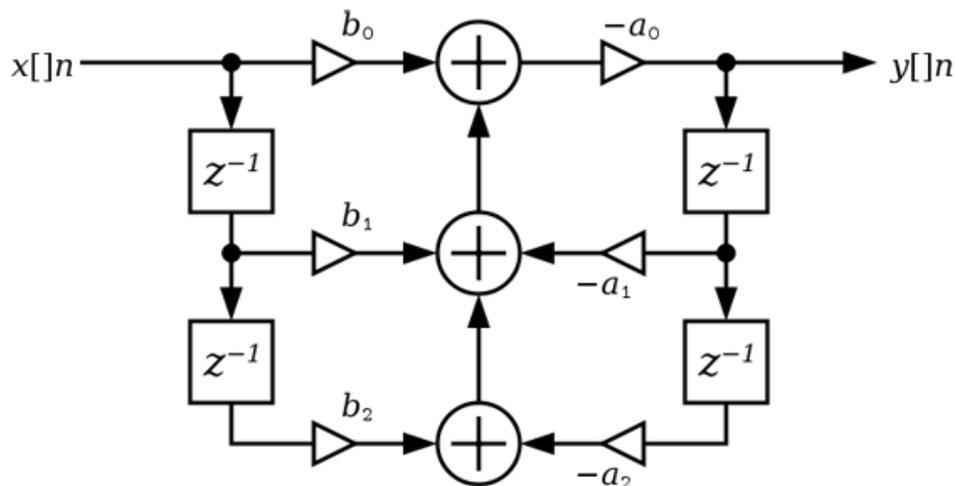


Ilustración 5 Diagrama de bloques de filtro IIR (POLARIDAD.ES)

Nótese que como podemos ver en este segundo diagrama de bloques, el filtro IIR se puede asemejar a dos filtros FIR conectados, estando el segundo invertido en forma de espejo. Esto en lenguaje matemático no es más que el cociente de dos funciones de transferencia FIR.

El filtro tipo IIR suele utilizarse para eliminar realimentaciones acústicas. Sin embargo, debido a la realimentación que tienen, su número de polos en el plano estrictamente negativo no es nulo, y por lo tanto por el criterio de Nyquist se puede afirmar que son potencialmente inestables.

A pesar de que la capacidad de eliminar realimentaciones acústicas es una ventaja significativa para sistemas de cancelación activa de ruido, los filtros IIR pueden presentar problemas de estabilidad. Por esta razón, se suele preferir el uso de otros métodos para manejar la realimentación, como el uso de micrófonos y altavoces o acelerómetros, en lugar de confiar únicamente en los filtros IIR. (Neves, 2014)

2.1.4.3 CONCLUSIONES SOBRE LOS FILTROS FIR E IIR

Flexibilidad: La función de transferencia de los filtros FIR presenta únicamente polos en el origen, sin embargo, la función de transferencia de los filtros IIR presentan polos y ceros sobre todo el plano Z lo que garantiza una mayor flexibilidad en su diseño.

Estabilidad: Debido a que únicamente presentan polos en el origen, los filtros FIR son más estables, sin embargo, debido a la realimentación los filtros IIR presentan polos en el semiplano estrictamente negativo siendo por tanto inestables.

Número de coeficientes: El número de coeficientes que presente un filtro está directamente relacionado con el tiempo de procesado, es decir a mayor número de coeficientes que presente un control, mayor tiempo de ejecución tendrá.

Cualquier función de transferencia puede lograrse con un filtro FIR, ahora bien, se deben de añadir un amplio número de coeficientes. Por ello la principal ventaja de los filtros IIR frente a los filtros FIR es que los primeros requieren de menor número de coeficientes para realizar operaciones similares y por lo tanto son más rápidos.

Por otro lado, la desventaja que presentan los filtros IIR frente a los filtros FIR es que la respuesta de fase no es lineal, esto quiere decir que las respuestas que se obtienen se encuentran distorsionada la fase y son inestables. El tipo de filtro que se instalará para en el sistema de cancelación activa de sonido será un tipo de filtro FIR debido a la estabilidad que presentan. El sistema sabe que sonidos filtrar gracias al análisis espectral que se realiza mediante los algoritmos. Los ruidos de la carretera tienen espectros de frecuencia amplios sin picos a diferencia de la música de esta forma se sabe que señales filtrar, aquellos con frecuencias amplias y sin picos.

Ahora bien, sabemos la estructura de ambos filtros, pero aún queda un problema por resolver. No se conocen el valor que deben obtener los coeficientes de ambos filtros para de esta forma minimizar dinámicamente la señal en el micrófono de error, es decir, no se conoce qué valor dar para cancelar al máximo la señal acústica entrante en el automóvil. Además, estamos sujetos a continuas variaciones de temperatura, frecuencia... y los filtros que implementemos deben de ser capaces de adaptarse a las distintas variaciones que surgen en el entorno acústico. Ante este problema se puede solucionar mediante los algoritmos adaptativos de los que podemos diferenciar dos tipos.

2.1.5 ALGORITMOS ADAPTATIVOS

Como se ha mencionado anteriormente, los algoritmos adaptativos son los encargados de ajustar los valores de los coeficientes de los filtros ante variaciones del entorno acústico.

Su funcionamiento a grandes rasgos es sencillo. El algoritmo a partir de una señal de entrada x y la señal deseada d , realizará una serie de cálculos e irá dando distintos valores precisos para cada uno de los coeficientes, los cuales se irán actualizando para que de esta forma la señal de salida y sea lo más parecida a la señal deseada, es decir la señal atenuada.

Su misión será minimizar al máximo posible el error $e(k)$. Donde $e(k) = x(k) - y(k)$.

- $x(k)$: señal de entrada
- $y(k)$: señal de salida ya filtrada

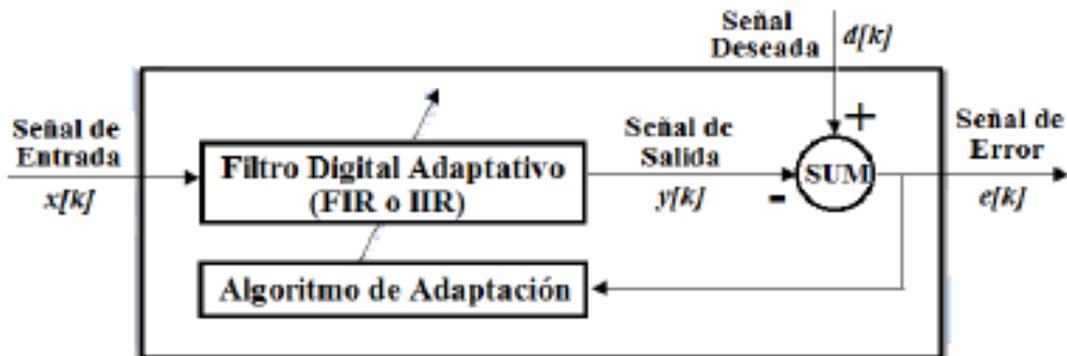


Ilustración 6 Diagrama de bloques algoritmo adaptativo

(L., Algoritmos LMS de filtrado adaptativo, 2014)

2.1.5.1 ALGORITMO LMS

El ruido en un entorno real es dinámico y variable, las condiciones como la velocidad o la temperatura a las que se ven expuestos los filtros varían. Los algoritmos LMS permiten que el filtro se ajuste continuamente a estas variaciones, garantizando una cancelación efectiva del ruido en todo momento. Busca encontrar los coeficientes del filtro que permiten obtener el valor mínimo esperado del cuadrado de la señal del error. El algoritmo se adaptará en base al gradiente de la tendencia estadística relacionando la señal a filtrar con la señal de ruido. Se trata de un sistema similar al ECM (Error Cuadrático medio) empleado en varios campos de la estadística y de la ingeniería.

El algoritmo actualiza los coeficientes en función de los cambios del gradiente. El gradiente se define como la derivada del cuadrado del error con respecto a cada uno de los coeficientes.

El análisis matemático para obtener el error mínimo es el siguiente:

Por un lado, tenemos la ecuación del error, que no es más que la diferencia entre la señal deseada y la señal de salida.

$$e(n) = x(n) - y(n)$$

La señal de salida para un filtro de tipo FIR es:

$$y(n) = \sum_{k=0}^M b * x(n - k)$$

Por lo tanto, el error será igual a:

$$e(n) = x(n) - \sum_{k=0}^M b * x(n - k)$$

El error cuadrático instantáneo no es más que el error al cuadrado. Por lo tanto:

$$e(n)^2 = [x(n) - \sum_{k=0}^M b * x(n - k)]^2$$

A continuación, se emplea el método de máxima pendiente, (Metodo de La Maxima Pendiente, 2021) método matemático que nos permite obtener el valor óptimo. De esta forma obtenemos los distintos coeficientes para b.

En este método tomaremos como referencia el valor actual de nuestro coeficiente. El valor del coeficiente siguiente será el valor del coeficiente actual menos el gradiente del error cuadrático instantáneo multiplicado por una constante μ . (Lavinia, 2024)

$$W(n + 1) = W(n) + \mu E(n)x(n)$$

Este parámetro μ no es más que un regulador de velocidad en la adaptación del sistema. Hay que tener especial cuidado a la hora de proporcionar un valor para μ , ya que este además de determinar la velocidad de adaptación de los coeficientes también determina el error en régimen permanente. Por ello para que el funcionamiento del algoritmo adaptativo sea correcto este parámetro debe estar comprendido el siguiente umbral. $0 < \mu < 2 / \sum x(i)^2$. (L., Algoritmos LMS , 2014)

Este algoritmo se aplica sobre todo para los filtros FIR ya que los filtros IIR presentan dos problemas:

En primer lugar, como ya ha sido comentado la inestabilidad que proporciona. Por otro lado, este sistema tiene varios mínimos lo cual es un gran problema para el algoritmo LMS ya que su función principal es básicamente encontrar mínimos, por ello, podría obtener un mínimo que se trate de un mínimo local y no de un mínimo absoluto. (L., Algoritmos LMS , 2014)

2.1.5.2 ALGORITMOS GENÉTICOS

EL algoritmo adaptativo LMS busca los coeficientes óptimos para los filtros FIR e IIR mediante el método matemático de máxima pendiente. Además, también tienen que estimar la función de transferencia formada por los coeficientes que se buscan optimizar para conseguir una interferencia destructiva de la onda. En ocasiones se dan errores de fase de más de $\pi/2$ rad, estos errores que suelen aparecer con bastante frecuencia hacen que el algoritmo adaptativo sea inestable y sea impreciso. Este tipo de errores se dan cuando la función de transferencia de la señal del audio entre la salida y el error es no lineal, en estas ocasiones no se puede emplear algoritmos adaptativos y se deben de emplear una herramienta distinta.

Los algoritmos genéticos sí que son capaces de adaptar los coeficientes de los filtros para funciones de transferencia no lineales. Además, son más sencillos que los algoritmos adaptativos ya que no necesitan estimar funciones de transferencia para posteriormente cancelarlas, por lo tanto, supone un gran ahorro de cálculo ya que no necesitan analizar el error y todo el proceso matemático descrito para los algoritmos adaptativos.

El procedimiento de búsqueda de los coeficientes de este tipo de algoritmos se fundamenta en el comportamiento evolutivo de los seres vivos. Es una especie de simulación del sistema de selección natural descrito por Darwin. Lo que hace es combinar la supervivencia del más apto entre distintas secuencias en un determinado intervalo aleatorizado.

Para encontrar el coeficiente idóneo se tomará un conjunto de posibles coeficientes de forma aleatoria estableciéndose de esta forma una población y se someterá a las siguientes fases: generación de una población aleatoria y selección, reproducción y mutación.

1. Generación de una población aleatoria y selección

En primer lugar, el algoritmo genético dará valores aleatorios a la población. A continuación, estos valores aleatorios se someten al proceso de selección, en los que se tomaran aquellos valores más aptos, aquellos con menor error cuadrático medio.

2. Reproducción o cruzamiento

La siguiente fase es la reproducción donde entre los valores seleccionados, se forman parejas de forma aleatoria para la creación de nuevos valores. Por cada pareja se crean dos valores nuevos los cuales tienen información de cada miembro de la pareja.

3. Mutación

Después del proceso de reproducción viene el proceso de mutación de los nuevos valores, alterándose de forma aleatoria la información que contienen de forma que se generen todos los valores posibles de búsqueda y se obtenga el mínimo global.

Como se puede observar en la Ilustración 7 Algoritmos genéticos el ciclo de selección, cruzamiento y mutación se repite múltiples veces hasta que los coeficientes convergen, es decir, los valores obtenidos comienzan a ser muy similares entre sí. La convergencia indica que el algoritmo ha encontrado una solución óptima o cercana a la óptima para los coeficientes de los filtros, adaptándose eficazmente a funciones de transferencia no lineales sin la complejidad de los métodos adaptativos tradicionales. (Batista, 2009)

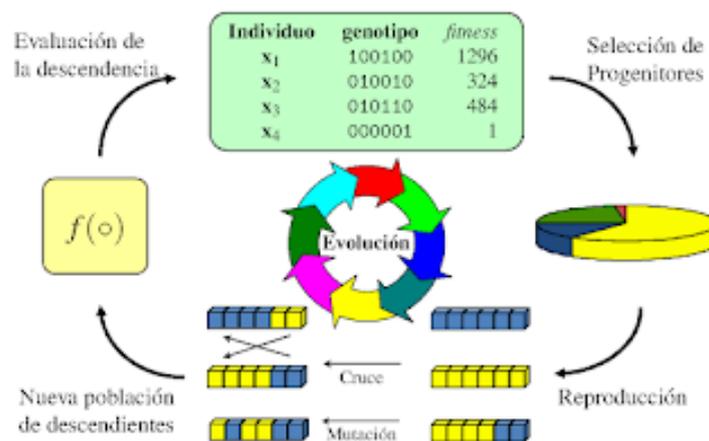


Ilustración 7 Algoritmos genéticos (Batista, 2009)

2.1.6 LA IA EN LA CANCELACIÓN ACTIVA DE SONIDO

Como se ha explicado anteriormente, el sistema de ANC es un sistema de cancelación de ruido capaz de captar señales por unos micrófonos y de reproducir unas señales opuestas por unos altavoces y de esta forma cancelar la onda.



Ilustración 8 IA en la cancelación activa del sonido (Alonso, 2021)

La IA también podría ser capaz de realizar esta misma función, simplemente necesita una base de datos para saber que ruidos debe suprimir y como todos los algoritmos de IA, mediante esta base de datos se realiza un Machine Learning. El Machine Learning, o aprendizaje automático, es un campo de la inteligencia artificial que permite a las máquinas aprender y mejorar automáticamente a partir de la experiencia sin ser programadas explícitamente para ello.

Esto nos da un amplio abanico de posibilidades, además de poder realizar la función del ANC y poder cancelar ruidos, también podemos atenuar otro tipo de ruidos que son molestos y simplemente enviar un mensaje a la pantalla del vehículo. Como puede ser el ruido de una sirena el cual sería detectado, reconocido e interpretado por la IA y posteriormente se le comunicaría al conductor.

2.1.7 EMPRESAS DEL SECTOR

Hay varias empresas en la carrera por encontrar la mejor forma para conseguir insonorizar el interior de un automóvil. Estas empresas aportan soluciones tanto de software como de hardware, listas para poder ser aplicadas al automóvil. Algunas de ellas son:

- **Bose Corporation:** La empresa estadounidense trabaja en desarrollar un sistema de cancelación de ruido óptimo en automóviles. Han desarrollado un sistema de cancelación de ruido específica para asientos, de esta forma se logra una experiencia de conducción más cómoda ya que se reduce considerablemente el ruido transmitido a través de los asientos. Para poder anticiparse al sonido y poder cancelarlo lo más rápido posible Bose Corporation utiliza sensores como acelerómetros para mejora la precisión y efectividad. (Garcia A. , 2023)
- **Bosch** por su parte también ha realizado múltiples avances en el campo de la cancelación activa de ruido en automóviles. Además de implementar una tecnología de ANC adaptativa e integrar sensores para mejorar la eficacia y la precisión de la cancelación de ruido, Bosch también ha trabajado en la cancelación de ruido en múltiples zonas del interior del vehículo. El sistema genera ondas de sonidos inversas y muy específicas en el interior del vehículo para cancelar el ruido en zonas específicas del interior del automóvil. Por otro lado, Bosch trabaja también en que su sistema ANC sea lo más energéticamente eficiente posible, se trata de un aspecto muy importante para el futuro próximo cara al desarrollo de los automóviles eléctricos.
- **Faurecia** ha trabajado en el desarrollo de la cancelación de ruidos generados por componentes eléctricos y electromecánicos que emiten frecuencias específicas como el motor eléctrico o los inversores.
- **Silentium**, esta última empresa con sede en Israel ya ha implementado su sistema de cancelación activa de ruido en algunos vehículos como el *Jaguar F-Pace* y *XF* o en modelos de la marca Range Rover como el modelo *Range Rover Velar*. El sistema Silentium se encarga de cancelar todos aquellos sonidos no deseados de la carretera fruto de la fricción de los neumáticos con el asfalto. Consiguen eliminar el 90% del

ruido mediante un sistema de acelerómetros y micrófonos. Gracias a un sistema de micrófonos y altavoces son capaces de crear burbujas alrededor de cada pasajero creando burbujas de sonido. Actualmente están desarrollando una tecnología para que cada pasajero, mediante su burbuja pueda escuchar su propio contenido sin que el resto de pasajeros lo escuchen. (Ltd, 2021)

- Hyundai Motor Group ha desarrollado un sistema de control de ruido denominado RANC (Road Active Noise Control). Un diagrama explicativo de este sistema se muestra en la **¡Error! No se encuentra el origen de la referencia.** La novedad de este sistema es que es capaz de analizar varios tipos de ruido en tiempo real y producir ondas sonoras invertidas y de esta forma lograr cancelarlo. Para ello utiliza un sensor de aceleración el cual calcula la vibración de la carretera y posteriormente analiza el ruido en breves instantes de tiempo (unos 0.002 segundos), hay que tener en cuenta, que el ruido de la carretera tarda unos 0,009 segundos en llegar a los pasajeros, por lo que los resultados obtenidos con esta tecnología son ampliamente satisfactorios. Con esta tecnología Hyundai espera ser líder tecnologías acústicas para proporcionar una mayor comodidad en el interior de las cabinas de sus vehículos. (Garcia G. , 2019).

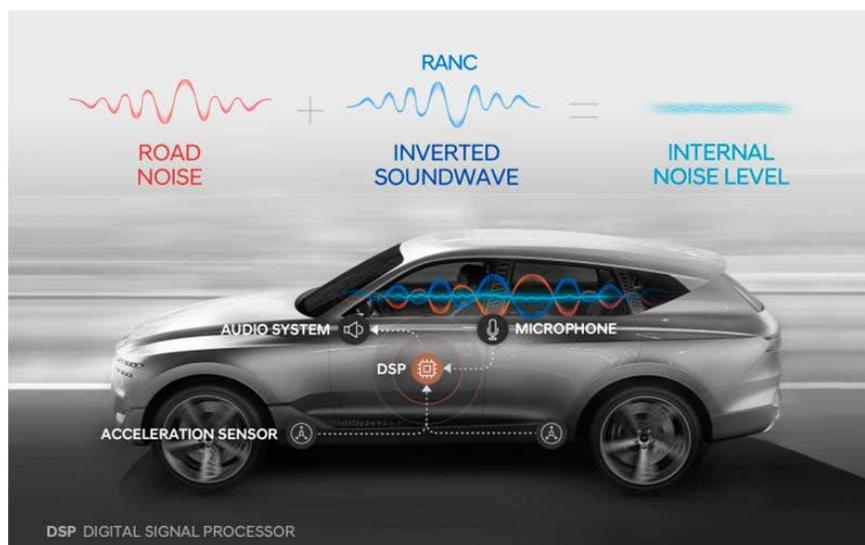


Ilustración 9 Sistema cancelación de sonido Hyundai

Bien es cierto que a través del sistema denominado RANC se logra una cancelación del ruido blanco de la carretera. El ruido blanco es un tipo de sonido que contiene todas las frecuencias audibles en igual medida. Se caracteriza por ser un sonido constante.

2.1.8 PROXIMOS AVANCES

Recientemente la industria automovilística está trabajando en aislar la experiencia auditiva de cada pasajero que viaje en el vehículo. Marcas como Kia o SEAT están trabajando en ello e incluso han llegado a presentar algún modelo

La propuesta de Kia consiste en dirigir la música hacia la persona que desea oírla, de tal forma que el resto de los pasajeros no puedan percibir ese sonido. SEAT propone algo similar junto con con la empresa Noveto System, quién cuenta entre sus inventos con el altavoz SoundBeamer. (Romero, El algoritmo que ayudará a viajar en coche con niños: burbujas de sonido delimitadas a cada asiento, 2020) pero ampliando y disminuyendo las ondas de sonido, para que al encontrarse se anulen.

SEAT para mejorar su sistema utilizo un dispositivo para colocar las ondas justo a la entrada de los oídos y así evitar que nadie más pueda escucharlo. Previamente se ha analizado donde se encuentra la cabeza con un sistema de detección 3D para poder percibir su posición y una vez el sistema sabe dónde se encuentra las orejas, crea unas pequeñas bolsas donde la frecuencia disminuye hasta convertirse en ondas audibles y de esta forma solo lo puede escuchar persona.



Ilustración 10 Micrófonos colocados en un coche de prueba -Universidad de Le Mans (Romero, El algoritmo que ayudará a viajar en coche con niños: burbujas de sonido delimitadas a cada asiento, 2022)

A pesar de ser una idea interesante para poder ser aplicadas y crear zonas de sonido personales, no terminan de dar resultados. Existen varios aspectos que provocan cambios en el funcionamiento como por ejemplo el cambio de temperatura, la cantidad de pasajeros que hay en el vehículo o a que frecuencia bajar las ondas para poder crear esas bolsas.

Por todo esto se puede concluir que el sistema no es válido para poder lanzarlo todavía. Otra opción para adaptar el vehículo con zonas de sonido personales es implementar en los vehículos una idea similar a la que sugiere la empresa alemana RECALM. La idea es crear burbujas de sonido mediante los reposacabezas que se muestran en Ilustración 11 Recalm anchor infotainment-system . Mediante micrófonos captarían la señal. (RECALM, 2024)



Ilustración 11 Recalm anchor infotainment-system (RECALM, 2024)

Sin embargo, se trata de una idea complicada de llevar a cabo debido a distintos aspectos como por ejemplo la posición de la cabeza, los distintos sensores que hay que utilizar para desviar el sonido, por lo tanto se trata de una idea que sigue en desarrollo. Se ha llegado a admitir sonidos entre 100-1000 Hz, frecuencias que equivalen a sonidos graves o conversaciones a bajo volumen. La investigación todavía continua para tratar de aumentar el ancho de banda, por lo que se trata de un proyecto.

2.2 RECONOCIMIENTO DE LA VOZ

El reconocimiento del sonido, una rama avanzada del procesamiento de señales ha revolucionado la forma en que interactuamos con la tecnología en nuestra vida diaria. Desde los inicios con la invención de los primeros dispositivos de grabación y reproducción de audio, hasta los sofisticados sistemas actuales que pueden entender y responder a comandos de voz con una precisión sorprendente. El reconocimiento del habla ha tenido un impacto significativo en diversos campos, incluyendo la asistencia personal, la seguridad, la accesibilidad y el entretenimiento. Esta tecnología no solo facilita una comunicación más natural y eficiente entre humanos y máquinas, sino que también abre nuevas posibilidades para personas con discapacidades, permitiéndoles interactuar con el entorno. El reconocimiento del sonido no es solo una herramienta de conveniencia; representa un paso crucial hacia un futuro donde las máquinas entienden y responden al lenguaje humano de manera casi indistinguible de una interacción entre personas.

El reconocimiento de voz se encuentra ya muy presente en la industria del automóvil. En muchos vehículos tienen la posibilidad de poder comunicarte con la computadora a través de la voz, activando esta opción con un botón el cual normalmente viene colocado en el volante. El sistema que se trata de desarrollar va mucho más allá. Con este sistema se busca en primer lugar eliminar todo tipo de botones además de dar la posibilidad de comunicarse con la computadora (función no disruptiva ni innovadora ya que se encuentra muy desarrollada). Por otro lado, busca el poder hacer otras funciones como abrir el coche o al comunicarte con él que sea capaz de identificar quien es el conductor para configurar el puesto de conducción según quien corresponda, con unas configuraciones previamente preinstaladas: posición del asiento, posición de los espejos retrovisores, o la lista de música favorita, etc... La dificultad en estas funciones son aprendizaje previo del programa necesario para que funcione.

2.2.1 FUNCIONAMIENTO GENERAL

La posibilidad de inventar interfaces para crear una comunicación fluida entre humano y máquina es de gran interés. Esta posibilidad de que simplemente con nuestra voz se puedan dar órdenes a una máquina ofrece múltiples ventajas respecto a otros métodos, como con un teclado o una pantalla táctil. Algunas de estas ventajas son una comunicación más rápida, ofrece menos distracciones, es más agradable y permite liberar nuestras manos entre otras. (La evolución de las interfaces de comunicación, 2018)

Se trata de un tema que presenta gran complejidad. La comunicación que resulta tan sencilla entre personas, no lo es tanto cuando se trata de comunicarnos con una computadora. Ya no tanto la dificultad que presenta representar palabras que no es tanta, sino la dificultad que es que una máquina sea capaz de reconocer el habla. A partir de la señal de voz, un proceso de clasificación, basado en reconocimiento de patrones asociados a diferentes unidades lingüísticas (palabras, fonemas, sílabas, etc.), permite a una interfaz de comunicaciones extraer de la base de datos la información solicitada por el usuario.

El ASR (por sus siglas en inglés Automatic Speech Recognition) utiliza algoritmos avanzados de procesamiento de señales y modelos de lenguaje para analizar y comprender el habla humana. El proceso de reconocimiento automático de voz se puede dividir en varias etapas. (Rufiner, 2004). En la Ilustración 12 se presenta un diagrama con las diferentes etapas que comprende este sistema.

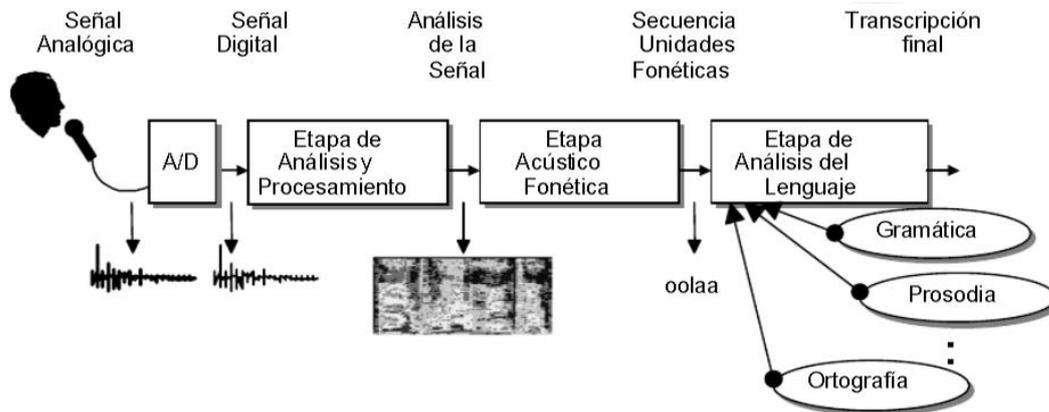


Ilustración 12 Etapas de un sistema de ASR (Rufiner, 2004)

La primera etapa es la captura del habla, donde se utiliza un micrófono o dispositivo de entrada de audio para recoger la señal acústica de la voz humana. Esta señal se convierte en una forma de onda eléctrica que luego se procesará. Esta parte del proceso se ha explicado anteriormente en el apartado 2.1.2.

En la etapa siguiente, la señal acústica se preprocesa para eliminar el ruido y mejorar la calidad del sonido. Luego, se utilizan técnicas de análisis de señales como la transformada de Fourier para extraer características relevantes del habla, como frecuencias, duraciones y amplitudes. En esta etapa también se busca reducir ecos. Un eco es una repetición de un sonido original, causada por la reflexión de las ondas sonoras en superficies. La presencia de ecos en una señal de audio puede degradar significativamente la calidad de la señal y dificultar la comprensión del habla.

Una vez extraídas las características, se procede a la etapa de reconocimiento, donde se comparan las características extraídas con modelos de lenguaje previamente entrenados. Estos modelos incluyen diccionarios de palabras, gramáticas y probabilidades de transiciones entre palabras.

Los algoritmos de reconocimiento automático de voz utilizan métodos estadísticos y de aprendizaje automático para encontrar la mejor coincidencia entre las características extraídas y los modelos de lenguaje. A medida que se realiza esta comparación, se genera una secuencia de palabras que representa la transcripción o la interpretación del habla.

Finalmente, la secuencia de palabras generada se puede utilizar para diversas aplicaciones, como transcripción de voz a texto, comandos de voz en sistemas interactivos, asistentes virtuales, control de voz en dispositivos electrónicos, entre otros.

2.2.2 DIGITALIZACIÓN DE LA VOZ

2.2.2.1 LA TRANSFORMADA DE FOURIER

El primer paso consiste en segmentar la señal en marcos o ventanas, que consiste en dividir la señal en pequeñísimos intervalos de tiempo constantes. El propósito de esto es modelar pequeñas porciones de la señal que sean estables en el tiempo, con el fin de generar un vector. Dicho vector se refiere a la transformación de una señal en el dominio de la frecuencia mediante una función logarítmica, obteniéndose un conjunto de valores numéricos procedentes de una señal de audio y que permite analizarla de manera profunda. El propósito de esto es modelar una señal de audio en una señal en el dominio de la frecuencia para posteriormente poder analizarla.

El análisis de Fourier se basa en el análisis de una señal periódica $x(n)=x(n+N)$ en una suma de exponenciales complejas. Este proceso permite representar la señal original como una serie de componentes sinusoidales. La senoide con la menor frecuencia en este crecimiento se conoce como la frecuencia fundamental. Los demás senoides tienen frecuencias que son múltiplos de la fundamental y se denominan armónico. (Ibarrola, 2020)

Para cada frecuencia, la magnitud (valor absoluto) del valor complejo representa la amplitud de una senoide compleja constituyente con esa frecuencia, y el argumento del valor complejo representa el desplazamiento de fase de esa senoide compleja. Si una frecuencia no está presente, la transformada tiene un valor de 0 para esa frecuencia. Este comportamiento se observa en la Ilustración 13. (Academialab, 2024)

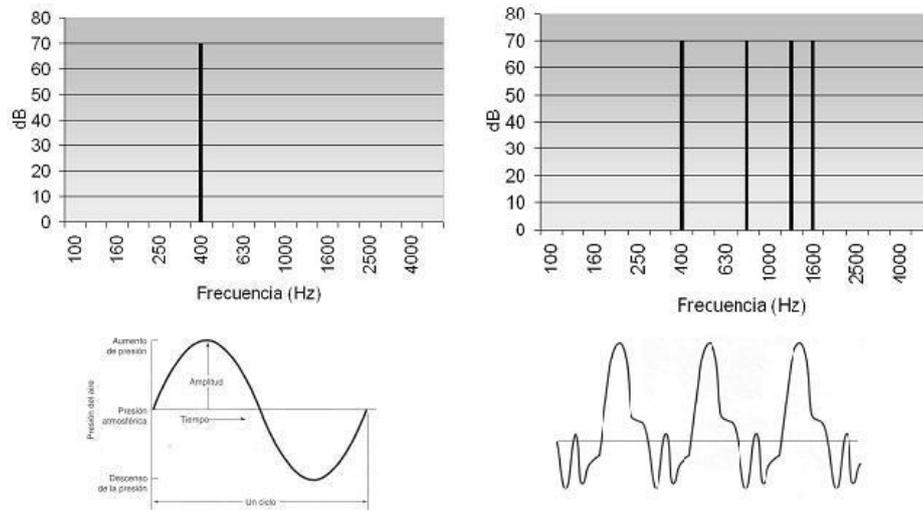


Ilustración 13 Transformada de Fourier

(Academialab, 2024)

Se calcula la magnitud del espectro, que representa la energía en cada frecuencia, aplicando la siguiente fórmula:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}$$

Donde:

- $F(\omega)$ es la función con dominio en frecuencia
- $f(t)$ es la función periódica
- ω es la frecuencia angular

Esta magnitud del espectro proporciona información sobre la distribución de energía en el espectro de frecuencias del segmento de audio. A partir de la magnitud del espectro, se extraen características acústicas relevantes para el reconocimiento del habla

2.2.2.2 BANCO DE FILTROS MEL

También conocidos como Mel-Frequency Cepstral Coefficients (MFCCs). Su función es representar la envolvente espectral del habla en un conjunto de bandas de frecuencia. Se define un conjunto de filtros triangulares, cada uno con una banda de frecuencia específica.

La forma de estos filtros se basa en la percepción auditiva humana perceptualmente relevantes, lo que facilita la extracción de información útil para la identificación de fonemas y palabras.

Para transformar los Hz a Mels se emplea la siguiente fórmula:

$$\phi = 2595 \log_1 \log_{10} \left(\frac{f}{700} + 1 \right)$$

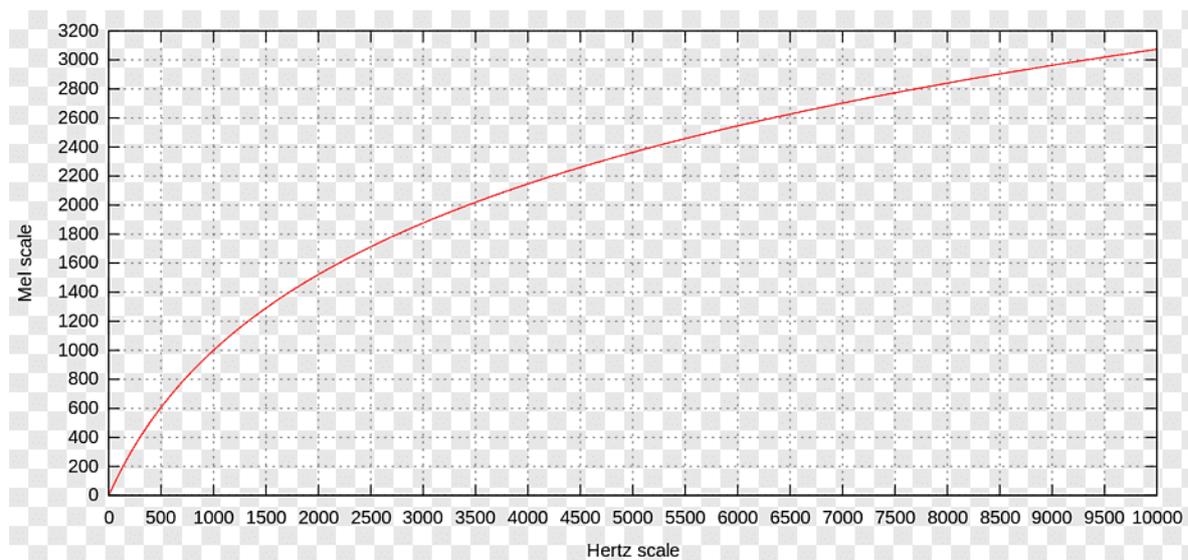


Ilustración 14 Escala de Mel

(PNGWING)

Aunque en Hertz, las frecuencias centrales están repartidas logarítmicamente, en la escala de Mel las frecuencias centrales están linealmente distribuidas.

$$\Delta\phi = \frac{\phi_{max} - \phi_{min}}{M + 1}$$

donde ϕ_{max} y ϕ_{min} son los límites del rango de frecuencias de la señal bajo análisis correspondientes a f_{max} y f_{min} respectivamente y M es el número de filtros que conforman el banco de filtros. Mediante la fórmula se calcula el incremento o separación entre frecuencias centrales adyacentes en la escala Mel. La división entre $M+1$ asegura que las frecuencias centrales estén distribuidas uniformemente la frecuencia máxima y la mínima. El objetivo de obtener $\Delta\phi$ es dividir la señal de audio en segmentos o ventanas para poder considerar dicho tramo estacionario y poder analizarlo.

Las frecuencias centrales de Mel se obtienen mediante la siguiente fórmula:

$$\Delta\phi_c(m) = m\Delta\phi \quad m=1,2, 3., M$$

Para obtener las frecuencias centrales se aplica la siguiente fórmula

$$f_c(m) = 700(10^{\frac{\phi_c(m)}{2595}} - 1)$$

Los filtros de Mel se basan en la percepción auditiva humana, lo que significa que capturan información que es relevante para el sistema auditivo humano y que es útil para el reconocimiento del habla.

Para implementar un sistema de reconocimiento de voz en base a MFCC se propone utilizar quince filtros triangulares de Mel, es decir $M = 15$, una frecuencia inferior f_{min} de 80 Hertz puesto que es la frecuencia mínima que un tracto vocal puede emitir, y una frecuencia superior f_{max} de 4000 Hertz (Academialab, 2024)

Una vez se han obtenido los MFCCs, se almacenan en un vector con su etiqueta correspondiente (en nuestro caso la etiqueta es el nombre del vocalista al que pertenece dicho MFCC)

2.2.2.3 RECONOCIMIENTO DE LA VOZ MEDIANTE REDES NEURONALES

Las redes neuronales son modelos matemáticos que se inspiran en el funcionamiento y la interacción de las neuronas en el cerebro humano. Aunque inicialmente estudiadas por biólogos, las redes neuronales se han convertido en un campo interdisciplinario que abarca ciencias de la computación, ingeniería eléctrica, matemáticas, física y psicología, entre otros.

El principal objetivo de las redes neuronales es crear sistemas que muestren un comportamiento inteligente, lo que implica la capacidad de aprender a realizar tareas específicas. La neurona, unidad básica de una red neuronal, es una célula con características propias que le permiten comunicarse con otras neuronas a través de sinapsis, proceso que realizan las neuronas humanas para realizar conexiones entre ellas (Labra, 2018)

Las redes neuronales son hoy día parte fundamental de la IA y en concreto del Deep Learning y como veremos a continuación, al contrario de la creencia popular, no son conceptos nuevos. El primer algoritmo que presentaba una red neuronal simple se llamó Perceptrón, creado por Frank Rosenblatt en 1958. (Ramírez, 2018)

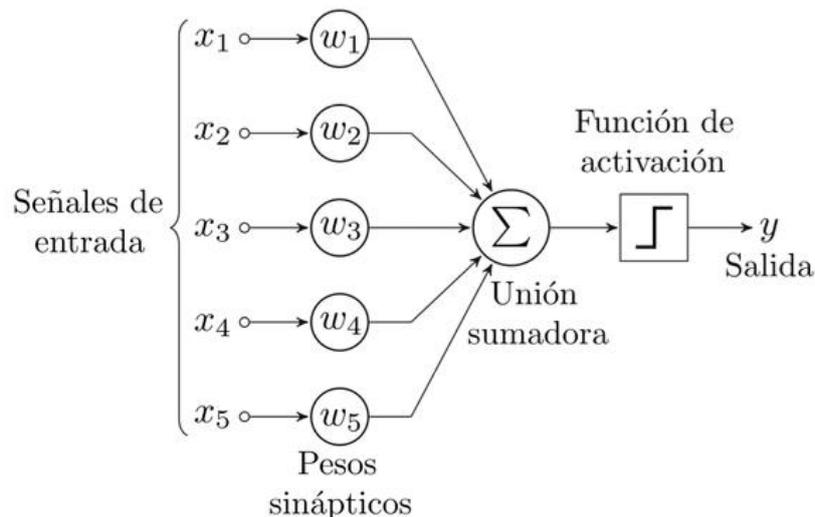


Ilustración 15 Esquema de una red neuronal

(Ramírez, 2018)

Tal y como se muestra en la Ilustración 16, la primera capa de entrada está compuesta por todos los MFCC (Mel-Frequency Cepstral Coefficients) obtenidos durante el preprocesamiento del audio, que, como se mencionó anteriormente, se encuentran almacenados en un vector. Por ello, el primer paso es transformar ese vector de MFCCs en una capa de entrada, tal como se muestra en la figura. Luego, se agregaron tres capas ocultas con 512, 256 y 64 neuronas, respectivamente, hasta llegar a la capa de salida que tiene 4 neuronas, donde se identificarían 4 voces, que forman la capa de salida como se ejemplifica en la siguiente ilustración (Villanueva, 2020)

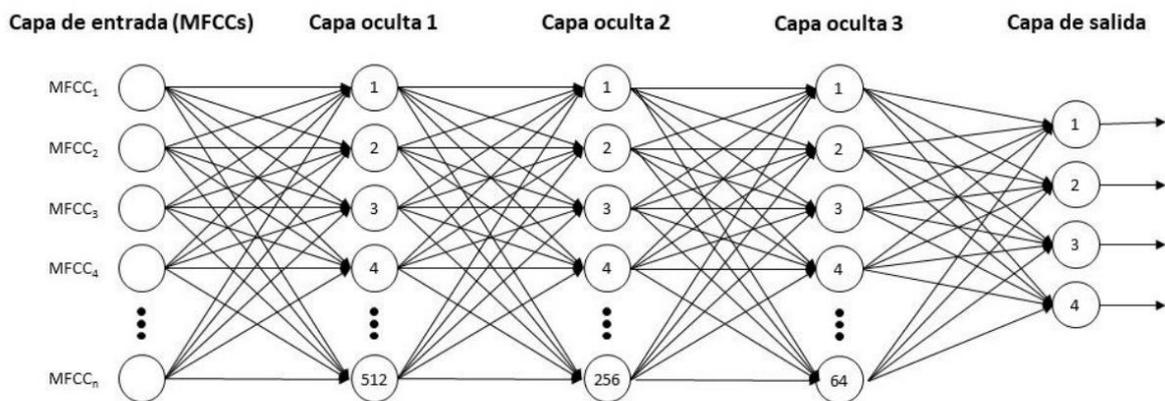


Ilustración 16 Estructura red neuronal

(Villanueva, 2020)

2.2.3 EMPRESAS DEL SECTOR – ESTUDIO DEL MERCADO

- **Kardome:** Su sistema de reconocimiento cuenta con un sistema de reducción de ruido, comunicación con la computadora a través de la voz, IA de voz, múltiples hablantes, servicio offline y servicio en la nube. Su tecnología avanzada permite distinguir entre múltiples fuentes de sonido, lo que es esencial para aplicaciones en entornos ruidosos y con múltiples hablantes. (LTD, 2023)
- **DSP Concepts:** Su sistema de reconocimiento de voz AWE se especializa en la reducción de ruido y comunicación de voz. También ofrecen IA de voz y servicios

offline. Sin embargo, no soportan múltiples hablantes ni servicios en la nube. (ASR en AWE , 2024)

- iFlyTek: Proporciona una amplia gama de características, incluyendo reducción de ruido, comunicación de voz, IA de voz y servicio offline. Aunque no soportan múltiples hablantes ni servicios en la nube, su sistema de reconocimiento de voz es eficiente para aplicaciones de reconocimiento de voz en tiempo real. (iFlytek)
- Krisp: El sistema de reconocimiento de voz de la empresa presenta un sistema de reducción de ruido y comunicación de voz. No ofrecen IA de voz, soporte para múltiples hablantes, servicios offline ni servicios en la nube.
- Hi Auto: Ofrece un sistema de reducción de ruido, comunicación de voz y soporte para IA de voz. No soportan múltiples hablantes, servicios offline ni servicios en la nube.
- PowerVoice: Proporciona una solución integral que incluye reducción de ruido, comunicación de voz, IA de voz y servicio offline. No soportan múltiples hablantes ni servicios en la nube.
- Retune DSP: Su tecnología ofrece reducción de ruido, comunicación de voz, soporte para múltiples hablantes y servicio offline. No proporcionan servicios en la nube.
- Alango: Su sistema de reconocimiento de voz proporciona una gama completa de servicios, incluyendo reducción de ruido, comunicación de voz, IA de voz, soporte para múltiples hablantes y servicio offline. No ofrecen servicios en la nube.

A continuación, se recogen en la tabla a continuación las características que cumplen o no cada uno de los programas de reconocimiento de sonido creados por distintas empresas.

	Kardome	DSP Concepts	iFlyTek	Krisp	Hi Auto	PowerVoice	Retune DSP	Alango
Reducción de Ruido	✓	✓	✓	✓	✓	✓	✓	✓
Comunicación de Voz	✓	✓	✓	✓	✗	✓	✓	✓
IA de Voz	✓	✓	✓	✗	✓	✓	✓	✓
Múltiples Hablantes	✓	✗	✗	✗	✗	✗	✗	✗
Servicio Offline	✓	✓	✓	✗	✓	✓	✓	✓
Servicio en la Nube	✓	✗	✓	✓	✗	✗	✗	✗

Tabla 1 Tabla de características recogidas a través de un estudio de mercado

Además de estos sistemas presentados en la tabla se pueden encontrar otros como, por ejemplo:

- **Google Cloud Speech-to-Text:** Google Cloud Speech-to-Text es una tecnología avanzada de reconocimiento de voz ofrecida por Google Cloud. Proporciona un interfaz potente y escalable que permite convertir grabaciones de voz en texto con una precisión y velocidad altas.
- **Amazon Transcribe:** Amazon Transcribe es un servicio de reconocimiento automático de voz basado en la nube ofrecido por Amazon Web Services (AWS).

Permite convertir grabaciones de voz en texto de manera rápida y precisa, y también incluye capacidades avanzadas como la identificación de parlantes y la transcripción de varios idiomas simultáneamente.

2.2.3.1 KARDOME COMO MEJOR OPCIÓN

- En cuanto a las ventajas que ofrece Kardome se pueden destacar:
- En primer lugar, rempazan los formadores de haz tradicionales con algoritmos de separación de fuentes de IA, lo que permite producir más del 90% mejores resultados de activación de voz en entornos acústicamente desafiantes.
- Gracias a la Biometría de voz, solo las personas autorizadas tendrán la capacidad de arrancar y controlar el vehículo. Con Kardome, se asegura de que las virtudes de esta tecnología funcionen bien incluso en las condiciones acústicas más difíciles.
- Asistentes de voz_múltiples en un solo vehículo, Kardome, puede ofrecer a cada pasajero una experiencia de asistente de voz personal adaptada a sus necesidades únicas y al conjunto de habilidades requeridas.
- La salida de audio la tecnología central de Kardome incluye módulos de cancelación de eco y reducción de ruido que facilitan la salida de audio más clara en escenarios ruidosos.

2.2.3.1.1 HARDWARE DE KARDOME

Los componentes principales de un sistema de Kardome para el interfaz de voces multiusuario son los siguientes:

El GAVEL™ de Kardome es una matriz de 8 micrófonos diseñada para una amplia gama de aplicaciones como interfaz de usuario de voz, vigilancia acústica, radar acústico, investigación de audífonos, llamadas de conferencia y análisis de campos de sonido. Es necesario para capturar la señal acústica.

Con la potencia informática del económico CPU de doble núcleo Cortex-A7, el R328 ofrece una capacidad de cálculo general óptima a un costo extremadamente bajo. Este sistema es capaz de manejar la captura y reproducción de voz convencional sin necesidad de un chip

de voz DSP (Procesador de Señal Digital) externo, gracias a su circuito CODEC altamente integrado, que es responsable de la codificación y decodificación de señales de audio.

El R328 incluye 8 canales de PDM (Modulación de Densidad de Pulso), 16 canales de I2S (Inter-IC Sound), 3 canales de ADC (Convertidores Analógico-Digitales) y 1 canal diferencial de DAC (Convertidores Digital-Analógico), proporcionando una interfaz de entrada/salida de audio rica y flexible. Esta configuración permite diversas opciones de disposición de micrófonos, facilitando la captura de audio.

Además, el módulo de activación de voz de bajo consumo (VAD) integrado en el R328 permite un consumo de energía muy bajo cuando el dispositivo está en estado de espera, esperando comandos de voz. El motor de audio del R328 es compatible con todos los formatos de audio de alta resolución más populares y funciona de manera óptima con la CPU. Su función principal es acelerar los algoritmos multimedia. Esto incluye el procesamiento eficiente de diferentes formatos de audio, como el procesamiento de señales para mejorar la calidad del sonido o reducir el ruido. La aceleración mejora la velocidad de procesamiento, permitiendo una experiencia de usuario fluida.

Este SoC (Sistema en un Chip) integra un Cortex-A7 de doble núcleo, junto con 64 MB de DDR2 o 128 MB de DDR3, así como 3 ADC, 1 DAC, 3 I2S y 8 DMIC (Micrófonos Digitales) de alto rendimiento, ofreciendo una solución de interacción de voz avanzada.

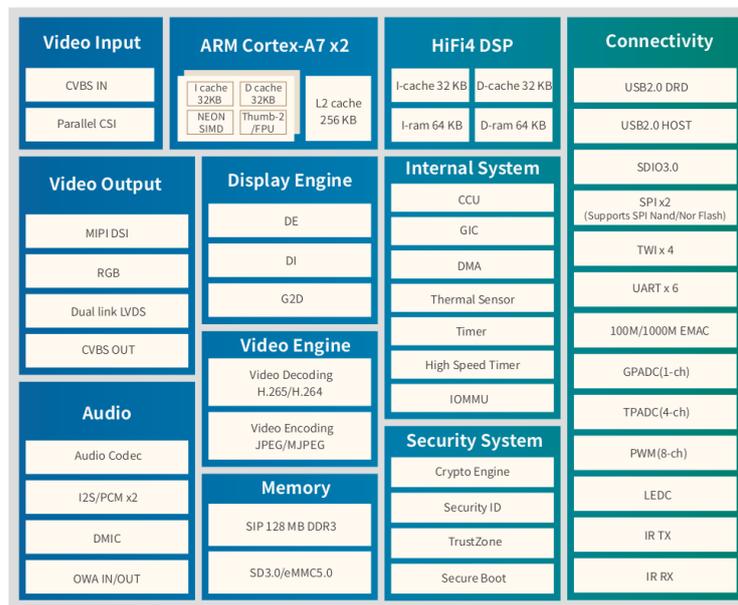


Ilustración 17 Diagrama R328

((CNXSOF), 2022)

Los algoritmos beamforming (algoritmos de formación de haces) se encargan de procesar las señales de audio capturadas por los micrófonos direccionales. Su objetivo principal es enfocar y amplificar la voz del usuario deseado, mientras disminuyen el ruido y los ecos que puedan provenir de otros usuarios o del entorno. El beamforming mejora significativamente la claridad y la separación de las distintas voces.

Después de separar y rastrear las voces de los diferentes usuarios en un entorno con múltiples personas, los algoritmos de procesamiento espacial de usuario se encargan de mezclar y ajustar el audio de cada usuario. Esto se hace para crear una señal de salida unificada y equilibrada, asegurando que todas las voces sean escuchadas de manera clara y sin interferencias entre los usuarios en el entorno.

En resumen, el hardware de Kardome utiliza un GAVEL™, una matriz de 8 micrófonos, para capturar la señal acústica, que luego el R328 junto con los convertidores ADC y DAC trabajan con el fin de detectar el mensaje y reproducirlo.

CAPÍTULO 3. SOLUCIÓN PROPUESTA

En este capítulo se buscará desarrollar una solución basándose en los conceptos explicados en el CAPÍTULO 2. Mediante los sistemas ya estudiados se propondrán los sistemas más adecuados para conseguir la cancelación activa de sonido y el reconocimiento de voz en el interior del automóvil.

El desarrollo de un sistema que combine estas dos tecnologías, para que trabajen de manera sinérgica, es un desafío debido a los requisitos contrapuestos de ambas aplicaciones. Un sistema de cancelación activa de sonido idealmente requiere minimizar la presencia de sonido para que los pasajeros del vehículo no perciban ruidos externos. Por otro lado, un sistema de reconocimiento de voz debe ser capaz de detectar y procesar la voz humana con alta precisión, incluso en un entorno con ruido.

Para abordar el primer tema de cancelar o atenuar los ruidos externos al vehículo, en primer lugar, se deben instalar 16 micrófonos de tipo MICRÓFONOS MEMS, como se describía en el apartado 2.1.2.1, estos micrófonos destacan por su eficiencia energética, característica realmente interesante sobre todo por el avance de los coches eléctricos. Estos micrófonos se colocarán 4 detrás de cada reposacabezas. Esta disposición se debe a que al ser la posición más próxima al oído la cancelación activa de sonido será más sencilla. En este mismo lugar se colocarán los ALTAVOCES. Estos serán de tipo rango completo y subgrave, dos de cada tipo para asegurar que se puede reproducir todo el espectro de frecuencias, esto nos da un total de ocho altavoces de cada tipo. Además, el sistema se debe equipar con un filtro FILTROS FIR (FINITE IMPULSE RESPONSE), cuya característica más destacable es la estabilidad y de una serie de ALGORITMOS GENÉTICOS, cuya función será adaptar el filtro para las distintas situaciones a las que se pueda ver expuesta el vehículo como la intensidad del viento o la rugosidad de la carretera.

Por otro lado, con el fin de conseguir el reconocimiento de voz en el interior del vehículo utilizare la placa base de KARDOME COMO MEJOR OPCIÓN. Esta placa que dispone ya

de toda la parte del hardware y del software por lo que solo se necesita decidir su posición y probar su funcionamiento que se realizará en **¡Error! No se encuentra el origen de la referencia.** a forma de simulación. Tras varias pruebas se llega a la conclusión de que la mejor disposición de esta placa será en el volante del conductor, la posición más cercana para poder capturar la voz de la forma más sencilla, estéticamente también es la posición idónea ya que se puede cubrir con el logotipo de la marca.

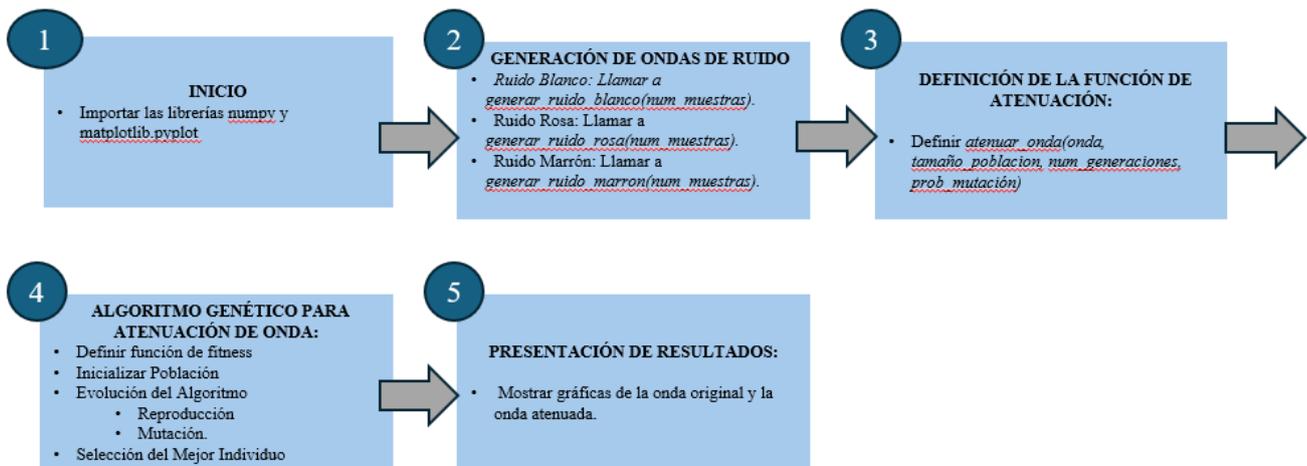


Ilustración 18 Diagrama disposición de elementos

A continuación, se explican los programas pertenecientes al software de la solución propuesta:

3.1 PROGRAMA DE CANCELACIÓN ACTIVA DEL SONIDO

3.1.1 CANCELACIÓN ACTIVA DE LOS RUIDOS



CARACTERÍSTICAS

Ilustración 19 Diagrama del programa de cancelación activa de ruidos

La simulación parte del hecho de tener ya instalado todo el hardware en el interior del automóvil teniendo solo que programar la parte del software de los algoritmos adaptativos para conseguir la cancelación activa del sonido.

En primer lugar en el punto 1 del diagrama se importan las librerías necesarias para ejecutar el código:

Numpy: se utiliza para trabajar con arrays (matrices) multidimensionales y funciones matemáticas de alto nivel.

matplotlib.pyplot: es un módulo dentro de Matplotlib que proporciona una interfaz de estilo similar a MATLAB para la creación de gráficos.

En el punto 2 del diagrama, se generan ondas aleatorias de ruido blanco, rosa o marrón, esto se consigue gracias a la importación de la librería *numpy* encargada de generar valores de forma aleatoria, por su parte, la librería *matplotlib.pyplot* se encarga de representar en una gráfica dichos valores.

Como se ha explicado anteriormente, el ruido blanco, marrón y rosa tienen características diferentes que deben verse representadas en el código:

Para generar una onda de ruido blanco, caracterizada por una densidad espectral de potencia constante, se utiliza la función *def generar_ruido_blanco(num_muestras)*. Esta función espera la entrada *num_muestras*, un entero que indica la cantidad de muestras de ruido blanco que se desea generar y devuelve una matriz de números aleatorios con una distribución normal estándar.

A la hora de conseguir una onda de ruido rosa, se utiliza la función *def generar_ruido_rosa(num_muestras)* que aplica una suma acumulativa al ruido blanco con *np.cumsum*, lo que da como resultado una señal con una disminución de potencia a medida que aumentan las frecuencias (característica del ruido rosa). Además, se debe normalizar la señal dividiéndola por su valor máximo absoluto para asegurarse de que la amplitud máxima sea 1.

Por último, para generar un espectro de onda marrón o browniana, se utiliza la función *def generar_ruido_marron(num_muestras)*. A partir de un espectro de ruido blanco se normaliza la señal para que tenga una energía promedio constante dividiéndola por la raíz cuadrada de la suma de los cuadrados de los valores de la señal, escalada por el número de muestras, de esta forma se consigue un ruido a frecuencias bajas.

Una vez hemos conseguido generar los tres tipos de ruidos que queremos atenuar de forma aleatoria, se debe definir una función con el fin de atenuar dichos ruidos, esto se realiza en el punto 3 del diagrama. Para ello se utiliza la función *def atenuar_onda(onda, tamaño_poblacion, num_generaciones, prob_mutacion)* Esta función utiliza un algoritmo

genético para encontrar un conjunto de coeficientes que atenúen la onda de entrada. Las entradas de la función son:

- onda: La señal de entrada que se desea atenuar.
- tamaño_poblacion: Número de individuos en la población inicial. (50-200 individuos)
- num_generaciones: Número de iteraciones del algoritmo genético.(50-200 generaciones)
- prob_mutacion: Probabilidad de que ocurra una mutación en un individuo. (1%-10%) La probabilidad de mutación debe ser lo suficientemente baja para que el algoritmo se enfoque en mejorar las mejores soluciones encontradas, pero lo suficientemente alta para mantener la diversidad genética y evitar el estancamiento en mínimos locales.

En el punto 4 del diagrama con la función *def atenuar_onda(onda, tamaño_poblacion, num_generaciones, prob_mutacion)* da como salida la secuencia de coeficientes que produce la onda atenuada con el menor error cuadrático medio posible. Si la onda tiene 1000 muestras, cada individuo será un array de 1000 números.

Aun que ha sido explicado anteriormente un algoritmo genético en el CAPITULO 2: ESTADO DE LA CUESTIÓN es un algoritmo adaptativo que busca los coeficientes que debe de obtener el filtro inspirándose en los procesos evolutivos naturales.

El proceso comienza con la creación de una población inicial de individuos. Aquí es donde se realiza la transición de las entradas proporcionadas por el usuario (tamaño de la población, número de generaciones, probabilidad de mutación) hacia la creación de estos individuos.

```
poblacion = np.random.uniform(0, 1, (tamaño_poblacion, len(onda)))
```

np.random.uniform(0, 1, (tamaño_poblacion, len(onda))) crea una matriz donde cada fila representa un individuo y cada columna representa un coeficiente de atenuación para una muestra en la onda de ruido.

Tamaño de Población (*tamaño_poblacion*): Este valor determina cuántos individuos se generan inicialmente. Por ejemplo, si *tamaño_poblacion* es 50, se crean 50 individuos.

Longitud del Individuo (*len(onda)*): La longitud de cada individuo es igual al número de muestras en la onda de ruido. Si *len(onda)* es 1000, cada individuo tendrá 1000 coeficientes.

Una vez creada la población inicial, cada individuo se evalúa en términos de su fitness, que mide qué tan efectivo es un individuo para atenuar el ruido.

def calcular_fitness(individuo):

*onda_generada = individuo * onda*

error = onda_generada - onda

mse = np.mean(np.square(error))

return 1 / mse

Multiplicación Elemento a Elemento: *onda_generada = individuo * onda* genera una onda atenuada (una versión modificada de la onda original) aplicando los coeficientes de atenuación del individuo.

Cálculo del Error: *error = onda_generada - onda* determina la diferencia entre la onda atenuada y la onda original.

Error Cuadrático Medio (MSE): *mse = np.mean(np.square(error))* mide el error promedio entre la onda original y la onda atenuada.

Fitness Inverso al MSE: El fitness es la inversa del MSE, así que un menor error resultará en un mayor fitness.

Una vez que se ha calculado el fitness de cada individuo, se procede a la selección, cruce y mutación para evolucionar la población hacia individuos mejores (más aptos).

for i in range(tamaño_poblacion):

padre1 = poblacion[i]

padre2 = poblacion[np.random.randint(0, tamaño_poblacion)]

hijo = (padre1 + padre2) / 2

Selección de Padres:

padre1: Se selecciona directamente de la población ordenada.

padre2: Se selecciona aleatoriamente de la población.

Cálculo del Hijo: $hijo = (padre1 + padre2) / 2$ genera un nuevo individuo (hijo) tomando el promedio de los coeficientes de los padres. Esto es un cruce simple. El cruce permite que los hijos hereden características de ambos padres, lo que facilita la exploración de nuevas áreas en el espacio de soluciones.

Después de crear un hijo, se introduce la mutación para mantener la diversidad genética.

for j in range(len(onda)):

if np.random.rand() < probab_mutacion:

hijo[j] = np.random.uniform(0, 1)

Proceso de Mutación: Para cada coeficiente en el hijo, existe una *probab_mutacion* probabilidad de que se reemplace por un nuevo valor aleatorio entre 0 y 1. La mutación introduce cambios aleatorios para evitar que la población se estanque en soluciones subóptimas.

El algoritmo continúa iterando a través de generaciones, seleccionando los mejores individuos, aplicando cruce y mutación, y evolucionando la población hacia soluciones mejores.

for generacion in range(num_generaciones):

num_generaciones: Determina cuántas veces se repite el ciclo de evaluación, selección, cruce y mutación.

Mejoramiento Progresivo: Con cada generación, los individuos de la población deberían, en promedio, volverse mejores para atenuar el ruido.

En el punto 5 simplemente se muestran los resultados a través de las siguientes líneas de código: `plt.subplot(212)`

plt.plot(onda_atenuada)

plt.title("Onda Atenuada (Nivel de Atenuación: {:.2f}").format(nivel_atenuacion))

plt.xlabel("Muestra")

plt.ylabel("Amplitud")

plt.tight_layout()

plt.show()

3.2 EXPLICACIÓN PROGRAMA DE RECONOCIMIENTO DE SONIDO

3.2.1 EXPLICACIÓN PROGRAMA ESPECTRO DE AUDIO EN TIEMPO REAL

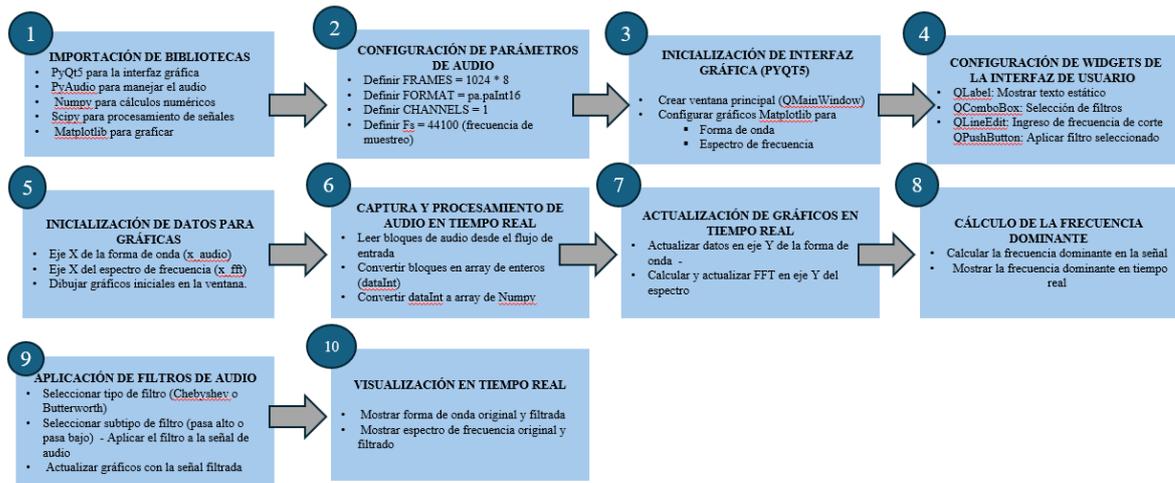


Ilustración 20 Diagrama del programa de espectro de audio en tiempo real

El primer programa, consiste en un espectro de audio en tiempo real, con una serie de filtros específicos para poder segmentar la señal. En este programa se simula un sistema de obtención de las características del habla a través de la transformada de Fourier, en concreto la frecuencia dominante o frecuencia fundamental. Se trata de la frecuencia más baja en el espectro de una señal de voz y corresponde a la frecuencia con la que las cuerdas vocales vibran durante la producción de un sonido. Esta frecuencia varía según la persona y por ello es el parámetro más relevante para la biometría de la voz.

El código comienza importando las bibliotecas necesarias, que incluyen *PyQt5* para la interfaz gráfica, *PyAudio* para el manejo del audio, *Numpy* para poder realizar cálculos numéricos, *Scipy* para el procesamiento de señales, y *Matplotlib* para poder dibujar gráficas. Estos pasos corresponden al punto 1 del diagrama.

A continuación, en el punto 2 del diagrama, se deben de configurar los parámetros de audio. Los *FRAMES* son unidades de audio que representan una muestra del sonido en un instante dado, se les otorga el valor de $1024 \cdot 8$. Este valor es muy común en el procesamiento de señales de audio ya que al ser múltiplo de 2 hace que el cálculo de la FFT (transformada de Fourier) sea muy eficiente. Ofrece un buen equilibrio entre resolución en frecuencia y tiempo, y se beneficia de la eficiencia computacional de la FFT. El parámetro de *FORMAT* define el tipo de datos que se han empleado para representar la muestra de audio, en nuestro caso, *FORMAT* = *pa.paInt16*, que indica que cada muestra será un número entero de 16 bits con signo. Esto significa que cada muestra puede tomar valores entre -32768 y 32767. El formato determina a precisión de la señal que se está representando a mayor número de bits será un formato más preciso. *CHANNELS=1* nos indica el número de canales de audio. El parámetro *Fs=44100*, es la frecuencia de muestreo, el número de muestras tomadas por segundo medidas en Hz. El valor 44100 es una frecuencia de muestreo común para señales de audio de alta calidad.

En el punto 3 del diagrama, se establecen las bases para poder visualizar el espectro de frecuencias de una señal de audio a través de *AudioSpectrumApp* obtenida de la clase *QMainWindow* obtenida de *PyQt5*. Las siguientes líneas del código tienen la función de capturar datos de audio en tiempo real y crea dos gráficos de Matplotlib para visualizar tanto la forma de onda original como su correspondiente espectro de frecuencia. Además, se establecen los límites de los ejes en los gráficos para asegurar una visualización adecuada de los datos.

Para continuar en el punto 4 del diagrama, se deben establecer los Widgets de Interfaz de Usuario, que hacen referencia a los componentes visuales que permiten interactuar al usuario con el programa. Se distinguen distintos tipos. En primer lugar, el *QLabel* se utiliza para mostrar texto estático o imágenes es decir, simplemente muestran información al usuario. El *QComboBox* es un widget que le permite al usuario seleccionar una opción a través de un desplegable. El *QLineEdit* sirve para ingresar números en el programa, en este código sirve para ingresar la frecuencia de corte. *QPushButton* nos permite aplicar un filtro seleccionado.

Aparece la opción de seleccionar filtros que pueden ser filtros Chebyshev o Butterworth. Cada uno tiene características distintivas que lo hacen adecuado para diferentes propósitos, son importantes para los sistemas de reconocimiento de audio para mejorar la calidad del sonido. Debido a sus características los filtros Chebyshev se utilizan para eliminar el ruido, eliminando frecuencias no deseadas. Por su parte los filtros Butterworth son adecuados para suavizar la señal y sin distorsión en la banda de paso, lo cual es crucial para mantener la integridad de las características de la voz.

A continuación, en el punto 5, se inicializan los datos de los gráficos y se inicializan arrays (vector que almacena datos) Estos arrays se denominan *x_audio* para representar en el eje x la forma de onda y *x_fft* con valores lineales que representan las frecuencias en Hz. Posteriormente se “llama” a la función *self.canvas.draw()* para que se muestren los elementos gráficos iniciales en la ventana de la aplicación. Aunque inicialmente las líneas de los filtros muestren valores aleatorios, se actualizarán con los datos reales de la señal y su espectro una vez se empiece a capturar audio.

El programa actualiza el gráfico en tiempo real (punto 6) a través de *def update_plot(self*. En primer lugar lee un bloque de audio del flujo abierto, *data = self.stream.read(FRAMES)*, una vez ha recogido los bloques de audio los desempaqueta en una array de enteros de 16 bits *dataInt* este formato ya se había definido previamente.

Lo siguiente que hace el programa es convertir el array de enteros definido en *dataInt* en un array de la librería *numpy* para poder procesarlos y representarlos en el gráfico. A través del comando *self.line.set_ydata(self.data)* se representan los datos en el eje.

El comando *self.line_fft.set_ydata(abs(fourier.fft(self.data))[:FRAMES // 2])* calcula la Transformada Rápida de Fourier (FFT) de la señal (*self.data*) usando la librería *fourier*. Una vez calculada se toma el módulo del resultado y se divide entre dos para representar frecuencias positivas. Y se ajusta el límite del eje para adaptarse al rango de frecuencias. Esto se realiza en el punto 7 del diagrama.

En las siguientes líneas del código, en el punto 8 del diagrama:

```
dominant_freq_audio =self.calculate_dominant_frequency(self.data, Fs)
```

```
self.dominant_freq_label.setText(f"Frecuencia dominante: {dominant_freq_audio:.2f} Hz")
```

En la primera se llama a la función `calculate_dominant_frequency` para calcular la frecuencia dominante en la señal de audio. En la segunda línea se visualiza la cifra de la frecuencia dominante en tiempo real.

El fragmento de código que viene a continuación se encarga de aplicar un filtro de audio seleccionado por el usuario a la señal de entrada y actualizar los gráficos en tiempo real para visualizar tanto la señal original como la filtrada. Si el usuario ha habilitado la opción de filtrado, el código verifica el tipo de filtro (Chebyshev o Butterworth) y su subtipo (pasa alto o pasa bajo) para seleccionar la función de filtrado adecuada `self.current_filter_type` y `self.current_filter_subtype`. Luego, aplica este filtro a la señal de audio y almacena el resultado en una cola (punto 9). La cola se utiliza para mantener un historial limitado de la señal filtrada, lo que permite visualizar su evolución en el tiempo. Finalmente, se actualizan las líneas de los gráficos correspondientes para mostrar tanto la forma de onda como el espectro de frecuencia de la señal original y la filtrada (punto 10). Esta actualización se realiza de manera periódica para proporcionar una visualización en tiempo real del proceso de filtrado.

3.2.2 EXPLICACIÓN PROGRAMA DICTADO DE VOZ

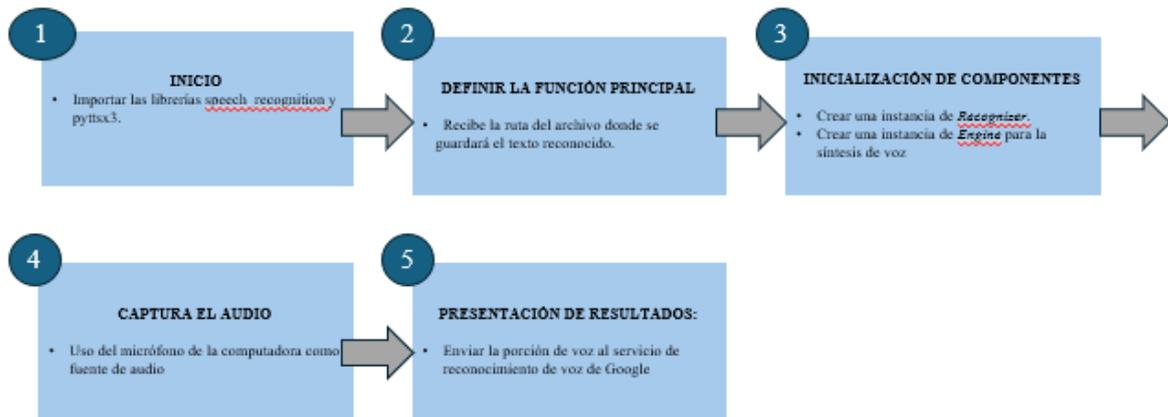


Ilustración 21 Diagrama del programa de dictado de voz

Inicialmente, importa las librerías necesarias: `speech_recognition` para interactuar con dispositivos de audio y servicios de reconocimiento de voz, y `pytsx3` para convertir texto en audio (punto 1 del diagrama). Luego, en el punto 2, se define una función que recibe como parámetro la ruta de un archivo donde se guardará el texto reconocido. Dentro de esta función, en el punto 3 se crea instancias de las clases `Recognizer` y `Engine` para realizar el reconocimiento y la síntesis de voz respectivamente. A continuación, como se indica en el punto 4 del diagrama, utilizando el micrófono como fuente de audio, captura una porción de voz y la envía al servicio de reconocimiento de voz de Google para obtener el texto correspondiente. El texto obtenido se guarda en el archivo especificado y se reproduce a través de los altavoces (punto 5). El código también incluye bloques `try-except` para manejar posibles errores que puedan surgir durante el proceso de reconocimiento de voz. Finalmente, se ejecuta la función con la ruta del archivo de salida especificada.

3.2.3 EXPLICACIÓN DEL PROGRAMA DE RECONOCIMIENTO DE VOZ

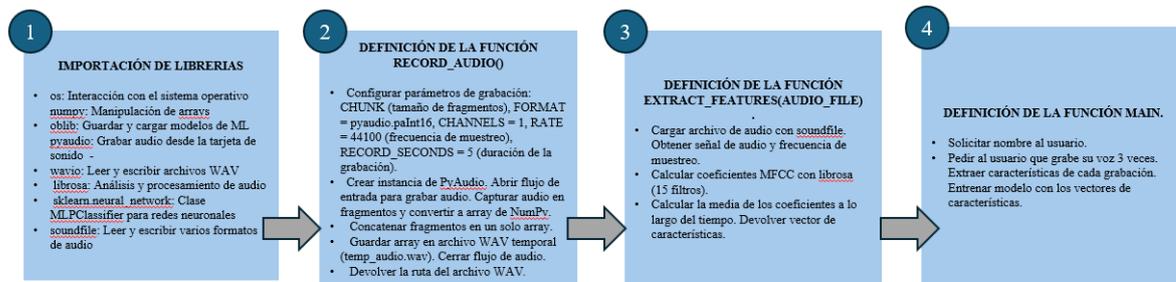


Ilustración 22 Diagrama del programa de reconocimiento de voz

En primer lugar, (punto 1) debemos importar las librerías de Python. Se importan las siguientes librerías:

- `os`: Proporciona funciones para interactuar con el sistema operativo, como crear o eliminar archivos.
- `numpy`: Librería base para el cálculo y la manipulación de arrays en Python.
- `joblib`: Librería para guardar y cargar modelos de aprendizaje automático.
- `pyaudio`: Librería para grabar audio a través de la tarjeta de sonido.
- `wavio`: Librería para leer y escribir archivos de audio WAV.
- `librosa`: Librería especializada en análisis y procesamiento de audio.
- `sklearn.neural_network`: Módulo de `scikit-learn` que proporciona la clase `MLPClassifier` para redes neuronales multicapa.
- `soundfile`: Librería para leer y escribir varios formatos de audio, incluyendo WAV.

A continuación en el punto 2 del diagrama, se define la función `def record_audio()`. Esta función se encarga de capturar audio del micrófono del usuario, configurando parámetros como el tamaño de los fragmentos de audio (`CHUNK`), el formato (`FORMAT = pyaudio.paInt16`), el número de canales (`CHANNELS`) la frecuencia de muestreo (`RATE =`

44100) y la duración de la grabación ($RECORD_SECONDS = 5$). Crea una instancia de *PyAudio* para interactuar con la tarjeta de sonido y abre un flujo de entrada para recibir el audio. A medida que se captura el audio, cada fragmento se convierte a un array de *NumPy* y se almacena en una lista. Es decir, la salida que nos da esta función sería *temp_wav_file* (*str*) que es la ruta del archivo WAV temporal creado, que contiene la grabación de audio. Una vez finalizada la grabación, todos los fragmentos se concatenan en un único array y se guardan en un archivo WAV temporal (*temp_audio.wav*) utilizando *wavio*. Finalmente, se cierra el flujo de audio y se devuelve la ruta del archivo WAV creado, el cual servirá como entrada para las siguientes etapas del procesamiento de audio.

Una vez capturado el audio, en el punto 3 del diagrama se define *def extract_features(audio_file)* la función encargada de extraer las características acústicas relevantes a partir del archivo WAV creado previamente en la función *record_audio()*. La entrada de esta función es el audio es la ruta del archivo de audio el cual queremos reconocer. La salida de la función es *mfccs_mean* (*numpy.ndarray*), un vector que representa la media de los coeficientes MFCC extraídos del audio. Calcula los coeficientes MFCC (Mel-Frequency Cepstral Coefficients), que son ampliamente utilizados en el reconocimiento de voz. Para ello, primero carga el archivo de audio utilizando la librería *soundfile* y obtiene la señal de audio y su frecuencia de muestreo. A continuación, emplea la función *mfccs* de la librería *librosa* para calcular los coeficientes MFCC, explicados previamente en el documento los cuales representan una forma compacta y discriminativa de la información espectral del audio. Cabe destacar que el número de filtros se iguala a 15 que como se mencionó anteriormente es el número necesario de filtros para poder implementar un sistema de reconocimiento de voz. Finalmente, se calcula la media de los coeficientes a lo largo del tiempo, obteniendo así un vector de características que representa de manera global al audio. Este vector de características es el que se devuelve como salida de la función y servirá como entrada para el modelo de clasificación.

Una vez definidas estas funciones se debe programar el *main*, punto 4 del diagrama. En primer lugar se solicita el nombre al usuario y se le pide que grabe su voz un total de 3 veces para poder extraer las características, de esta forma se entrena el programa.

Por último se crea un bucle infinito en el que graba la voz del usuario, extrae sus características, aplica el modelo previamente entrenado para predecir su identidad basándose en los coeficientes de Mel y muestra su nombre.

3.2.4 EXPLICACIÓN PROGRAMA SEPARACION DE VOCES

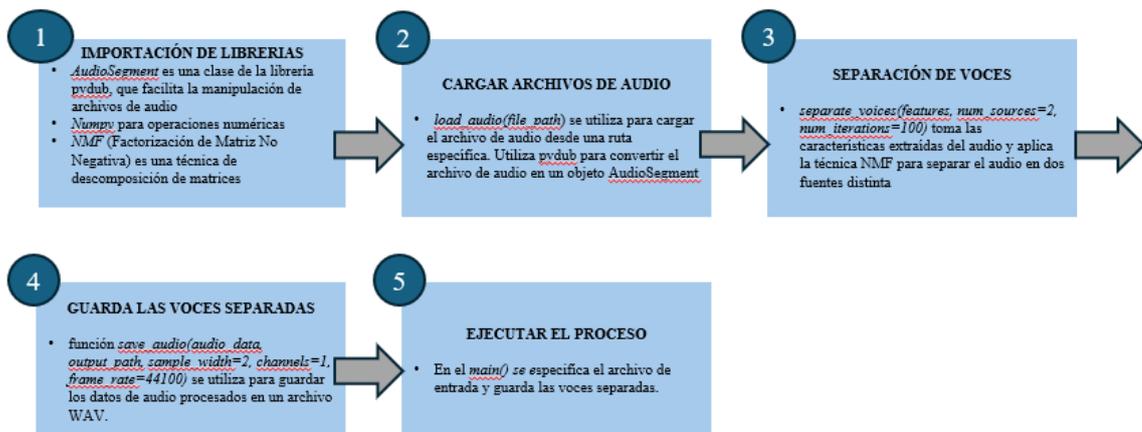


Ilustración 23 Diagrama del programa de separación de voces

Lo primero de todo es importar las librerías, punto 1 del diagrama que nos serán necesarias para ejecutar el código:

- `from pydub import AudioSegment`: Importa `AudioSegment` de la librería `pydub`, que facilita la manipulación de archivos de audio.
- `import numpy as np`: Importa la librería `numpy` para manipulación de arreglos.
- `from sklearn.decomposition import NMF`: Importa `NMF` (Non-negative Matrix Factorization) de `scikit-learn` para la descomposición de matrices.

A continuación, en el punto 2 del diagrama, se define la función `def load_audio(file_path)`, que utiliza la librería `pydub` para cargar el archivo de audio.

En las siguientes líneas del código, en el punto 3 del diagrama, se define la función `separate_voices(features, num_sources=2, num_iterations=100)`, que toma como entradas las características que han sido extraídas del audio (`features`), el número de fuentes de audio (`num_sources`), por defecto 2) y el número máximo de iteraciones del algoritmo NMF,

(*num_iterations*, por defecto 100). El siguiente paso es crear un modelo NMF (*Non-Negative Matrix Factorization*). Un modelo NMF no es más que una técnica de análisis de datos que consiste en descomponer una matriz no negativa (el array donde se ha guardado el audio con las dos voces) en un producto de dos matrices no negativas. Esta descomposición es de gran utilidad cuando las matrices resultantes nos aportan información valiosa, como es en nuestro caso donde buscamos obtener dos matrices donde se recojan los datos de las dos voces por separado. Esta función nos devuelve dos matrices W y H.

Posteriormente, en el punto 4, se define la función para guardar el audio *def save_audio(audio_data, output_path, sample_width=2, channels=1, frame_rate=44100)* donde se define una función para guardar los datos de audio en un archivo en formato WAV.

Por último en el *main()* se especifica la ruta del archivo de audio con las dos voces que se tratan de separar (*input_file*) y se devuelven las dos voces por separado *output_files*. (punto 5)

CAPÍTULO 4. ENSAYOS Y RESULTADOS

4.1 CÓDIGOS

4.1.1 CODIGOS PYTHON CANCELACIÓN ACTIVA DE SONIDO

```
import numpy as np
import matplotlib.pyplot as plt

# Generar ruido blanco
def generar_ruido_blanco(num_muestras):
    return np.random.randn(num_muestras)

# Generar ruido rosa
def generar_ruido_rosa(num_muestras):
    ruido_blanco = np.random.randn(num_muestras)
    filtro = np.cumsum(ruido_blanco)
    return filtro / np.max(np.abs(filtro))

# Generar ruido marrón
def generar_ruido_marron(num_muestras):
    ruido_blanco = np.random.randn(num_muestras)
    filtro = np.cumsum(ruido_blanco)
    return filtro / np.sqrt(np.sum(np.square(filtro)) /
num_muestras)

# Algoritmo genético para atenuar la onda
def atenuar_onda(onda, tamaño_poblacion, num_generaciones,
prob_mutacion):
    def calcular_fitness(individuo):
        onda_generada = individuo * onda
        error = onda_generada - onda
        mse = np.mean(np.square(error))
        return 1 / mse

    poblacion = np.random.uniform(0, 1, (tamaño_poblacion,
len(onda)))

    for generacion in range(num_generaciones):
```

```
    fitness = np.array([calcular_fitness(individuo) for
individuo in poblacion])
    indices_ordenados = np.argsort(fitness)[::-1]
    poblacion =
poblacion[indices_ordenados[:tamaño_poblacion]]

    for i in range(tamaño_poblacion):
        padre1 = poblacion[i]
        padre2 = poblacion[np.random.randint(0,
tamaño_poblacion)]
        hijo = (padre1 + padre2) / 2

        for j in range(len(onda)):
            if np.random.rand() < probab_mutacion:
                hijo[j] = np.random.uniform(0, 1)

        poblacion[i] = hijo
    mejor_individuo = poblacion[0]
    onda_atenuada = mejor_individuo * onda
    nivel_atenuacion = np.mean(mejor_individuo)

    return onda_atenuada, nivel_atenuacion

# Parámetros del programa
duracion = 1 # Duración en segundos
tasa_muestreo = 1000 # Tasa de muestreo en muestras por segundo
num_muestras = int(duracion * tasa_muestreo)

# Selección del tipo de ruido
tipo_ruido = input("Selecciona el tipo de ruido (A: ruido blanco,
B: ruido rosa, C: ruido marrón): ")

if tipo_ruido.lower() == 'a':
    onda = generar_ruido_blanco(num_muestras)
elif tipo_ruido.lower() == 'b':
    onda = generar_ruido_rosa(num_muestras)
elif tipo_ruido.lower() == 'c':
    onda = generar_ruido_marron(num_muestras)
else:
    print("Opción inválida.")
    exit()

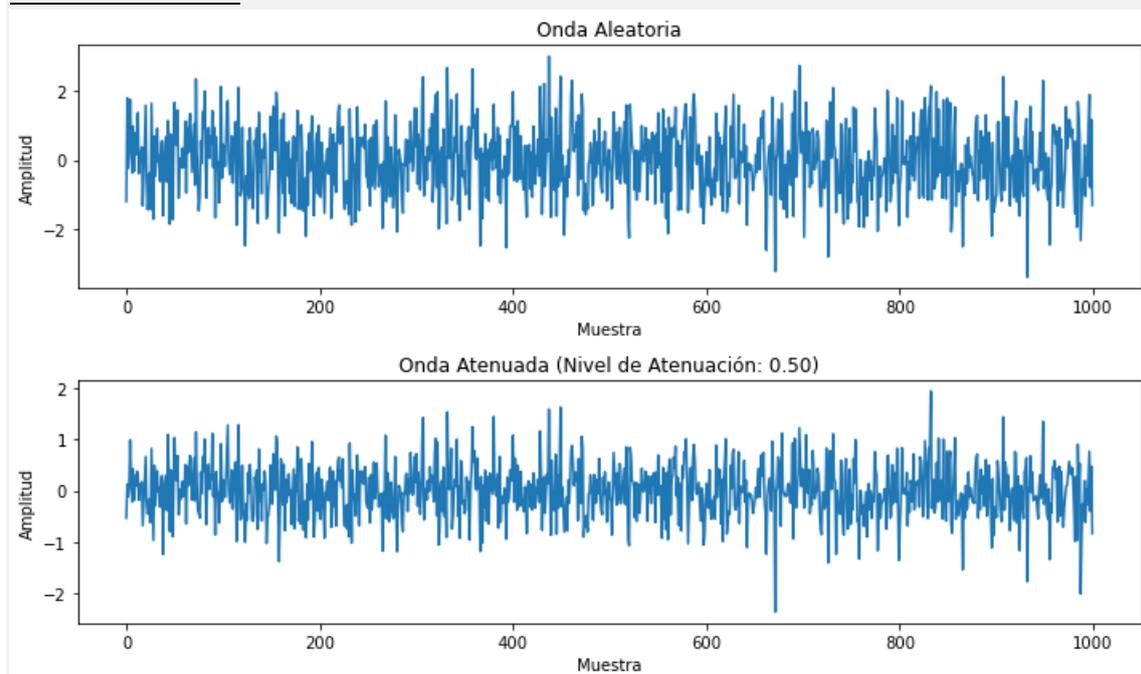
# Generar la onda aleatoria
plt.figure(figsize=(10, 6))
```

```
plt.subplot(211)
plt.plot(onda)
plt.title("Onda Aleatoria")
plt.xlabel("Muestra")
plt.ylabel("Amplitud")

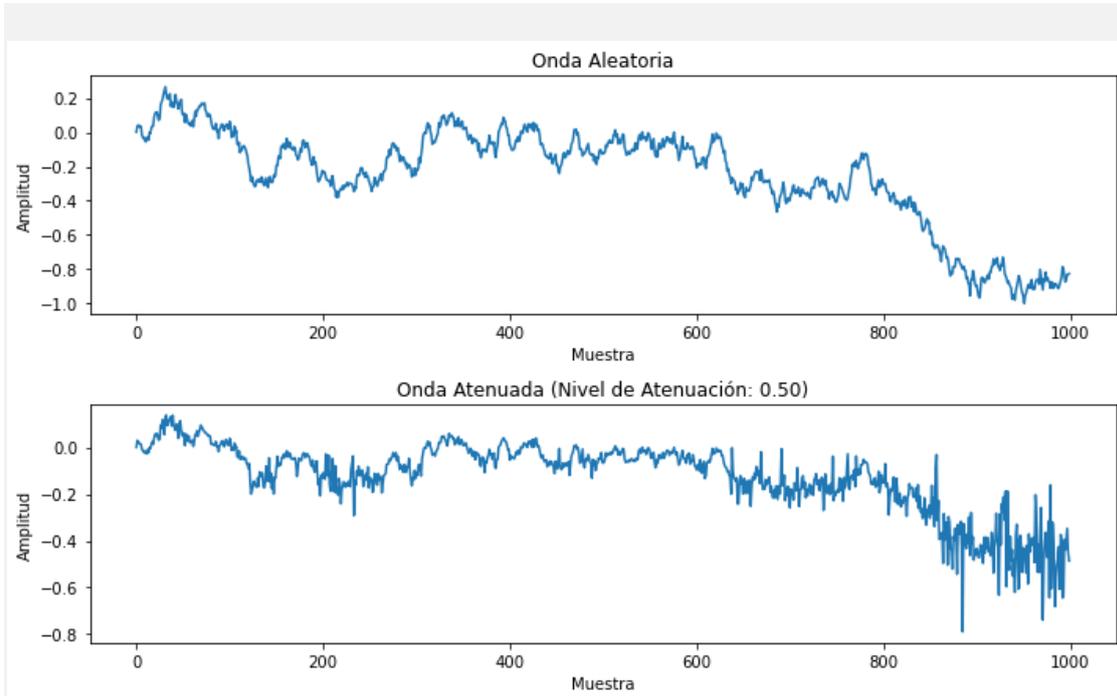
# Atenuar la onda con algoritmo genético
tamaño_poblacion = 50
num_generaciones = 100
prob_mutacion = 0.1
onda_atenuada, nivel_atenuacion = atenuar_onda(onda,
tamaño_poblacion, num_generaciones, prob_mutacion)
# Mostrar la onda atenuada
plt.subplot(212)
plt.plot(onda_atenuada)
plt.title("Onda Atenuada (Nivel de Atenuación:
{: .2f})".format(nivel_atenuacion))
plt.xlabel("Muestra")
plt.ylabel("Amplitud")

plt.tight_layout()
plt.show()
```

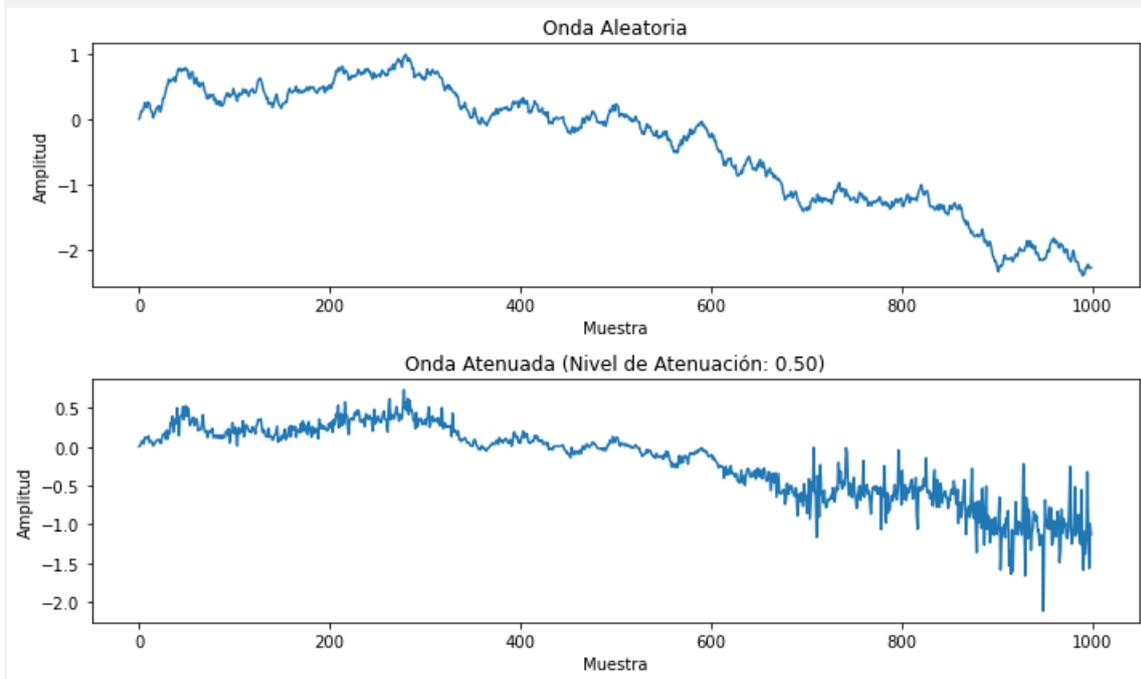
Ruido Blanco:



Ruido Rosa:



Ruido Marrón:



4.1.2 CÓDIGOS DE PYTHON RECONOCIMIENTO DE SONIDO

ESPECTRO DE AUDIO CON DETECCIÓN DE LA FRECUENCIA DOMINANTE PARA PODER DIGITALIZAR LA SEÑAL DE AUDIO

```
import sys
import struct
import pyaudio as pa
import numpy as np
import scipy.fftpack as fourier
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QVBoxLayout,
QPushButton, QComboBox, QLabel
from PyQt5.QtWidgets import QLineEdit
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
from matplotlib.figure import Figure
from PyQt5.QtCore import QTimer
from scipy import signal

FRAMES = 1024 * 8
FORMAT = pa.paInt16
CHANNELS = 1
Fs = 44100

class AudioSpectrumApp(QMainWindow):
    def __init__(self):
        super().__init__()

        self.p = pa.PyAudio()

        self.stream = self.p.open(
            format=FORMAT,
            channels=CHANNELS,
            rate=Fs,
            input=True,
            output=True,
            frames_per_buffer=FRAMES
        )

        self.central_widget = QWidget()
        self.setCentralWidget(self.central_widget)

        self.layout = QVBoxLayout()
        self.central_widget.setLayout(self.layout)

        self.canvas = FigureCanvas(Figure())
        self.layout.addWidget(self.canvas)

        self.ax = self.canvas.figure.add_subplot(211)
        self.ax1 = self.canvas.figure.add_subplot(212)

        self.ax.set_ylim(-32500, 32500)
        self.ax.set_xlim(20, 20 + (FRAMES - 1) * Fs / FRAMES)
```

```
self.Fmin = 20
self.Fmax = 20000
self.ax1.set_xlim(self.Fmin, self.Fmax)

self.ax1.set_xlabel('Frecuencia (Hz)')
self.ax1.set_ylabel('Transformada de Fourier')

self.ax1.set_title('Espectro de frecuencia')

self.filter_choice_label = QLabel("¿Desea filtrar la señal?")
self.layout.addWidget(self.filter_choice_label)

self.filter_choice_combobox = QComboBox()
self.filter_choice_combobox.addItem("No")
self.filter_choice_combobox.addItem("Sí")
self.layout.addWidget(self.filter_choice_combobox)

self.filter_type_label = QLabel("Tipo de filtro:")
self.layout.addWidget(self.filter_type_label)

self.filter_type_combobox = QComboBox()
self.filter_type_combobox.addItem("Chebyshev")
self.filter_type_combobox.addItem("Butterworth")
self.layout.addWidget(self.filter_type_combobox)

self.filter_subtype_label = QLabel("Tipo de filtro:")
self.layout.addWidget(self.filter_subtype_label)

self.filter_subtype_combobox = QComboBox()
self.filter_subtype_combobox.addItem("Paso alto")
self.filter_subtype_combobox.addItem("Paso bajo")
self.layout.addWidget(self.filter_subtype_combobox)

self.cutoff_label = QLabel("Frecuencia de corte:")
self.layout.addWidget(self.cutoff_label)

self.cutoff_input = QLineEdit()
self.layout.addWidget(self.cutoff_input)

self.apply_button = QPushButton("Aplicar filtro")
self.apply_button.clicked.connect(self.apply_filter)
self.layout.addWidget(self.apply_button)

self.filter_active = False
self.current_filter_type = None
self.current_filter_subtype = None
self.current_cutoff_freq = None

self.data = np.zeros(FRAMES, dtype=np.int16)
self.filtered_data = np.zeros(FRAMES, dtype=np.int16)
self.signal_queue = []

self.x_audio = np.linspace(20, 20 + (FRAMES - 1) * Fs / FRAMES, FRAMES)
self.x_fft = np.linspace(20, 20000, FRAMES // 2)
```

```
self.line, = self.ax.plot(self.x_audio, self.data, 'r')
self.line_fft, = self.ax1.plot(self.x_fft, np.random.rand(FRAMES // 2),
'b')
self.line_filtered, = self.ax.plot(self.x_audio, self.filtered_data, 'g')
self.line_fft_filtered, = self.ax1.plot(self.x_fft, np.random.rand(FRAMES
// 2), 'm')

self.canvas.draw()

self.timer = QTimer()
self.timer.timeout.connect(self.update_plot)
self.timer.start(10)

self.dominant_freq_label = QLabel("Frecuencia dominante:")
self.layout.addWidget(self.dominant_freq_label)

self.setWindowTitle("Audio Spectrum App")
self.setGeometry(100, 100, 800, 600)

def apply_filter(self):
    filter_choice = self.filter_choice_combobox.currentText()

    if filter_choice == "Sí":
        self.filter_active = True
        self.current_filter_type =
self.filter_type_combobox.currentText().lower()
        self.current_filter_subtype =
self.filter_subtype_combobox.currentText().lower()
        self.current_cutoff_freq = float(self.cutoff_input.text())
    else:
        self.filter_active = False

def update_plot(self):
    data = self.stream.read(FRAMES)
    dataInt = struct.unpack(str(FRAMES) + 'h', data)
    self.data = np.array(dataInt)

    self.line.set_ydata(self.data)
    self.line_fft.set_ydata(abs(fourier.fft(self.data))[:FRAMES // 2])
    self.ax1.set_ylim(0, np.max(abs(fourier.fft(self.data)[:FRAMES // 2])) +
10)

    dominant_freq_audio = self.calculate_dominant_frequency(self.data, Fs)
    self.dominant_freq_label.setText(f"Frecuencia dominante:
{dominant_freq_audio:.2f} Hz")

    if self.filter_active:
        if self.current_filter_subtype == "paso alto":
            if self.current_filter_type == "chebyshev":
                self.filtered_data =
self.apply_chebyshev_highpass_filter(self.data, self.current_cutoff_freq)
            elif self.current_filter_type == "butterworth":
```

```
        self.filtered_data =
self.apply_butterworth_highpass_filter(self.data, self.current_cutoff_freq)
        elif self.current_filter_subtype == "paso bajo":
            if self.current_filter_type == "chebyshev":
                self.filtered_data =
self.apply_chebyshev_lowpass_filter(self.data, self.current_cutoff_freq)
            elif self.current_filter_type == "butterworth":
                self.filtered_data =
self.apply_butterworth_lowpass_filter(self.data, self.current_cutoff_freq)
            self.signal_queue.append(self.filtered_data.copy())
            if len(self.signal_queue) > 100:
                self.signal_queue.pop(0)
            self.line_filtered.set_ydata(self.signal_queue[-1])
            self.line_fft_filtered.set_ydata(abs(fourier.fft(self.signal_queue[-
1]))[:FRAMES // 2])

        self.canvas.draw()

def apply_chebyshev_highpass_filter(self, data, cutoff_freq):
    nyquist_freq = 0.5 * Fs
    normal_cutoff = cutoff_freq / nyquist_freq
    b, a = signal.cheby1(4, 0.5, normal_cutoff, btype='high', analog=False)
    filtered_data = signal.lfilter(b, a, data)
    return filtered_data

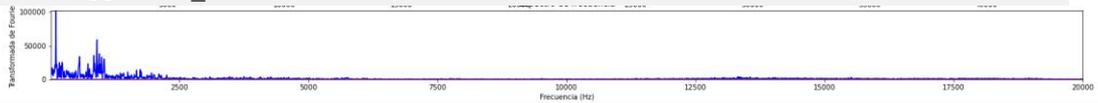
def apply_butterworth_highpass_filter(self, data, cutoff_freq):
    nyquist_freq = 0.2 * Fs
    normal_cutoff = cutoff_freq / nyquist_freq
    b, a = signal.butter(4, normal_cutoff, btype='high', analog=False)
    filtered_data = signal.lfilter(b, a, data)
    return filtered_data

def apply_chebyshev_lowpass_filter(self, data, cutoff_freq):
    nyquist_freq = 0.5 * Fs
    normal_cutoff = cutoff_freq / nyquist_freq
    b, a = signal.cheby1(4, 0.5, normal_cutoff, btype='low', analog=False)
    filtered_data = signal.lfilter(b, a, data)
    return filtered_data

def apply_butterworth_lowpass_filter(self, data, cutoff_freq):
    nyquist_freq = 0.2 * Fs
    normal_cutoff = cutoff_freq / nyquist_freq
    b, a = signal.butter(4, normal_cutoff, btype='low', analog=False)
    filtered_data = signal.lfilter(b, a, data)
    return filtered_data

def calculate_dominant_frequency(self, data, fs):
    spectrum = abs(fourier.fft(data))[:FRAMES // 2]
    freqs = np.fft.fftfreq(FRAMES, d=1.0/fs)[:FRAMES // 2]
    dominant_index = np.argmax(spectrum)
    dominant_frequency = freqs[dominant_index]
    return dominant_frequency
```

```
if __name__ == "__main__":
    app = QApplication(sys.argv)
    audio_spectrum_app = AudioSpectrumApp()
    audio_spectrum_app.show()
    sys.exit(app.exec_())
```



¿Desea filtrar la señal?

SI

Tipo de filtro:

Chebyshev

Tipo de filtro:

Paso alto

Frecuencia de corte:

Aplicar filtro

Frecuencia dominante: 527.56 Hz

DICTADO DE VOZ

```
import speech_recognition as sr          #Importamos estas dos librerías para poder
reconocer y sintetizar voz respectivamente
import pyttsx3

def listen_and_save(file_path):          #En estas líneas del programa defino la
función encargada de recibir el archivo de audio, especificando su ruta.
                                          #También se crean dos instancias, una la
instancia Recognizer() de la biblioteca speech_recognition y otra instancia de
init() de la biblioteca pyttsx3

    r = sr.Recognizer()
    engine = pyttsx3.init()

    with sr.Microphone() as source:
        print("Di algo...")
        audio = r.listen(source)

    try:                                  #Dentro de un bloque try-except , utilizo el método
recognize_google() de la instancia r para convertir el audio a texto utilizando
el servicio de reconocimiento de voz de Google.
                                          #Mostramos un mensaje en consola para indicar que se está
procesando el audio y luego imprimimos el texto reconocido.

        print("Procesando...")
        text = r.recognize_google(audio, language="es-ES")
```

```
print("Texto reconocido: ", text)

# Generar archivo de texto con el texto reconocido
with open(file_path, "w") as file:
    file.write(text)
    print(f"Se ha guardado el dictado en {file_path}")
# Reproducir el texto en voz
engine.say(text)
engine.runAndWait()
except sr.UnknownValueError:
    print("No se pudo reconocer el audio")
except sr.RequestError as e:
    print("Error al solicitar el reconocimiento de voz: ", e)

# Especifica la ruta y el nombre del archivo donde deseas guardar el dictado
ruta_archivo = "C:/Users/Usuario/Downloads/DICTADO DE VOZ"

listen_and_save(ruta_archivo)
```

PROGRAMA RECONOCIMIENTO DE LA VOZ

```
import os
import numpy as np
import joblib
import pyaudio
import wavio
import librosa
from sklearn.neural_network import MLPClassifier
import soundfile as sf

def record_audio():
    CHUNK = 1024
    FORMAT = pyaudio.paInt16
    CHANNELS = 1
    RATE = 44100
    RECORD_SECONDS = 5

    p = pyaudio.PyAudio()

    print("Habla para capturar tu voz...")
    frames = []

    stream = p.open(format=FORMAT,
                    channels=CHANNELS,
                    rate=RATE,
                    input=True,
                    frames_per_buffer=CHUNK)

    for _ in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
        data = stream.read(CHUNK)
        frames.append(np.frombuffer(data, dtype=np.int16)) # Convertir datos a
arreglo numpy

    print("¡Grabación finalizada!")
```

```
stream.stop_stream()
stream.close()
p.terminate()

# Guardar el audio en un archivo WAV temporal utilizando wavio
temp_wav_file = "temp_audio.wav"
wavio.write(temp_wav_file, np.concatenate(frames), RATE, sampwidth=2)

return temp_wav_file

def extract_features(audio_file):
    y, sr = sf.read(audio_file)
    mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)
    return np.mean(mfccs, axis=1)

def main():
    print(";Bienvenido! Por favor, introduce tu nombre:")
    user_name = input()

    if os.path.exists(f"{user_name}.pkl"):
        model = joblib.load(f"{user_name}.pkl")
    else:
        print("No se encontraron datos de usuario. Por favor, graba tu voz para
el entrenamiento.")
        features_list = []

        for _ in range(3): # Grabar 3 veces para obtener datos de entrenamiento
            audio_file = record_audio()
            features = extract_features(audio_file)
            features_list.append(features)

        features_array = np.array(features_list)
        labels = np.full(features_array.shape[0], user_name)

        model = MLPClassifier(hidden_layer_sizes=(100,), max_iter=500)
        model.fit(features_array, labels)

        joblib.dump(model, f"{user_name}.pkl")
        print("Datos de entrenamiento guardados.")

    print(";Hola,", user_name, "! Habla para la identificación de voz:")

    while True:
        audio_file = record_audio()
        features = extract_features(audio_file)
        prediction = model.predict([features])
        print("Identificado como:", prediction[0])

        # Eliminar el archivo WAV temporal
```

```
os.remove(audio_file)

if __name__ == "__main__":
    main()

PROGRAMA SEPARACIÓN DE VOCES

from pydub import AudioSegment
import numpy as np
from sklearn.decomposition import NMF

def load_audio(file_path):
    audio = AudioSegment.from_file(file_path)
    return audio

def extract_features(audio):
    samples = np.array(audio.get_array_of_samples())
    return np.abs(samples).reshape(-1, 1) # Valor absoluto y reshape a matriz
bidimensional

def separate_voices(features, num_sources=2, num_iterations=100):
    model = NMF(n_components=num_sources, init='random', random_state=0,
max_iter=num_iterations)
    W = model.fit_transform(features)
    H = model.components_
    return W, H

def save_audio(audio_data, output_path, sample_width=2, channels=1,
frame_rate=44100):
    audio = AudioSegment(
        audio_data.tobytes(),
        frame_rate=frame_rate,
        sample_width=sample_width,
        channels=channels
    )
    audio.export(output_path, format="wav")

def main():
    input_file = r"C:/Users/Alvaro.Gonzalez/Downloads/audio/Kardome_Audio_2021-
05-05_16-48-51_all_mics_BIRDS - Canal derecho posterior surround.wav"
    output_files = ["voz_1.wav", "voz_2.wav"]

    audio = load_audio(input_file)
    features = extract_features(audio)
    sources, _ = separate_voices(features, num_sources=len(output_files))

    for i, output_file in enumerate(output_files):
        save_audio(sources[i], output_file)

if __name__ == "__main__":
    main()
```

4.2 RESULTADOS SOBRE LA SIMULACIÓN DE LA CANCELACIÓN ACTIVA DEL SONIDO

El micrófono captará tres tipos de ruidos: ruido blanco, ruido rosa y ruido marrón. Estos ruidos se distinguen principalmente por la distribución de sus frecuencias y la percepción auditiva resultante. El ruido blanco contiene todas las frecuencias audibles con igual intensidad, creando un sonido uniforme similar al estático de una televisión sin señal. El ruido rosa también abarca todas las frecuencias audibles, pero su intensidad disminuye a medida que aumenta la frecuencia, produciendo un sonido más suave y relajante. Por último, el ruido marrón, o Browniano, enfatiza las frecuencias bajas, menos de 200Hz y reduce las altas, más de 2000 Hz, generando un sonido profundo y grave, similar al trueno o a una cascada.

El ruido blanco se caracteriza por una densidad espectral de potencia constante. En el caso del ruido rosa, se trata de un ruido con menos amplitud para las altas frecuencias; es decir, el sonido decae a medida que aumenta la frecuencia. Es el ejemplo del ruido de la carretera o la lluvia. Por último, los sonidos marrones o sonido de los movimientos brownianos son los característicos de los truenos. (Entendiendo el espectro: explicación del ruido blanco, rosa y marrón, 2024)

El punto de partida es un sistema de reconocimiento de voz y separación de voces donde el hardware ya se encuentra previamente instalado.

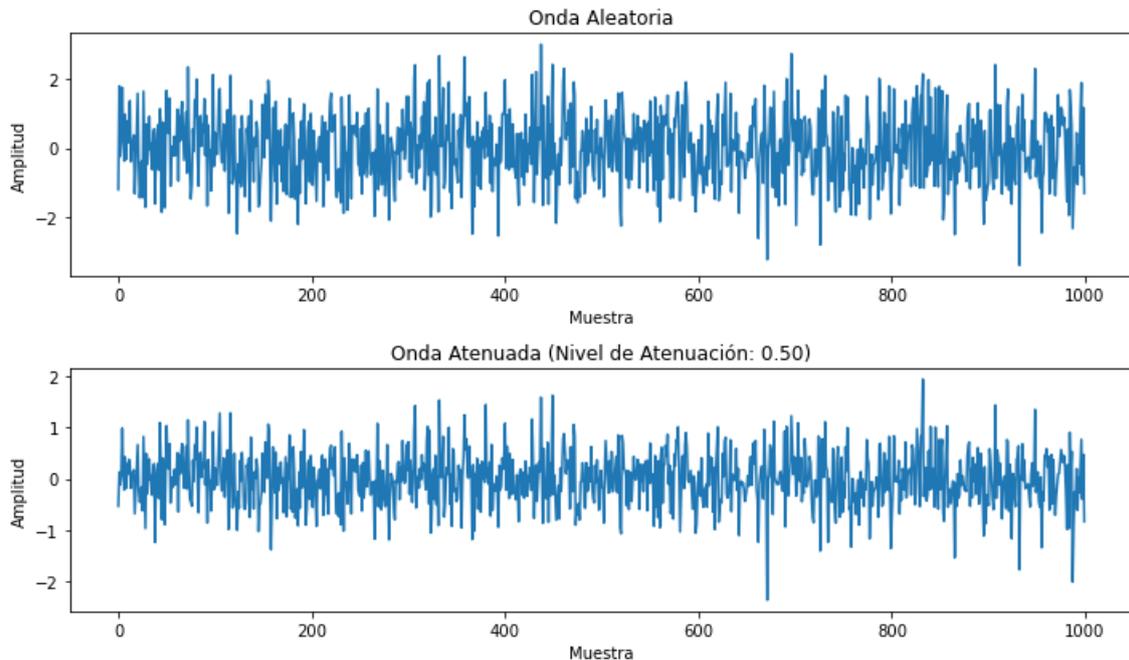


Ilustración 24 Atenuación ruido blanco

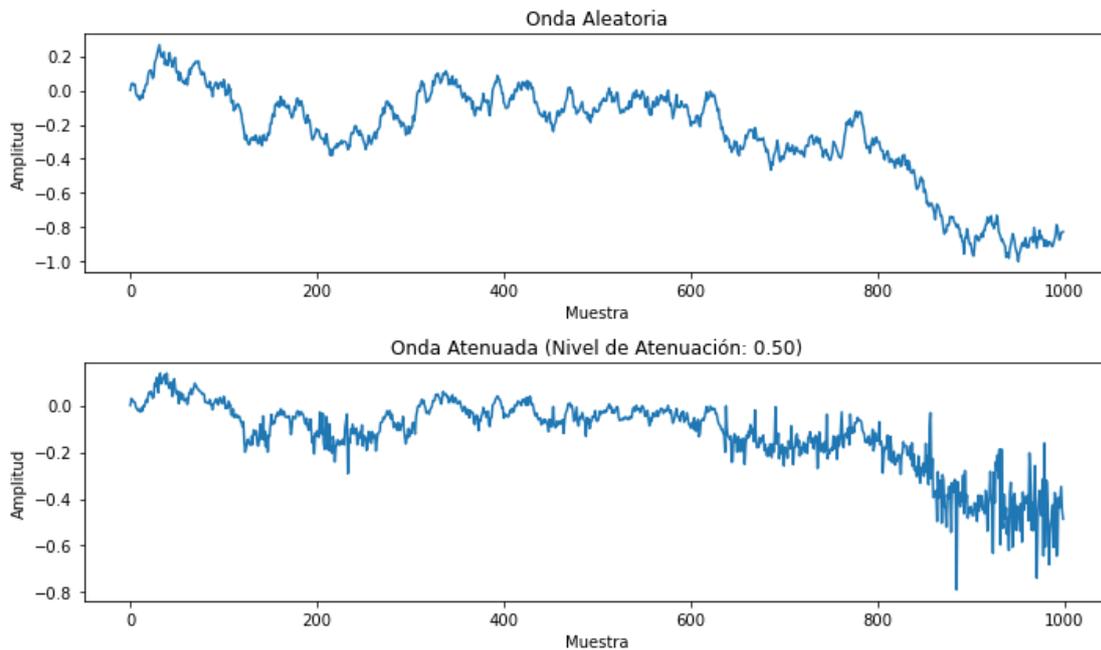


Ilustración 25 Atenuación ruido rosa

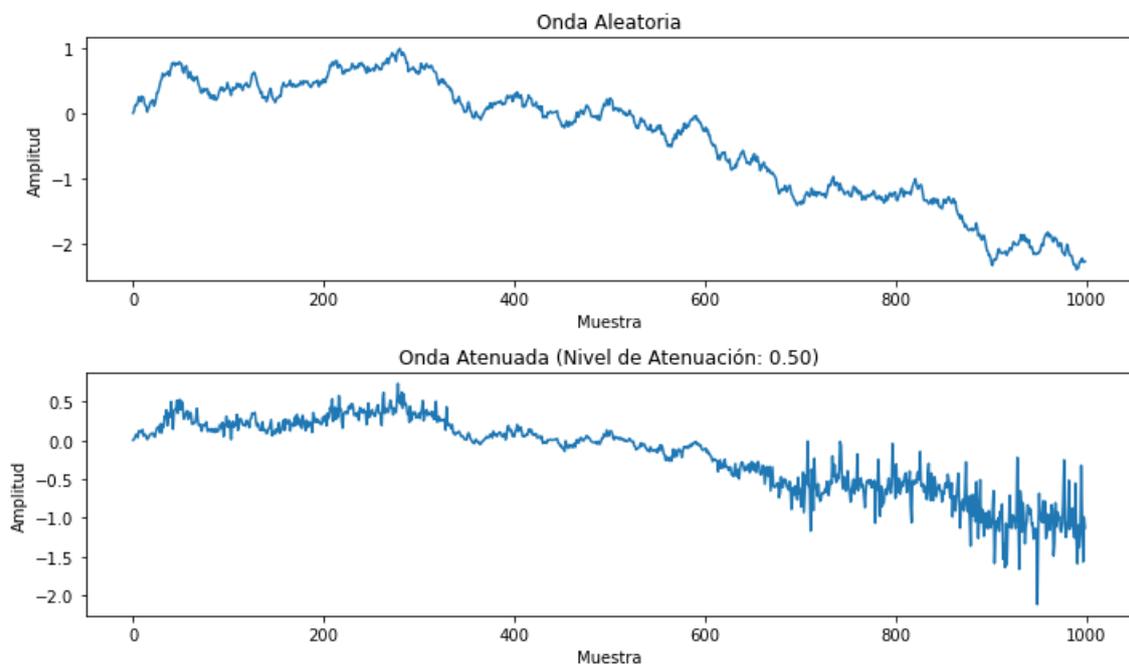


Ilustración 26 Atenuación ruido marrón

Como conclusión se puede afirmar que se han cumplido los objetivos en la simulación. Se pretendía conseguir un programa en el que se generasen ondas aleatorias características de

los posibles ruidos más frecuentes que podemos encontrar en el interior de un automóvil y posteriormente atenuar dichas ondas. En esta simulación se ha atenuado en una escala de 0.5, es decir, las potencias del espectro de audio se han reducido a la mitad. Tal y como se muestra en la , Ilustración 25 Atenuación ruido rosae Ilustración 26 Atenuación ruido marrón De tal forma que por ejemplo, una onda de 20 dB gracias al software planteado se reduce a 10 dB Para ello se ha aplicado la teoría explicada a lo largo del capítulo (ALGORITMOS GENÉTICOS) para abordar el problema y conseguir una correcta atenuación.

4.3 RESULTADOS SOBRE LAS SIMULACIONES DEL RECONOCIMIENTO DE LA VOZ

En el primer programa, el espectro de audio en tiempo real se logran los objetivos. Se esperaba un espectro de audio en tiempo real donde se mostrasen las frecuencias en Hz El programa consigue recoger un espectro de audio en tiempo real y representar la frecuencia de la señal de audio en Hz en el gráfico (eje X) y amplitud (eje Y). Se muestra en la siguiente ilustración:

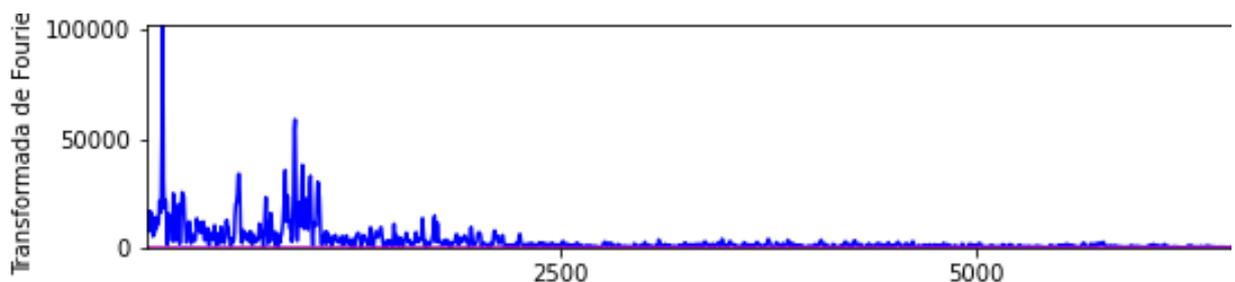


Ilustración 27 Espectro de frecuencias en tiempo real

Frecuencia dominante: 527.56 Hz

Ilustración 28 Frecuencia dominante en tiempo real

Además, también nos muestra la frecuencia dominante o fundamental, parámetro esencial para el reconocimiento de la voz. Este parámetro es el característico de la vibración de las cuerdas vocales, que lo hace fundamental para poder reconocer la voz.

Por otro lado, la simulación cuenta con filtros que se pueden ajustar y disponer como quiera el usuario.

¿Desea filtrar la señal?

Sí

Tipo de filtro:

Chebyshev

Tipo de filtro:

Paso alto

Frecuencia de corte:

Ilustración 29 Mando de opciones del programa

En el segundo programa, el dictado de voz se pretende simular cómo funciona la comunicación usuario computadora. Este código utiliza librerías de Google para transcribir un mensaje de audio, creando un archivo de texto que se almacena en la computadora. En este programa también se cumplen los objetivos, logrando su correcto funcionamiento.

En los últimos programas de reconocimiento de voz y separación de voces, la dificultad para programar el código incrementa. Bien es cierto que la fase de preentrenamiento, en el programa de reconocimiento de la voz el código es capaz de guardar los espectros de voz de cada usuario sin embargo es incapaz de posteriormente poder reconocer a los usuarios de las distintas voces., a la hora de ejecutar dicho código, no se obtiene ningún resultado y no logra reconocer la voz, simplemente no hace nada. Por lo tanto, en este tercer y cuarto código no se cumplen los objetivos ya que no se ha logrado un funcionamiento preciso.

CAPÍTULO 5. ESTUDIO ECONÓMICO

El propósito de este apartado es estimar el coste económico de un sistema de cancelación activa del sonido y reconocimiento de la voz en el interior de los automóviles. Para ello debemos de tener en cuenta varios factores y considerar la instalación de varios componentes. El precio de todos estos componentes ha sido obtenido del portal de venta de dispositivos electrónicos [DigiKey](#).

En la siguiente tabla se recogen los precios de instalación de los elementos necesarios en un único vehículo para conseguir un sistema de cancelación activa de sonido y reconocimiento biométrico de la voz.

	€/u	TOTAL €
Micrófonos MEMS (16u)	5	80
Altavoces (16u)	5	80
Filtro FIR	5	5
Unidad de control electrónico (1u)	10	10
Sensores y acelerómetros (20u)	3	60
Unidad de procesamiento (1u)	20	20
Costes de equipo y maquinaria		200
Equipo de ingenieros electrónicos		300

Tabla 2 Tabla de precios de instalación de los elementos necesarios

Al calcular el precio de la instalación de estos sistemas en un único vehículo obtenemos la cifra de 755€. Cifra que únicamente cubre los costes de personal y compra de componentes para cada vehículo. A este precio debemos de sumarle el beneficio que obtendrá la empresa por la instalación de ambos sistemas y el precio del software.

En la tabla a continuación se recoge el precio de licencias y desarrollo de software:

Software + licencia para un sistema de cancelación activa de sonido	100.000.000€
Software + licencia para un sistema de reconocimiento de voz	100.000.000€

Tabla 3 Tabla de precios de licencias y desarrollo de software

A estos precios también se le debe de añadir el beneficio que obtendrá la empresa por instalar estos sistemas en el interior de los vehículos que será de 50€ por coche producido. De esta forma, si se espera amortizar la compra de los softwares y licencias en un tiempo de 10 años, y contando con que una empresa líder del sector como Honda, Ford, Nissan o Hyundai venden en torno a 5 millones de vehículos al año, de los cuales se podría estimar que tan solo el 10% de los compradores (500.000 vehículos) equiparía su vehículo con esta tecnología, el precio de ambos softwares por vehículo debería de ser de 40€ para conseguir amortizarlo en 10 años.

Finalmente sumando todos los costes de instalación más los del software y licencia, e incluyendo el beneficio que obtiene la empresa, se obtiene un precio total de en torno a 845€ (en esta cifra solo vienen dados los gastos técnicos de la empresas, sin impuestos) Se debe tener en cuenta que el gasto medio de un español a la hora de comprar un vehículo es de 25.000€. (Gómara, 2024) Es decir que equipar un vehículo con los sistemas que se desarrollaran en este documento supondrían un incremento del precio de un vehículo medio en España. Precio que poca parte de la población estaría dispuesta a pagar. Por este motivo estos sistemas hoy en día están orientados a vehículos de alta gama o tienen una mayor comercialización en países como Arabia o China.

CAPÍTULO 6. CONCLUSION

En el presente documento se ha abordado la implementación de tecnologías avanzadas de cancelación activa de sonido (ANC) y reconocimiento de voz en el interior de automóviles, analizando tanto los aspectos técnicos como los económicos asociados a su implementación.

En este capítulo se busca repasar los objetivos citados al principio del documento y ver si han sido cumplidos a lo largo del proyecto.

En primer lugar, se han abordado todos los aspectos técnicos relacionados con la regulación automática, con relación a los algoritmos adaptativos y los filtros de los sistemas de cancelación activa de sonido, el procesamiento de señales, en temas como la transformada de Fourier o los filtros Mel del reconocimiento de la voz, la física de ondas o la electrónica, en temas como el hardware de los sistemas. Además, se han explicado mediante fórmulas y teoremas matemáticos. De esta forma se ha conseguido un claro entendimiento de ambos temas. Por otro lado, se ha desarrollado un estado del arte, lo que hay desarrollado de ambos temas, las empresas líderes de mercado y en que se diferencian unas de otras. Además, se han explicado distintos sistemas y se han mencionado sistemas ya desarrollados por compañías. Este objetivo queda cumplido por lo expuesto en el capítulo 2

En segundo lugar, se han seleccionado una serie de elementos para confeccionar el sistema y se han dispuesto en la mejor ubicación posible dentro del vehículo para hacer lo más fácil posible la tarea del sistema de cancelación activa del sonido y reconocimiento de la voz.

El punto más importante de este proyecto han sido las simulaciones de ambos sistemas, a través de la teoría explicada a lo largo del documento. Mediante el lenguaje de programación de Python se han desarrollado varios códigos con el fin de simular sistemas de cancelación activa de sonido y de reconocimiento de la voz. Además, se han explicado y mostrado su funcionamiento con imágenes. Bien es cierto, que dos de los programas (reconocimiento de la voz y separación de voces) no se ha logrado un funcionamiento preciso.

Por último, se ha llevado a cabo un pequeño análisis económico, justificando los precios del sistema y el coste que tendría su instalación en un vehículo. Se ha llegado a la conclusión de que a día de hoy se trata de una tecnología destinada a la alta gama, sin embargo, con el tiempo, con nuevos avances y al amortizarse los gastos de desarrollos podrá popularizarse. Este objetivo queda cumplido por lo expuesto en el capítulo 5.

En conclusión, la integración de sistemas de cancelación activa de sonido y reconocimiento de voz en el interior de los automóviles representa una evolución natural hacia vehículos más confortables, seguros y tecnológicamente avanzados. A medida que estas tecnologías continúan desarrollándose y los costos asociados disminuyen, es previsible que su adopción se generalice, transformando la experiencia de conducción y consolidando su posición en la industria automotriz. La inversión en estas innovaciones, por tanto, no solo es una apuesta por la mejora del producto, sino también por la sostenibilidad y la satisfacción del consumidor a largo plazo.

CAPÍTULO 7. BIBLIOGRAFÍA

(CNXSOFT), J.-L. A. (2022). *El SoC Allwinner T113-S3 de doble núcleo Cortex-A7 cuenta con 128 MB de DDR3 y está destinado a aplicaciones automotrices e industriales.*

Academialab. (2024). *Transformada de Fourier.*

Alonso, R. (2021). *Así logra la Inteligencia Artificial mejorar la cancelación de ruido.*

Antolín. (2023). *Antolin y Biometric Vox colaboran para un acceso seguro y cómodo al vehículo mediante voz.*

(2024). *ASR en AWE .*

Batista, M. B. (2009). *Algoritmos Genéticos. Una visión práctica.*

Beléndez, A. (2015). Thomas Young y la naturaleza ondulatoria de la luz. *BBVA Open Mind*, 1.

Beléndez, A. (2015). *Thomas Young y la naturaleza ondulatoria de la luz.*

Cancelación Activa de Ruido. (2020). *Electrónica, Electricidad y Telecomunicaciones.*

Chávez, J. M. (2005). *IMPLEMENTACIÓN DE FILTROS DIGITALES TIPO FIR EN FPGA'S CON COEFICIENTES RECONFIGURABLES.*

(2024). *Entendiendo el espectro: explicación del ruido blanco, rosa y marrón.*

Garcia, A. (2023). *Bose Trabaja en la Cancelación de Ruido de la Suspensión y el Ruido de los Neumáticos.*

Garcia, G. (2019). *Cancelación activa de ruido en los coches eléctricos, así es la tecnología RANC de Hyundai.* Madrid.

GeneratePress. (2024). *Guías Prácticas*.

Gómara, J. (2024). *Esto es lo que se gastó de media un español en comprarse un coche nuevo el año pasado, menos de lo esperado*.

Hansen, R. (2023). ¿Qué es un micrófono de condensador? *SoundSCAPEHQ*.

Herrera, J. (2023). MEMS: qué es esta tecnología y dónde se aplica. *Guía Hardware*.

Ibarrola, D. J. (2020). *Procesamiento Digital de Señales de Voz*.

iFlytek. (s.f.). *Create a better word with AI*.

L., F. R. (2014). *ALGORITMOS LMS DE FILTRADO*.

L., F. R. (2014). *ALGORITMOS LMS DE FILTRADO ADAPTATIVO PARA CANCELACIÓN DE ECO ACÚSTICO EN SISTEMAS DE TELECOMUNICACIONES*.

(2018). *La evolución de las interfaces de comunicación hombre-máquina: escribir, tocar, hablar... pensar*.

Labra, S. F. (2018). *API de Reconocimiento de Voz para Lenguaje*.

Lavinia. (2024). *Algoritmo de mínimos cuadrados medios (Algoritmo LMS)*.

LTD, K. T. (2023).

Ltd, P. C. (2021). *Silentium Active Acoustics on road noise*.

Martín, C. F. (2009). *Criterio de estabilidad de Nyquist*.

(2021). *Metodo de La Maxima Pendiente*. Library.

Mosquera, H. (2014). *FILTROS DIGITALES FIR*.

Neves, F. (2014). *Algoritmos DSP con a placa STM32F4Discovery*.

PNGWING. (s.f.). *Escala Mel Freqüência Sonora Passo Cepstrum*.

POLARIDAD.ES. (s.f.). *IIR: Qué es y cómo funciona el filtro de respuesta infinita al impulso*.

Ramírez, F. (2018). *Historia de la IA: Frank Rosenblatt y el Mark I Perceptrón, el primer ordenador fabricado específicamente para crear redes neuronales en 1957*.

RECALM. (2024). *RECALM ANCOR INFOTAINMENT-SYSTEM*.

Romero, M. S. (2020). El algoritmo que ayudará a viajar en coche con niños: burbujas de sonido delimitadas a cada asiento. *EL ESPAÑOL*.

Romero, M. S. (2022). *El algoritmo que ayudará a viajar en coche con niños: burbujas de sonido delimitadas a cada asiento*.

Rufiner, H. L. (2004). *Sistema de reconocimiento automático del habla*.

Salcedo, J. A. (2019). *Tratamiento digital de señales analógicas*.

Sánchez, J. (s.f.). Interferencias de ondas. *El físico loco*, 1.

Villanueva, A. F. (2020). *Reconocimiento de vocalistas mediante redes neuronales profundas*.