Deconvolución acústica basada en métodos de aprendizaje profundo y filtros adaptativos no lineales

Pablo Giovanni Maldonado Alvarez

UVG | UNIVERSIDAD DEL VALLE DE GUATEMALA

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Deconvolución acústica basada en métodos de aprendizaje profundo y filtros adaptativos no lineales

Trabajo de graduación presentado por Pablo Giovanni Maldonado Alvarez para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2025

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Deconvolución acústica basada en métodos de aprendizaje profundo y filtros adaptativos no lineales

Trabajo de graduación presentado por Pablo Giovanni Maldonado Alvarez para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2025

Vo.Bo.:

(f) 

M.Sc. Miguel Enrique Zea Arenales

(f) 

M.Sc. Carlos Alberto Esquit Hernández

Fecha de aprobación: Guatemala, 28 de noviembre de 2025.

Prefacio

Primeramente quiero agradecer a Dios por los sueños que me ha dado y el camino que juntos hemos recorrido hasta este momento. Gracias por tu infinito amor, por la paz en momentos adversos y por las puertas que has abierto, en verdad me sobran razones para decir que estoy bendecido.

Agradezco a mis padres, Sara Alvarez y Giovanni Maldonado, por su apoyo incondicional a lo largo de toda mi vida y por el esfuerzo que han hecho para ayudarme a conquistar cada una de mis metas. Gracias por sus consejos, por cuidar de mi con sus oraciones y por sus palabras de motivación. Gracias por la confianza que han puesto en mí, los amo.

A mi hermana, Daniela Maldonado, gracias por el amor que todos los días me muestras de tantas formas y por llenar mi vida de alegría. Espero verte crecer y ayudarte a cumplir todos tus sueños. Sé que volarás muy alto.

A los amigos que conocí en la universidad, gracias por todas las risas, apoyo y momentos que compartimos. Sin duda alguna han sido de los mejores años de mi vida y eso es gracias a ustedes, los llevo en el corazón y sé que seguiremos alcanzando más éxitos.

A mi asesor, M.Sc. Miguel Zea, por compartir conmigo su conocimiento, su indispensable retroalimentación y valiosa guía para el desarrollo de este proyecto. Gracias por su disposición y constante ayuda para materializar este trabajo.

A esta casa de estudios y a sus catedráticos por cada una de sus valiosas enseñanzas que han impactado mi vida, tanto en lo profesional como en lo personal. Gracias por su orientación, apoyo y comprensión para llegar a este momento.

A todos ustedes, gracias por haber contribuido de una u otra forma a alcanzar esta meta. Este logro es nuestro.

Índice general

Índice de figuras

Índice de cuadros

Resumen

La deconvolución acústica busca recuperar señales afectadas por las reflexiones del recinto en entornos sin tratamiento acústico. Este estudio evaluó tres enfoques representativos para esta tarea mediante una base de datos unificada de señales limpias y contaminadas, junto con un flujo de trabajo reproducible para su comparación. Se consideraron filtros adaptativos no lineales basados en *kernel*, modelos autorregresivos no lineales con entradas exógenas (NLARX) y redes neuronales profundas de tipo red convolucional temporal (*temporal convolutional network*, TCN).

La evaluación incluyó señales sintéticas y grabaciones reales de voz e instrumentos musicales, utilizando el error cuadrático medio (RMSE) y la relación señal-ruido (SNR) como métricas de desempeño. Los métodos basados en *kernel* no lograron una reconstrucción efectiva en señales reales, presentando valores de SNR cercanos a 0 dB. El modelo NLARX obtuvo un RMSE cercano a 0.32 en señales sintéticas, pero en señales reales produjo valores de SNR entre -9 dB y -24 dB debido a un comportamiento equivalente al de un filtro pasa bajas. En contraste, la TCN presentó el mejor desempeño global, con un RMSE cercano a 0.17 en señales sintéticas y un SNR de -0.08 dB en señales reales, preservando componentes de alta frecuencia que los otros métodos no lograron recuperar.

Los resultados posicionan a las arquitecturas profundas como el enfoque más adecuado para avanzar en la deconvolución acústica.

Palabras clave: deconvolución acústica, aprendizaje automático, filtros adaptativos, modelos no lineales, redes neuronales profundas.

Abstract

Acoustic deconvolution aims to recover signals degraded by room reflections in environments without acoustic treatment. This study evaluated three representative approaches to this problem using a unified database of clean and contaminated signals, together with a reproducible workflow for their comparison. The evaluated methods included kernel adaptive filtering (KAF), nonlinear autoregressive models with exogenous inputs (NLARX), and deep neural networks of the temporal convolutional neural network (TCN) type.

The evaluation considered both synthetic signals and real recordings of voice and musical instruments. Performance was assessed using root mean square error (RMSE) and signal-to-noise ratio (SNR). The KAF method did not achieve effective reconstruction in real signals, yielding SNR values close to 0 dB. The NLARX model achieved an RMSE close to 0.32 with synthetic signals, but for real recordings it produced SNR values between -9 dB and -24 dB due to a low-pass filtering effect. In contrast, the TCN achieved the most consistent performance, with an RMSE close to 0.17 for synthetic signals and an SNR of -0.08 dB for real recordings, preserving high-frequency components that the other methods failed to recover.

These results indicate that deep learning architectures constitute the most promising direction for advancing acoustic deconvolution.

Keywords: acoustic deconvolution, machine learning, adaptive filters, nonlinear models, deep neural networks.

CAPÍTULO 1

Introducción

La calidad de una grabación de audio realizada en entornos sin tratamiento acústico se ve afectada por reflexiones y reverberaciones que alteran la señal original. Este fenómeno plantea dificultades para tareas de análisis, restauración o procesamiento posterior, y ha motivado diversas investigaciones en la Universidad del Valle de Guatemala desde 2020. A lo largo de esta línea de investigación se han explorado múltiples enfoques, desde filtros adaptativos lineales y no lineales hasta modelos basados en redes neuronales. Sin embargo, los estudios previos emplearon metodologías distintas y evaluaron sus resultados bajo condiciones heterogéneas, lo cual impedía una comparación clara y una dirección metodológica unificada.

El presente trabajo se desarrolló con el propósito de consolidar esta línea de investigación y determinar, con evidencia experimental, qué enfoque resulta más adecuado para la deconvolución acústica en entornos no tratados. Para ello, se estableció un marco metodológico común que incluyó la integración de una base de datos estandarizada, la preparación sistemática de los datos y la evaluación de tres enfoques representativos: filtros adaptativos no lineales basados en *kernel*, modelos autorregresivos no lineales con entradas exógenas y redes neuronales profundas. Esta comparación permitió analizar su estabilidad, capacidad de reconstrucción y comportamiento frente a señales reales y sintéticas.

Los resultados obtenidos mostraron diferencias significativas entre los enfoques evaluados y evidenciaron que los métodos basados en redes neuronales profundas ofrecen una mayor capacidad para modelar relaciones no lineales complejas y recuperar señales degradadas. A partir de estos hallazgos, el trabajo identifica la dirección metodológica más pertinente para el avance de esta línea de investigación, destacando la conveniencia de orientar futuras iteraciones hacia modelos neuronales de mayor capacidad.

CAPÍTULO 2

Antecedentes

El problema de deconvolución acústica para señales de audio ha sido abordado en la Universidad del Valle de Guatemala mediante una línea de investigación centrada en desarrollar métodos capaces de reducir o eliminar los efectos de reverberación en entornos sin tratamiento acústico. Esta línea ha tenido hasta la fecha tres iteraciones consecutivas, cada una de ellas aportando soluciones, limitaciones y recomendaciones que han permitido trazar el camino para nuevas fases.

En la primera iteración, Méndez [1] centró sus esfuerzos en explorar distintos métodos de deconvolución de señales de audio, incluyendo funciones de deconvolución convencionales de *MATLAB*, técnicas basadas en el dominio de la frecuencia, identificación de sistemas y, finalmente, filtros adaptativos lineales y cuadráticos. Tras realizar una serie de pruebas se concluyó que los métodos clásicos no ofrecían una solución efectiva al problema, ya que en varios casos el error de reconstrucción superaba incluso al de la señal perturbada. En cambio, los filtros adaptativos especialmente el filtro LMS lineal y su versión cuadrática, mostraron resultados prometedores logrando una reconstrucción más fiel de la señal original y una reducción considerable del error cuadrático medio. Sin embargo, se identificó una limitación importante, la cual fue que estos métodos requerían una señal de referencia limpia en todo momento para su correcto funcionamiento, lo cual no siempre es viable en aplicaciones en entornos reales. Esta observación motivó la recomendación de explorar el uso de redes neuronales para enfrentar los desafíos asociados a la no linealidad y variabilidad temporal del entorno acústico.

La segunda iteración, desarrollada por López [2], dio el siguiente paso lógico al comparar directamente filtros adaptativos con redes neuronales regresivas. Se implementaron tres variantes de filtros adaptativos: {LMS estándar, LMS normalizado y RLS}. Y se implementaron dos arquitecturas neuronales: redes TDNN y Focused Gamma, cada una con distintas funciones de activación. Las pruebas incluyeron tanto señales determinísticas como clips musicales con distintos niveles de complejidad. Los resultados indicaron que el filtro RLS convencional ofrecía la mayor fidelidad en la reconstrucción de las señales, alcanzando un error prácticamente nulo en algunos casos. Si bien las redes neuronales lograron una

atenuación parcial del ruido de fondo, introdujeron distorsiones que afectaron la calidad del audio reconstruido. Aun así, este trabajo sentó las bases para futuras mejoras, al proponer la evaluación de arquitecturas neuronales con mayor capacidad de aprendizaje secuencial y extracción de características, como las redes neuronales recurrentes (RNN) y las redes neuronales convolucionales (CNN).

En la tercera iteración, Calí [3] dio continuidad a la línea de investigación mediante la implementación de técnicas de aprendizaje profundo, evaluando arquitecturas como RNN, CNN y redes convolucionales temporales (TCN). A pesar de que el tratamiento de las TCN fue limitado y explorado de forma tardía, sus resultados iniciales fueron prometedores, evidenciando una capacidad notable para modelar dependencias temporales y adaptarse a señales acústicas complejas. Estos hallazgos, aunque preliminares, posicionaron a las TCN como candidatas relevantes para una investigación más profunda. En general, los modelos se entrenaron utilizando una base de datos con grabaciones en ambientes con y sin tratamiento acústico, y fueron evaluados mediante métricas como el RMSE, una medida estadística que indica qué tan cerca están las predicciones de un modelo con respecto a los valores reales. Esta se calcula como la raíz cuadrada del promedio de los errores al cuadrado, lo que permite cuantificar la precisión del modelo: cuanto menor es el RMSE, mejor es el ajuste del modelo a los datos. Las redes RNN y CNN también ofrecieron resultados interesantes, especialmente en la extracción automática de características, pero el enfoque limitado hacia las TCN dejó abierta la oportunidad de investigar su potencial con mayor rigurosidad. Asimismo, esta iteración remarcó la importancia de contar con datos bien estructurados y grabaciones de referencia con baja reverberación, recomendando la estandarización de las bases de datos para futuras investigaciones.

Cada una de estas etapas ha permitido refinar tanto el enfoque metodológico como las herramientas utilizadas para abordar el problema de la deconvolución acústica. La evolución desde métodos clásicos basados en filtros hasta arquitecturas de aprendizaje profundo ha demostrado que, si bien se han logrado avances significativos, persisten desafíos asociados a la variabilidad de los entornos acústicos y a las características no lineales de las señales.

CAPÍTULO 3

Justificación

El avance de las redes sociales y plataformas de *streaming* ha permitido que artistas emergentes tengan la posibilidad de compartir sus composiciones desde cualquier lugar en el que se encuentren, sin necesidad de pertenecer a una disquera para poder alcanzar la fama a nivel global. Pese a estos avances, aún existen limitaciones que estos artistas deben superar para lograr ser competitivos en este mercado, siendo el enfoque de este trabajo la calidad de audio que estos pueden obtener.

Para resolver este inconveniente, lo ideal es trabajar en un cuarto tratado acústicamente, ya que en este se controlan los efectos de las reflexiones y las reverberaciones. No solo esto, sino también se debe considerar el equipo de grabación que el músico emplea. Crear un espacio con estas características necesita de una inversión significativa y rentarlos puede ser una limitante. Por tales motivos, este proyecto está enfocado en la deconvolución acústica a través de algoritmos de aprendizaje automático para abordar esta problemática mediante soluciones basadas en software.

En iteraciones previas de este proyecto se demostró que los filtros adaptativos eran una alternativa viable para abordar la deconvolución acústica. No obstante, debido a su naturaleza lineal, su desempeño se vio limitado frente a entornos con comportamiento no lineal. Esta limitación motivó la exploración de arquitecturas basadas en redes neuronales, particularmente redes convolucionales, las cuales mostraron una mayor capacidad de generalización y adaptación a entornos musicales diversos.

Con base en esos resultados, el presente trabajo se enfoca en profundizar en la comparación entre métodos de aprendizaje automático modernos, específicamente los filtros adaptativos no lineales como el *Kernel Adaptive Filtering*, los modelos auto-regresivos no lineales con entrada exógena y las redes neuronales con arquitectura TCN. Estos modelos han mostrado avances significativos en tareas de procesamiento de señales, gracias a su capacidad para capturar dependencias temporales y características no lineales. Esta comparación permitirá identificar el enfoque más adecuado para aplicaciones de grabación en entornos no tratados, contribuyendo al desarrollo de herramientas accesibles para músicos independientes.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Evaluar y experimentar con distintos métodos de aprendizaje automático para realizar deconvolución acústica de señales de audio grabadas en entornos con condiciones acústicas no controladas.

4.2. Objetivos específicos

- Recopilar y organizar las muestras de audio provenientes de investigaciones previas para crear una base de datos consolidada que facilite el entrenamiento y la evaluación de modelos de aprendizaje.
- Establecer un *pipeline* con las herramientas de *software* adecuadas para el diseño, implementación y entrenamiento de los modelos de aprendizaje para la deconvolución acústica.
- Comparar los métodos de *Kernel Adaptive Filtering* y Redes Neuronales Convolucionales Temporales para tareas de deconvolución acústica.

CAPÍTULO 5

Alcance

El alcance de este trabajo consistió en evaluar tres enfoques representativos para la deconvolución acústica en entornos sin tratamiento: filtros adaptativos no lineales basados en *kernel* (KAF), modelos autorregresivos no lineales con entradas exógenas (NLARX) y redes neuronales profundas de tipo *temporal convolutional network* (TCN). Más que comparar su rendimiento absoluto, el propósito central fue establecer una base metodológica coherente que orientara las siguientes fases de esta línea de investigación.

El proyecto se desarrolló utilizando exclusivamente una base de datos previamente existente, integrada por grabaciones limpias y contaminadas obtenidas en trabajos anteriores. No se generaron nuevas grabaciones, por lo que no fue posible controlar las condiciones acústicas ni la variabilidad de las señales. Esta dependencia de datos históricos constituyó una limitación importante, especialmente en escenarios que habrían requerido una mayor diversidad o un mayor control experimental.

El trabajo también estuvo condicionado por una curva de aprendizaje considerable, derivada de la necesidad de revisar en profundidad los fundamentos teóricos asociados a KAF, NLARX y TCN antes de implementar los experimentos. Este requisito metodológico implicó dedicar parte del esfuerzo a comprender los modelos y sus supuestos, lo que redujo el alcance práctico de las pruebas realizables dentro del tiempo disponible.

A pesar de estas limitaciones, el estudio permitió estandarizar las señales disponibles, establecer un flujo de trabajo reproducible para preparar los datos, estimar los modelos y evaluar su desempeño, y analizar las capacidades y limitaciones de cada enfoque con base en evidencia experimental.

Quedaron fuera del alcance el diseño de nuevas arquitecturas profundas, la obtención de grabaciones adicionales y la estimación detallada de parámetros acústicos del recinto. Con los resultados obtenidos, el estudio permitió definir que las futuras iteraciones de esta línea de investigación deberían centrarse en modelos neuronales de mayor capacidad, sustentándose en la base de datos unificada y estandarizada y en el marco metodológico consolidados en este trabajo.

CAPÍTULO 6

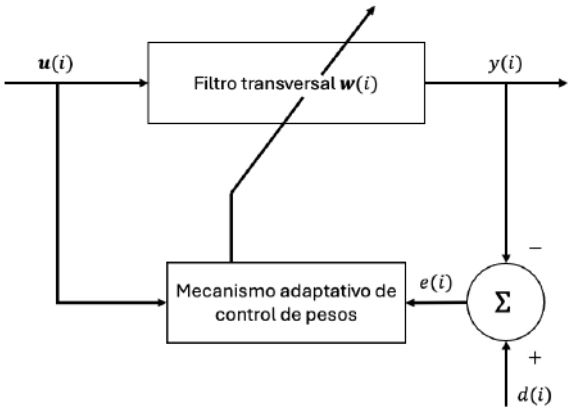
Marco teórico

En este capítulo se presenta el marco teórico que sustenta los modelos empleados en este trabajo para la deconvolución acústica de señales de audio. Dado que el problema puede formularse como la identificación de la relación entrada–salida de un sistema dinámico que incluye la respuesta impulsiva del recinto, se revisan primero los filtros adaptativos lineales y su extensión no lineal, que constituyen un punto de partida clásico para el ajuste iterativo de modelos a partir del error. A continuación se introducen los espacios de Hilbert con *kernel* de reproducción y los *kernel adaptive filters* (KAF), que permiten incorporar no linealidad mediante núcleos sin abandonar el marco de los filtros adaptativos. Sobre esta base se aborda el aprendizaje profundo mediante redes neuronales y sus detalles de entrenamiento, para finalmente discutir arquitecturas específicas orientadas a secuencias, como las redes convolucionales temporales (*temporal convolutional networks*, TCN), y los modelos auto-regresivos no lineales con entradas exógenas (NLARX). En conjunto, estos conceptos definen el conjunto de herramientas de modelado utilizado en este estudio para aproximar el comportamiento del recinto y recuperar señales de audio degradadas.

6.1. Filtros adaptativos lineales

Un filtro adaptativo es un tipo de filtro que dentro de su estructura cuenta con un mecanismo capaz de ajustar automáticamente sus parámetros libres en respuesta a las variaciones estadísticas del entorno en el que opera. Para esto, el filtro recibe una señal de entrada discreta $u(i)$ y la procesa mediante un conjunto de parámetros ajustables (pesos) $\mathbf{w}(i)$ para generar la señal de salida $y(i)$. La salida se compara con la referencia deseada $d(i)$ para obtener el error $e(i) = d(i) - y(i)$. Este error se utiliza para actualizar el vector de pesos, de manera que la salida se aproxime cada vez más a la señal deseada, como se muestra en la Figura 1 [4] [5].

Figura 1. Estructura básica de un filtro adaptativo lineal



Nota. Elaboración propia.

6.1.1. Algoritmo *Least-Mean-Squares*

La forma más simple y ampliamente utilizada de los algoritmos de filtros adaptativos lineales es el *least-mean-squares* (LMS). Este algoritmo opera minimizando la función de costo instantánea (1) para encontrar el vector de pesos $\mathbf{w}(i)$ mediante el cálculo del vector gradiente instantáneo [4].

$$J(i) = \frac{1}{2}e^2(i), \tag{1}$$

Este método cuenta con un parámetro de paso η el cual regula la convergencia del modelo. Y, por ende, controla la tasa de adaptación del mismo como se muestra en (2).

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta e(i)\mathbf{u}(i), \tag{2}$$

6.2. Filtros adaptativos no lineales

Un filtro adaptativo no lineal es un sistema que aprende, en línea, una relación entrada-salida arbitraria de la forma (3)

$$y(i) = f_i(u(i)), \tag{3}$$

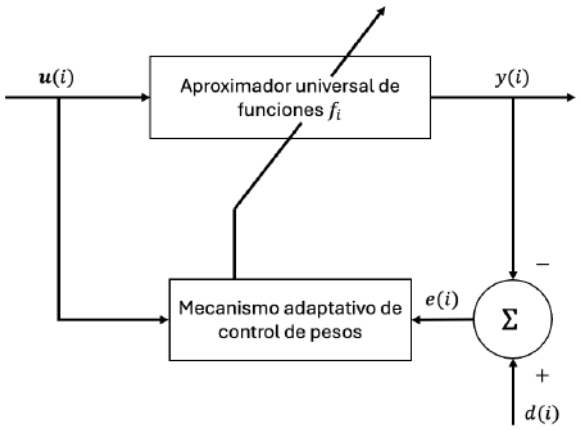
donde $u(i)$ es la señal de entrada en el instante discreto i, $y(i)$ la salida generada por el filtro y f_i la aproximación no lineal hasta ese instante. El aprendizaje es secuencial por corrección de error, donde la salida se compara con una referencia deseada $d(i)$ y se calcula $e(i) = d(i) - y(i)$. Con base en este error, el filtro actualiza su estimador funcional siguiendo una regla incremental general, descrita por la ecuación (4) [4]:

$$f_i \;=\; f_{i-1} \;+\; \mathrm{Gain}(i)\, e(i), \tag{4}$$

La ganancia $\mathrm{Gain}(i)$ es un operador dependiente de los datos que determina cuánto y en qué dirección se corrige f_{i-1}. Puede ser un escalar (tamaño de paso), un vector/matriz

o un operador en un espacio de funciones (espacios de Hilbert con *kernel* de reproducción). La estimación se compone de dos partes aditivas: (i) la mejor estimación previa f_{i-1} y (ii) un término de corrección proporcional al error de predicción con los datos nuevos. Esta naturaleza incremental permite operar en tiempo real [4].

Figura 2. Estructura básica de un filtro adaptativo no lineal



Nota. Elaboración propia.

Como se muestra en la Figura 2, esta arquitectura capta las no linealidades de la señal manteniendo un esquema de aprendizaje sencillo y estable [4].

6.3. Espacios de Hilbert con *kernel* de reproducción (RKHS)

Un espacio de Hilbert con *kernel* reproductor (RKHS) es un espacio de funciones $\mathcal{H}$ dotado de un producto interno tal que existe un *kernel* $k(\cdot,\cdot)$ con la propiedad de reproducción (5) [6]. Es decir, un punto de entrada $\mathbf{u}$ en el espacio de entrada se transforma en el punto $\varphi(\mathbf{u})$ dentro de un espacio de características de mayor dimensión, como se representa en la Figura 3 [4].

$$f(x) \;=\; \langle f,\, k(x,\cdot)\rangle_{\mathcal{H}}, \quad \forall\, f \in \mathcal{H}. \tag{5}$$

Equivalente a esto, toda función del espacio puede escribirse como combinación lineal de "secciones" del *kernel* (6):

$$f(\cdot) \;=\; \sum_{i=1}^{n} \alpha_i\, k(x_i,\cdot). \tag{6}$$

Estas dos ideas, reproducción y representación por *kernels*, caracterizan al RKHS.

La conexión con "características" (*features*) aparece vía *kernels* de Mercer: para los núcleos correctos existe un mapeo (posiblemente infinito-dimensional) ϕ hacia un espacio de Hilbert tal que

$$k(x,y) \;=\; \langle \phi(x), \phi(y)\rangle, \tag{7}$$

es decir, el *kernel* actúa como producto interno en dicho espacio, como se representa en la ecuación (7) [6].

Una consecuencia crucial es el teorema del *representer*: al minimizar una función de pérdida regularizada sobre datos $\{(x_i, y_i)\}_{i=1}^{n}$, la solución óptima tiene siempre la forma (8)

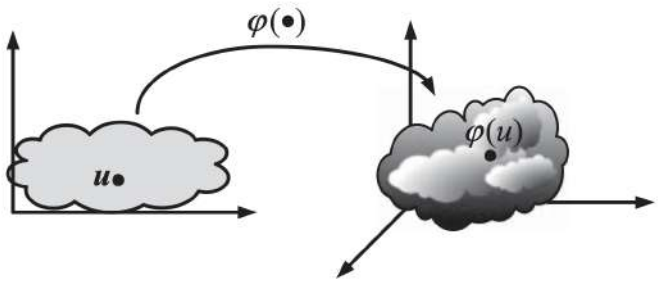
$$f^{\star}(\cdot) \;=\; \sum_{i=1}^{n} \alpha_i \, k(x_i, \cdot), \tag{8}$$

lo que reduce un problema potencialmente infinito-dimensional a n coeficientes [6].

Trabajar en un RKHS permite usar algoritmos lineales adaptativos (LMS, RLS, etc.) en el espacio de características, donde todo es lineal, y obtener filtros no lineales en el espacio de entrada por medio de evaluaciones del *kernel* ("*kernel trick*"). Esto ofrece capacidad de aproximación universal, optimización convexa (sin mínimos locales) y complejidad razonable frente a alternativas como redes neuronales [6].

Además, en RKHS se justifican propiedades clave: (i) la aproximación universal (p. ej., con *kernel* Gaussiano) y (ii) la forma expandida sobre puntos de entrenamiento dada por el *representer*, que fundamenta diseños como KLMS/KRLS [4] [7].

Figura 3. Mapeo no lineal $\varphi(\cdot)$ del espacio de entradas al espacio de características



Nota. Esta imagen fue obtenida de [4].

6.4. *Kernel adaptive filters*

Los *Kernel Adaptive Filters* (KAF) son filtros adaptativos no paramétricos que proyectan los datos de entrada a un espacio de características de alta dimensión mediante un *kernel* de RKHS. En ese espacio, las operaciones de producto interno se calculan de forma eficiente a través de evaluaciones de *kernel* ("*kernel trick*"), por lo que no es necesario construir el espacio explícitamente [6].

Sobre dicha representación se aplican métodos lineales de adaptación (p. ej., LMS/RLS) para ajustar el modelo en línea. El RKHS subyacente aporta una formulación lineal y típicamente convexa en los parámetros, además de capacidad de aproximación universal, lo que hace a los KAF adecuados para tareas como predicción, identificación y control [4]. El

aprendizaje es secuencial por corrección de error sobre el mapeo no lineal como se muestra en la ecuación (4).

En un RKHS con núcleo $\kappa(\cdot,\cdot)$, el estimador adopta una expansión sobre "secciones" del *kernel*, como se describió en la ecuación (6). En consecuencia, la salida para un punto nuevo obtiene la forma de la ecuación (9):

$$y(i) \;=\; f_i\big(u(i)\big) \;=\; \sum_{j=1}^{i} \alpha_j \, \kappa\big(u(j),\, u(i)\big). \tag{9}$$

Un caso prototípico es el *Kernel LMS* (KLMS). Esta formulación captura la no linealidad a través del núcleo mientras la adaptación permanece lineal en el RKHS, con buena eficiencia computacional para operación en línea [4].

6.4.1. Algoritmo *Kernel Least-Mean-Square* (KLMS)

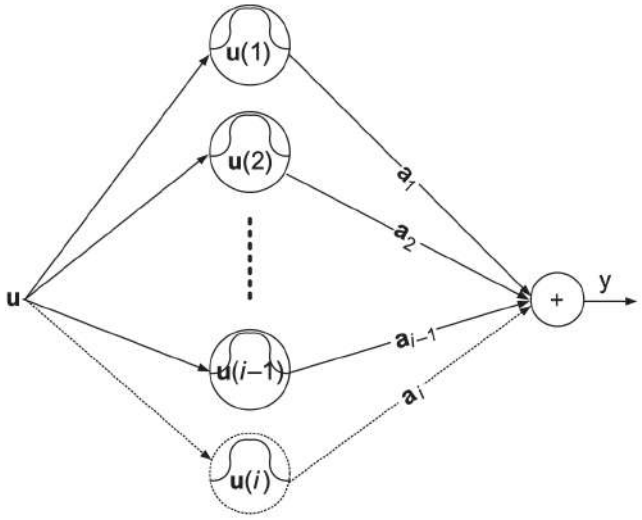
La idea base detrás del algoritmo *kernel least-mean-square* es ser la versión con *kernels* del LMS clásico. En lugar de ajustar un vector de pesos sobre las entradas $\mathbf{u}(i)$, ajusta una función f en un espacio de Hilbert con *kernel* de reproducciones (RKHS). La salida se obtiene como suma de *kernels* centrados en muestras previas; el error instantáneo corrige f con un paso proporcional al error [4].

Modelo y regla de actualización:

1. Predicción: $y(i) \;=\; f_{i-1}\big(u(i)\big) \;=\; \sum_{j=1}^{i-1} \alpha_j \, k\big(u(j),\, u(i)\big)$
2. Error: $e(i) \;=\; d(i) - y(i)$
3. Actualización funcional: $f_i \;=\; f_{i-1} \;+\; \eta \, e(i) \, k\big(u(i),\, \cdot\big)$,

Aquí η es el tamaño de paso y $k(\cdot,\cdot)$ el *kernel* (p. ej., Gaussiano o polinomial).El algoritmo KLMS conserva la simplicidad del LMS, opera en línea y captura las no linealidades vía el *kernel*, tal y como lo muestra la Figura 4 [4].

Figura 4. Topología de red de un KLMS en la iteración i



Nota. Imagen obtenida de [4].

6.5. Aprendizaje profundo a través de redes neuronales

El aprendizaje profundo (*deep learning*) es una subdisciplina del aprendizaje automático (*machine learning*) que busca modelar representaciones jerárquicas de los datos mediante redes neuronales artificiales con múltiples capas ocultas. Su principio fundamental radica en la capacidad de estas redes para aprender características de alto nivel de abstracción a partir de datos crudos, sin necesidad de diseñar manualmente las transformaciones previas [8].

Este enfoque ha sido impulsado por tres factores principales:

1. Disponibilidad de grandes volúmenes de datos, lo que permite entrenar modelos de gran capacidad.
2. Avances en *hardware*, especialmente el uso de unidades de procesamiento gráfico (GPU) para acelerar el cálculo matricial.
3. Desarrollo de nuevas arquitecturas y funciones de activación, que facilitan la propagación del gradiente a través de redes muy profundas, resolviendo en gran medida los problemas de saturación y desvanecimiento presentes en modelos clásicos [8].

6.5.1. Teorema de aproximación universal

El Teorema de aproximación universal es el fundamento teórico sobre el que se apoya el aprendizaje profundo. Este establece que una red neuronal de una sola capa oculta, con un

número finito de neuronas y una función de activación no lineal apropiada, puede aproximar cualquier función continua en un subconjunto compacto con la precisión deseada [9].

En términos matemáticos, sea $f : \mathbb{R}^n \to \mathbb{R}$ una función continua y sea $\sigma(\cdot)$ una función de activación no lineal (como la sigmoide o la ReLU). Entonces, existe un número finito de neuronas N, pesos w_i, sesgos b_i y coeficientes α_i tales que:

$$f(x) \approx \sum_{i=1}^{N} \alpha_i \, \sigma(w_i^T x + b_i) \tag{10}$$

para todo x en un dominio compacto de $\mathbb{R}^n$ [10].

6.5.2. Tipos de aprendizaje

El aprendizaje profundo puede dividirse en tres paradigmas fundamentales, de acuerdo con la disponibilidad y tipo de etiquetas en los datos [8]:

- **Aprendizaje supervisado:** el modelo aprende a partir de un conjunto de pares entrada–salida (x_i, y_i), buscando aproximar una función $f(x) \approx y$ que minimice el error entre la predicción y la etiqueta real.
- **Aprendizaje no supervisado:** el modelo no dispone de etiquetas explícitas y su objetivo es descubrir representaciones útiles en los datos. Entre los ejemplos más comunes se encuentran los *autoencoders*, las redes generativas adversarias (GAN) y los modelos de mezcla.
- **Aprendizaje por refuerzo:** este enfoque se basa en un agente que aprende a interactuar con un entorno mediante la maximización de una recompensa acumulada [8].

6.6. El perceptrón y redes neuronales

El perceptrón constituye la unidad fundamental de una red neuronal artificial. Las redes neuronales están compuestas por múltiples de ellos. Su funcionamiento se basa en el cálculo de una combinación lineal de las entradas ponderadas, seguida de la aplicación de una función de activación que determina la salida del modelo [11] [8].

De forma general, el perceptrón simple puede representarse como:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{11}$$

donde:

- x_i representa las entradas o características del sistema,
- w_i son los pesos asociados a cada entrada,
- b corresponde al sesgo (o término de desplazamiento),
- $\sigma(\cdot)$ es la función de activación que introduce no linealidad.

6.6.1. Pesos

Los pesos (w_i) determinan la influencia que cada entrada ejerce sobre la salida del perceptrón. Durante el entrenamiento, estos valores se ajustan iterativamente para minimizar la diferencia entre la predicción del modelo y el valor real esperado. Matemáticamente, los pesos se actualizan según la regla del gradiente descendente. Los pesos son los parámetros de mayor relevancia en una red, ya que encapsulan el conocimiento aprendido a partir de los datos [8] [12].

6.6.2. Sesgos

El sesgo (b) actúa como un desplazamiento del hiperplano de decisión generado por los pesos, permitiendo que la red neuronal modele funciones que no necesariamente pasen por el origen. Su inclusión incrementa la flexibilidad del modelo, especialmente en combinaciones lineales [12].

6.6.3. Funciones de activación

Las funciones de activación introducen la no linealidad necesaria para que una red neuronal pueda aproximar relaciones complejas entre las variables de entrada y salida. Sin estas funciones, la red se reduciría a una simple combinación lineal de sus entradas, perdiendo capacidad de representación [8].

Entre las funciones más utilizadas se encuentran:

- **Sigmoide:**
$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{12}$$
Produce salidas continuas entre 0 y 1, pero tiende a saturarse para valores extremos, afectando el flujo del gradiente [9].

- **Tangente hiperbólica:**
$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{13}$$
Centrada en cero, mejora la simetría respecto a la sigmoide, aunque sufre un problema similar de saturación [12].

- **Unidad lineal rectificada (ReLU):**

$$\text{ReLU}(x) = \text{máx}(0, x) \tag{14}$$

  Permite entrenar redes profundas al evitar la saturación del gradiente en valores positivos. Variantes como *LeakyReLU* y *ELU* extienden su comportamiento para valores negativos [12].

Estas funciones determinan la dinámica de activación de las neuronas, y su elección depende de la tarea y la arquitectura del modelo.

6.6.4. Cálculo del gradiente

El cálculo del gradiente es un proceso importante en el entrenamiento de redes neuronales, ya que permite ajustar los pesos y sesgos minimizando la función de pérdida. Este se realiza mediante el algoritmo de retropropagación (*backpropagation*), el cual aplica la regla de la cadena para propagar los errores desde la salida hasta las capas anteriores [8].

Para una función de pérdida $\mathcal{L}$ y un parámetro genérico θ (peso o sesgo), el gradiente se define como:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial y} \cdot \frac{\partial y}{\partial \theta} \tag{15}$$

Este gradiente cuantifica cómo un pequeño cambio en θ afecta la pérdida total del modelo. Posteriormente, se utiliza para actualizar los parámetros siguiendo la dirección del descenso más pronunciado [12].

6.7. Redes neuronales profundas y detalles de entrenamiento

Las redes neuronales profundas extienden el concepto del perceptrón simple al apilar múltiples capas ocultas, permitiendo la composición jerárquica de funciones no lineales. Esta profundidad les otorga la capacidad de modelar relaciones altamente complejas entre las variables de entrada y salida, convirtiéndolas en el núcleo del aprendizaje profundo moderno [8] [12].

A continuación, se detallan los principales componentes del proceso de entrenamiento de una red neuronal profunda.

6.7.1. Propagación hacia adelante

La propagación hacia adelante (*forward propagation*) consiste en calcular, de manera secuencial, las salidas de cada capa hasta obtener la predicción final del modelo. En cada paso, los datos se transforman linealmente mediante los pesos y sesgos, y luego se aplican

funciones de activación no lineales. Durante esta fase no se realiza ajuste de parámetros; únicamente se calcula la salida en función de las entradas actuales. El resultado obtenido se utilizará luego para calcular la función de pérdida [10].

6.7.2. Cálculo de la pérdida

La función de pérdida (*loss function*) cuantifica la discrepancia entre la salida predicha $\hat{y}$ y la salida real y. Su elección depende de la naturaleza del problema [8] [12]:

- En tareas de regresión, se suele utilizar el *Error Cuadrático Medio* (MSE):

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{16}$$

- En clasificación, se utiliza la *entropía cruzada*:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i). \tag{17}$$

6.7.3. Retropropagación

La retropropagación del error (*backpropagation*) es el algoritmo encargado de ajustar los pesos de la red en función del gradiente de la pérdida. Este método aplica la regla de la cadena para calcular las derivadas parciales de la función de pérdida con respecto a cada parámetro.

Este proceso se realiza de manera iterativa desde la capa de salida hacia las capas anteriores, acumulando los gradientes que posteriormente serán utilizados por el optimizador [12].

6.7.4. Iteraciones y épocas

Una época corresponde al proceso de entrenamiento de la red utilizando la totalidad del conjunto de datos una vez. En cada época, los datos suelen dividirse en lotes o *batches*, lo que permite actualizar los pesos varias veces por época, reduciendo el costo computacional y mejorando la estabilidad del aprendizaje. El número de épocas necesarias depende de la complejidad del problema y del tamaño de los datos [8].

6.7.5. Optimización

Los algoritmos de optimización determinan cómo se actualizan los parámetros del modelo a partir de los gradientes calculados. El método más básico es el descenso del gradiente estocástico (SGD), sin embargo; existen otros tales como: *Adam*, *RMSProp* y *Adagrad*. Estos métodos mejoran la velocidad de convergencia y la estabilidad del entrenamiento, especialmente en redes profundas con gran cantidad de parámetros [8] [12].

6.8. Redes convolucionales temporales (TCN)

Las redes convolucionales temporales (*Temporal Convolutional Networks*, TCN) constituyen una arquitectura de aprendizaje profundo diseñada específicamente para procesar datos secuenciales preservando el orden temporal y garantizando la causalidad. A diferencia de los modelos recurrentes tradicionales, que procesan la información de manera secuencial, las TCN emplean convoluciones unidimensionales causales y dilatadas, lo que les permite capturar dependencias a largo plazo de forma paralela, estable y eficiente [13]. Esta combinación ha demostrado ser altamente efectiva en tareas de procesamiento de audio, donde las relaciones temporales no lineales juegan un papel central en la representación de la señal.

La convolución causal garantiza que la salida en un instante t dependa únicamente de muestras presentes o pasadas de la señal, preservando la direccionalidad temporal:

$$y(t) = \sum_{k=0}^{K-1} f(k)\, x(t-k). \tag{18}$$

Sin embargo, el uso exclusivo de convoluciones causales limitaría el campo receptivo, requiriendo redes muy profundas o filtros excesivamente grandes. Para superar esta limitación, las TCN incorporan convoluciones dilatadas, en las que se introduce un factor de espaciamiento d entre las muestras utilizadas por el filtro:

$$y(t) = \sum_{k=0}^{K-1} f(k)\, x(t-d\times k). \tag{19}$$

El incremento exponencial del factor de dilatación entre capas permite que el campo receptivo crezca también de forma exponencial, logrando abarcar cientos o miles de muestras con un número relativamente pequeño de capas [14]. Esta propiedad es particularmente útil en audio, donde eventos relevantes pueden estar separados temporalmente pero seguir siendo dependientes.

Además, las TCN suelen emplear bloques residuales y conexiones de salto (*skip connections*), siguiendo principios introducidos en [15], lo que facilita el entrenamiento de redes profundas y mejora la estabilidad numérica. Estos bloques están compuestos por una convolución causal y dilatada, seguida de una no linealidad y normalización, y su salida se suma directamente a la entrada:

$$y = x + F(x), \tag{20}$$

donde $F(x)$ representa la transformación aplicada dentro del bloque residual. Esta estrategia no solo favorece la propagación del gradiente, sino que también permite que capas profundas modelen dependencias complejas sin degradar la información de las capas anteriores.

Investigaciones previas han mostrado que las TCN igualan o superan el desempeño de arquitecturas recurrentes en múltiples tareas secuenciales, manteniendo al mismo tiempo

un entrenamiento más estable y altamente paralelizable [13]. En el campo del audio, estos modelos se inspiran en arquitecturas como WaveNet [14], que utiliza pilas de convoluciones dilatadas causales para generar o reconstruir señales a nivel de muestra, capturando estructuras temporales de corto y largo alcance con alta fidelidad.

6.9. Modelos NLARX (auto-regresivos no lineales con entradas exógenas)

Los modelos auto-regresivos no lineales con entradas exógenas (*Nonlinear AutoRegressive with eXogenous inputs*, NLARX) constituyen una clase de modelos utilizados en la identificación de sistemas dinámicos no lineales. Estos modelos extienden la formulación clásica ARX al permitir que la relación entre entradas, salidas y sus retardos sea definida mediante una función no lineal con capacidad de aproximación universal [16] [17].

Un modelo NLARX describe la salida de un sistema como

$$y(t) = f\big(y(t-1), \ldots, y(t-n_y),\, u(t-1), \ldots, u(t-n_u)\big), \tag{21}$$

donde n_y y n_u representan los órdenes de regresión de la salida y la entrada, respectivamente, y $f(\cdot)$ es una función no lineal parametrizada. Esta estructura conserva el marco autoregresivo tradicional, pero reemplaza la combinación lineal de regresores por una representación no lineal capaz de capturar interacciones complejas entre las variables del sistema.

6.9.1. Estructura general

De acuerdo con la documentación oficial de MathWorks [17], un modelo NLARX se compone típicamente de:

- **Regresores**: conjunto de valores retardados de la entrada y la salida que definen la memoria del sistema.
- **Función no lineal**: aproximador universal que opera sobre los regresores. Se pueden emplear redes neuronales *feedforward*, funciones de base radial (RBF), árboles de regresión o funciones definidas por el usuario.
- **Mecanismo de estimación**: algoritmos iterativos, como variantes de Levenberg–Marquardt, que ajustan los parámetros de la función no lineal para minimizar el error de predicción.

6.9.2. Interpretación dinámica

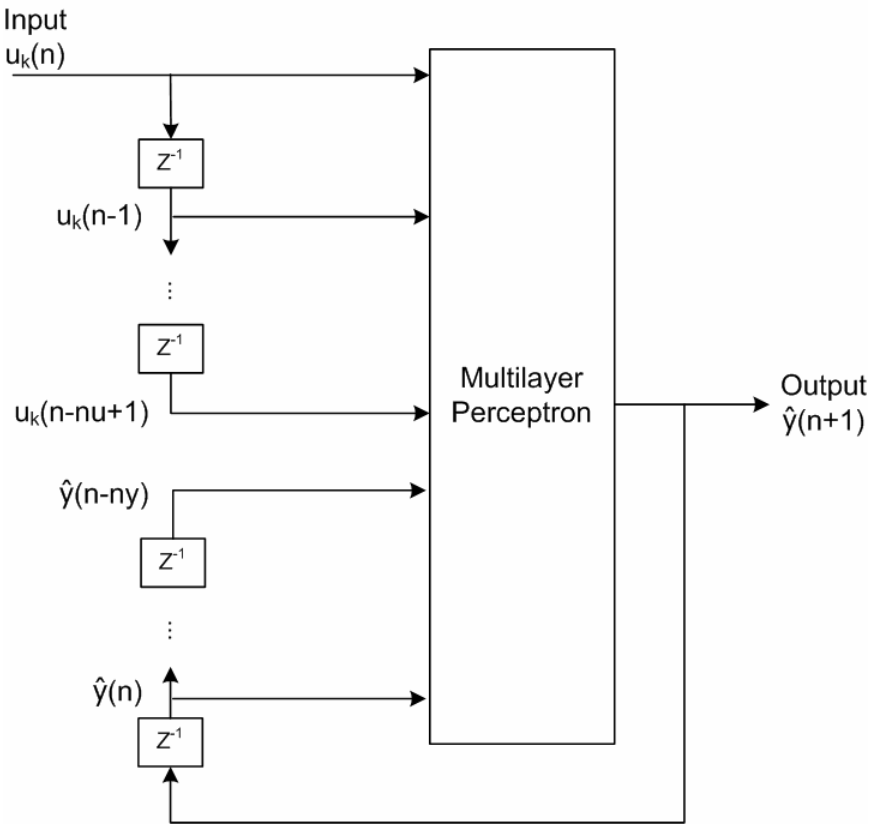
Según Leontaritis y Billings [16], la clase NARX/NLARX pertenece a los modelos no lineales en tiempo discreto que pueden aproximar sistemas con memoria finita mediante

combinaciones de regresores y funciones no lineales. Debido a que el modelo opera de forma recursiva, dependiendo de valores pasados de la salida real, posee un comportamiento equivalente al de un filtro IIR no lineal.

Esta equivalencia es conceptual, aunque el modelo no implementa coeficientes IIR lineales, la presencia de retardos de salida introduce una memoria efectiva que se preserva a lo largo del tiempo, rasgo característico de los sistemas recursivos.

La teoría clásica de identificación no lineal establece que, bajo condiciones adecuadas de los regresores, los modelos NARX y sus variantes no lineales pueden aproximar con precisión arbitraria un amplio conjunto de sistemas dinámicos [18]. La flexibilidad de estos modelos y su capacidad de incorporar funciones no lineales los convierten en una herramienta fundamental dentro del modelado de sistemas en tiempo discreto.

Figura 5. Estructura canónica de un modelo NLARX



Nota. Imagen obtenida de [19].

CAPÍTULO 7

Consolidación de una base de datos de audio

Uno de los aportes técnicos más relevantes de esta investigación es la consolidación de una base de datos unificada de señales de audio. En etapas anteriores de la línea de investigación sobre deconvolución acústica, se generaron múltiples grabaciones bajo distintas condiciones de captura, metodologías y configuraciones de equipo. Sin embargo, dichas grabaciones se encontraban dispersas, con diferencias notables en formato, duración, amplitud y fase, lo que dificultaba su uso conjunto para el entrenamiento y validación de modelos de aprendizaje automático.

En el presente trabajo se llevó a cabo un proceso sistemático de integración, depuración y estandarización de estos datos, con el propósito de construir un conjunto coherente y reproducible que sirva tanto para la evaluación de los métodos propuestos en este trabajo como para futuras investigaciones en el área de procesamiento de audio de la Universidad del Valle de Guatemala.

7.1. Fuentes de los datos

La base de datos consolidada se construyó a partir de grabaciones recopiladas en dos investigaciones previas. En particular, se retomaron los conjuntos de datos generados por Carlos López (2021) y Alexander Calí (2024), los cuales abordan el problema de la deconvolución acústica desde enfoques metodológicos distintos pero complementarios.

El trabajo de López [2] representó la segunda fase de esta línea de investigación, enfocándose en la aplicación de filtros adaptativos lineales y no lineales (LMS, NLMS y RLS) para eliminar la reverberación en grabaciones domésticas. Las señales utilizadas fueron generadas y capturadas en entornos no tratados acústicamente, empleando altavoces y micrófonos integrados en computadora dentro del entorno de desarrollo *MATLAB*. Este conjunto de

datos permitió evaluar el desempeño de los filtros en condiciones reales de ruido, aunque las grabaciones presentaban limitaciones en fidelidad y control acústico.

Por otro lado, la investigación de Calí [3] representó un avance significativo al incorporar grabaciones realizadas en el estudio audiovisual de la Universidad del Valle de Guatemala, un entorno profesional con tratamiento acústico y mediciones controladas de ruido de piso, presión sonora y tiempo de reverberación. Calí grabó muestras de diversos géneros musicales (electrónica, bachata, jazz, pop, regional mexicano, reguetón e instrumental) tanto en el estudio de grabación como en recintos no tratados, estableciendo así un marco comparativo entre entornos acústicamente ideales y no ideales.

Ambas investigaciones aportaron subconjuntos de datos valiosos que ya contemplaban pares de señales *entrada–salida*: una versión limpia o de referencia (*señal deseada*) y su respectiva versión degradada (*señal observada*) obtenida mediante la convolución con la respuesta acústica del entorno. Esta característica permitió disponer de datos comparables entre grabaciones en entornos tratados y no tratados, así como entre diferentes configuraciones de captura.

El Cuadro 1 resume las principales diferencias y características de las bases de datos empleadas.

Cuadro 1. Comparación entre las bases de datos de Carlos López (2021) y Alexander Calí (2024)

Característica	**Carlos López (2021)**	**Alexander Calí (2024)**
Tipo de entorno	Recintos domésticos sin tratamiento acústico	Estudio audiovisual UVG y recintos no tratados
Dispositivo de captura	Micrófono integrado	Behringer ECM8000 + AudioBox USB96
Software de grabación	MATLAB (`audiorecorder`)	Reaper 7.17
Frecuencia de muestreo	44.1 kHz	44.1 kHz
Resolución	24 bits	24 bits
Canales	Mono	Mono
Contenido grabado	Señales sintéticas y música breve	20+ canciones de distintos géneros
Nivel de ruido de piso	N/A	25–35 dB (medido)
Finalidad original	Evaluar filtros LMS y Volterra	Entrenar redes CNN, RNN y TCN

Nota. Elaboración propia.

En este trabajo se desarrolló un proceso de consolidación que integra y estandariza ambos conjuntos, garantizando la coherencia temporal, la correspondencia entre pares y la compatibilidad de formatos. Como resultado, se conformó una base de datos unificada, reproducible y documentada, que constituye un recurso experimental común para evaluar y

comparar distintos enfoques de deconvolución acústica, el cual se describe en detalle en la siguiente sección.

7.2. Equipamiento y configuración experimental

La consolidación de la base de datos requirió respetar las condiciones de captura originales de cada estudio y, a partir de ellas, definir criterios unificados de procesamiento. A continuación se describen las configuraciones empleadas por López y Calí, así como las decisiones adoptadas en este trabajo para integrar sus grabaciones en un mismo repositorio.

7.2.1. Configuración empleada por López (2021)

El conjunto de datos de López [2] fue generado en recintos domésticos sin tratamiento acústico, empleando equipamiento de bajo costo representativo de un escenario realista de uso. Como fuente emisora se utilizó una bocina estéreo iHome iHM9, mientras que la captura se realizó con el micrófono integrado de una computadora portátil ASUS X555L. La adquisición de las señales se efectuó mediante el entorno *MATLAB*, utilizando la función `audiorecorder`.

Para cada caso de prueba se dispuso de pares de señales con la siguiente estructura:

- `[nombre]_Original.wav`: señal limpia o de referencia, generada digitalmente.
- `[nombre]_Recorded.wav`: señal grabada tras la reproducción de `[nombre]_Original.wav` en el recinto, capturada por el micrófono integrado.

Las señales incluyen tonos sinusoidales, ondas cuadradas, barridos frecuenciales y fragmentos musicales breves. La frecuencia de muestreo utilizada fue de 44.1 kHz, con resolución de 24 bits en un solo canal. Si bien la respuesta en frecuencia del sistema iHome iHM9 + micrófono integrado es limitada y más susceptible al ruido ambiental, este conjunto resulta particularmente útil para modelar condiciones de reverberación y degradación típicas de entornos no controlados, aportando casos de prueba exigentes para los métodos de deconvolución.

7.2.2. Configuración empleada por Calí (2024)

El conjunto de datos de Calí [3] fue adquirido utilizando una cadena de grabación de mayor fidelidad, implementada en el estudio audiovisual de la Universidad del Valle de Guatemala y en recintos adicionales sin tratamiento acústico. La configuración principal incluyó:

- **Micrófono omnidireccional** *Behringer ECM8000* con alimentación *phantom* de +48 V.

- **Interfaz de audio** *AudioBox USB96*, conectada mediante cableado balanceado tipo XLR.
- **Software de grabación** *Reaper 7.17*.

Las grabaciones se realizaron en modo *mono*, con frecuencia de muestreo de 44.1 kHz y resolución de 24 bits. Se registraron fragmentos de más de veinte canciones pertenecientes a distintos géneros musicales, tanto en el estudio con tratamiento acústico como en recintos no tratados. Durante las sesiones se controlaron los niveles de presión sonora en el rango de 85–95 dB utilizando la aplicación *Decibel X*, y se verificaron niveles de ruido de piso entre 25 dB y 35 dB en el estudio.

Similar que en el caso de López, cada fragmento se organizó en pares:

- `[Género]_[nombre]_Original.wav`: señal limpia de referencia.
- `[Género]_[nombre]_Recorded.wav`: señal capturada tras la reproducción de `[Género]_[nombre]_Original.wav` en el entorno físico correspondiente.

Esta configuración proporciona pares con alta relación señal–ruido y mejor control de las condiciones acústicas, constituyendo una referencia más estable para la evaluación de modelos.

7.2.3. Criterios unificados de alineación y duración

Con el objetivo de integrar ambos conjuntos en una base de datos coherente y directamente utilizable para experimentos de aprendizaje automático, en este trabajo se implementaron dos procedimientos de estandarización basados en scripts desarrollados en *MATLAB*.

En ambos casos se aplicaron los siguientes pasos comunes:

1. **Emparejamiento automático**: identificación de pares `[nombre]_Original.wav` – `[nombre]_Recorded.wav` mediante la convención de nombres.
2. **Conversión a mono**: promediado de canales en caso de señales estéreo.
3. **Remuestreo**: ajuste de todas las señales a una frecuencia de muestreo objetivo de **44.1 kHz** (`TARGET_FS`).
4. **Alineación temporal**: cálculo de la correlación cruzada entre segmentos iniciales de las señales limpia y grabada, con un retardo máximo de ± 5 s, y compensación del desfase detectado desplazando la señal grabada. Este procedimiento corrige diferencias de latencia introducidas por la cadena de reproducción–captura.

Posteriormente, se unificó la duración efectiva de cada par a 30 segundos (`TARGET_LENGTH_SEC`), con decisiones específicas según el origen:

- **Conjunto de López (DATABASE_CARLOS)**: debido a que varias señales originales eran relativamente cortas, se aplicó un esquema de *circular wrapping*, repitiendo la señal las veces necesarias y, en caso requerido, completando con un segmento inicial para alcanzar exactamente 30 s. Esta decisión evita introducir silencios artificiales extensos y garantiza una ventana de observación uniforme para el análisis de los modelos, a costa de repetir contenido en algunos casos.
- **Conjunto de Calí (DATABASE_ALEX)**: dado que los fragmentos musicales disponen de mayor duración, la unificación a 30 s se realizó mediante truncamiento o rellenado con silencio (*zero-padding*) según fuera necesario. Este criterio preserva la integridad espectral sin necesidad de repetir la señal, manteniendo una estructura temporal continua representativa de cada grabación.

Finalmente, todos los pares alineados y ajustados fueron almacenados en carpetas diferenciadas por origen, con reportes de alineación (*alignment reports*) que documentan, para cada archivo, la longitud original, el desfase corregido y el método aplicado (truncamiento, relleno o repetición). Este procedimiento asegura trazabilidad, coherencia entre señales limpias y degradadas, y homogeneidad en la base de datos consolidada.

7.3. Estructura y organización de la base de datos consolidada

Tras completar el proceso de alineación, normalización y estandarización de las señales provenientes de los conjuntos de López (2021) y Calí (2024), se conformó una única base de datos consolidada en formato reproducible. Aunque durante la etapa de desarrollo se emplearon scripts en *MATLAB* para automatizar tareas de emparejamiento, alineación y estandarización —los cuales generaban carpetas y reportes individuales (`DATABASE_CARLOS`, `DATABASE_ALEX`, `alignment_report.txt`)—, la versión final de la base de datos fue unificada manualmente en una sola estructura, con el propósito de simplificar su distribución y uso en futuros trabajos.

7.3.1. Estructura general del repositorio

El repositorio final contiene todos los pares de señales *entrada–salida* en un único nivel jerárquico. Cada archivo conserva la nomenclatura establecida por Calí (2024), lo cual permite identificar el género musical, la pista y la condición de grabación sin necesidad de archivos de metadatos adicionales. La estructura general es la siguiente:

```
AUDIO_DATASET/
|-- [Género]_[Nombre]_Original.wav   # Señal limpia o de referencia
|-- [Género]_[Nombre]_Recorded.wav   # Señal degradada capturada en entorno físico
|-- ...
\‘-- README.txt             # Descripción general y créditos de origen
```

Esta organización facilita la reutilización directa del conjunto en experimentos de identificación de sistemas y deconvolución acústica, sin requerir procesamiento adicional. Los

pares de señales se distinguen exclusivamente por su sufijo (`_Original` o `_Recorded`), lo que permite automatizar su lectura en los scripts de entrenamiento.

7.3.2. Convención de nombres y etiquetado

Para mantener la coherencia con los estudios previos y garantizar la trazabilidad de los datos, se adoptó la misma convención de nombres utilizada por Calí (2024). Cada archivo sigue el formato:

`[Género]_[Nombre]_[Condición].wav`

donde:

- `[Género]` indica el género musical o tipo de señal: `El` (Electrónica), `Ba` (Bachata), `Ja` (Jazz), `Po` (Pop), `Ra` (Regional Mexicano), `Rg` (Reguetón), `Ins` (Instrumental), `Sg` (Señal sintética).
- `[Nombre]` corresponde al nombre de la canción o estímulo sonoro.
- `[Condición]` identifica la naturaleza de la grabación: `Original` para la señal limpia y `Recorded` para la señal capturada en el entorno acústico.

Ejemplos de nombres de archivo:

- `El_CarryYou_Original.wav`
- `El_CarryYou_Recorded.wav`
- `Ba_Eres_Original.wav`
- `Ba_Eres_Recorded.wav`

Esta convención permitió mantener la compatibilidad con el esquema original de Calí y procesar automáticamente los pares de señales en los scripts desarrollados para este trabajo.

7.3.3. Contenido de la base de datos consolidada

El conjunto resultante integra tanto las grabaciones musicales de Calí como las señales experimentales de López, incluyendo tonos sinusoidales, patrones instrumentales y fragmentos musicales de distintos géneros. El Cuadro 2 muestra las canciones incluidas en la base consolidada, mientras que las señales de López complementan el *dataset* con estímulos sintéticos y ejemplos de grabaciones domésticas.

Cuadro 2. Lista de canciones y estímulos grabados incluidos en la base consolidada

No.	Género o tipo de señal	Nombre de la canción o estímulo	Artista / Fuente
Grabaciones musicales (Calí, 2024)			
1	Electrónica	Carry You	Martin Garrix
2	Electrónica	Levels	Avicii
3	Electrónica	Haven Takes You Home	Swedish House Mafia
4	Electrónica	Get Lucky	Daft Punk
5	Bachata	Eres	Romeo Santos
6	Bachata	Llévame Contigo	Romeo Santos
7	Bachata	Corazón	Prince Royce
8	Jazz	It's Been a Long Long Time	Harry James
9	Jazz	What a Wonderful World	Louis Armstrong
10	Pop	A Man After Midnight	ABBA
11	Pop	Dancing Queen	ABBA
12	Regional Mexicano	Aquí Abajo	Christian Nodal
13	Regional Mexicano	De los Besos	Christian Nodal
14	Regional Mexicano	Nace	Christian Nodal
15	Regional Mexicano	Quién de los Dos Será	Diego Verdaguer
16	Regional Mexicano	Volver al Futuro	Oscar Maydon
17	Regional Mexicano	Madona	Natanael Cano
18	Reguetón	Volando	Mora
19	Reguetón	Ferxxo 151	Feid
20	Instrumental	11YOnce	Tainy
21	Instrumental	Sacrificio	Tainy
22	Instrumental	Cover Luna de Xelajú	Gaby Moreno
23	Instrumental	Solo de Piano	Tainy
24	Instrumental	Solo de Guitarra	Tainy
Señales experimentales (López, 2021)			
25	Instrumental	Cadence	The Long Faces
26	Instrumental	Lonely Cat	The Kooks
27	Instrumental	Week No. 8	Fabrizio Paterlini
28	Jazz	Atlantic Limited	Julian Lage
29	Señal sintética	AM_FM	Generada digitalmente
30	Señal sintética	Sine 250	Generada digitalmente

Nota. Elaboración propia con base en Calí [3] y López [2].

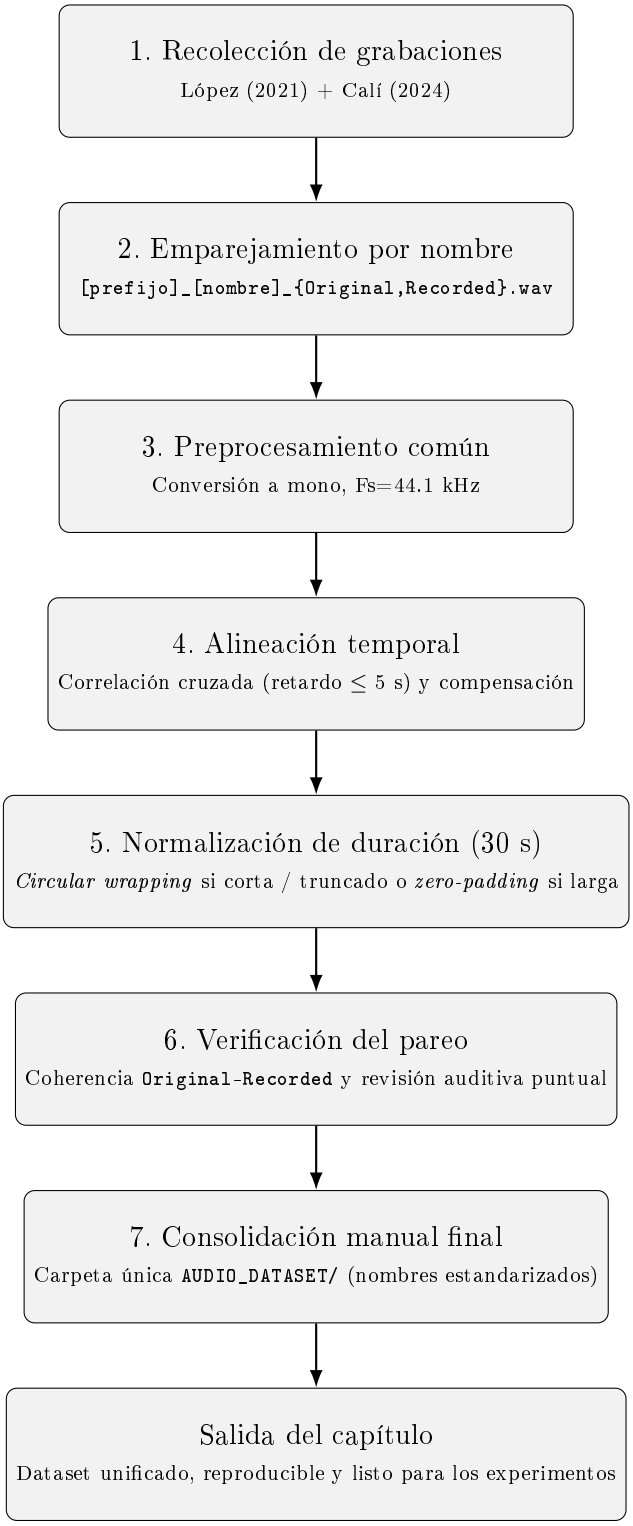
La base de datos consolidada representa un conjunto heterogéneo y completo de señales limpias y degradadas, abarcando desde estímulos sintéticos hasta grabaciones musicales de estudio. Esta diversidad en contenido refuerza su valor como recurso experimental para investigaciones futuras en deconvolución acústica y aprendizaje profundo.

Además de integrar la información proveniente de trabajos previos, la consolidación permitió cuantificar con claridad el crecimiento del conjunto experimental. La base de datos de Calí aportaba 24 señales, mientras que la base de López incorporaba 6 señales adicionales, para un total de 30 registros disponibles antes de este estudio. Tras el proceso de depuración, alineación temporal y estandarización, la base de datos consolidada quedó conformada por 68 señales en total, equivalentes a 34 pares de señales limpia/degradada. Este incremento respecto a las bases originales amplía de forma significativa la diversidad de géneros musicales, timbres y condiciones acústicas representadas y proporciona un conjunto de datos más robusto para el análisis comparativo y el entrenamiento de modelos en trabajos futuros.

7.4. Resumen del proceso de consolidación

La Figura 6 resume de manera integrada las etapas que permitieron unificar y estandarizar las grabaciones provenientes de López y Calí. Este flujo sintetiza el proceso completo desde la recolección y clasificación de señales hasta su alineación temporal, normalización y organización final en un único repositorio reproducible. Con ello, se cierra el trabajo de consolidación desarrollado en este capítulo, estableciendo una base de datos coherente y homogénea que sirve como punto de partida para los experimentos de identificación y deconvolución acústica que se presentan en los siguientes capítulos.

Figura 6. Flujo de procesamiento para la consolidación de la base de datos



Nota. Elaboración propia.

CAPÍTULO 8

Experimentación con identificación de sistemas

La experimentación presentada en este capítulo constituye una etapa preliminar destinada a evaluar el desempeño de los modelos seleccionados en un entorno controlado, antes de aplicarlos a grabaciones reales. Para ello se diseñaron señales sintéticas y un conjunto de pruebas que permiten observar cómo cada método responde ante ruido, variaciones espectrales y cambios en la dinámica temporal. Este análisis inicial no forma parte directa de los objetivos específicos del trabajo, pero proporciona información esencial sobre la estabilidad, convergencia y capacidad de generalización de los modelos, elementos fundamentales para sustentar la etapa posterior de deconvolución acústica.

8.1. Introducción y propósito de la experimentación

La identificación de sistemas se incorporó como una etapa intermedia del trabajo con el fin de evaluar, bajo condiciones controladas, la capacidad de los modelos propuestos para aprender relaciones entrada–salida antes de su aplicación a señales de audio reales. Aunque esta fase no forma parte de los objetivos específicos del trabajo de graduación, su ejecución ofrece un marco experimental que permite analizar convergencia, estabilidad y capacidad de generalización de tres enfoques representativos: *Kernel Least Mean Squares* (KLMS), *Nonlinear AutoRegressive with eXogenous input* (NLARX) y *Temporal Convolutional Networks* (TCN).

Metodológicamente, la identificación de sistemas proporciona un terreno de pruebas reproducible en el que se controla el estímulo de entrada y la dinámica a estimar. De este modo, se evita la complejidad propia del audio real y se aísla la capacidad de cada modelo para aproximar mapeos dinámicos causales. En este capítulo se documenta el diseño de señales sintéticas, el protocolo de entrenamiento–validación y la evaluación comparativa preliminar, estableciendo una base sólida para la etapa posterior de deconvolución acústica

con grabaciones reales.

8.2. Diseño de las señales de prueba

Las señales sintéticas de entrenamiento y validación se generaron con una frecuencia de muestreo de 44.1 kHz, duración de 10 s y normalización de amplitud al rango $[-1, 1]$. Esta configuración estandariza la energía de entrada, evita saturación numérica y permite comparar consistentemente el desempeño entre KLMS, NLARX y TCN.

8.2.1. Onda cuadrada (estímulo principal)

Se empleó una onda cuadrada de **440 Hz** (La_4) como estímulo principal, dado su alto contenido armónico y su relevancia en el dominio del audio. La presencia de armónicos a 880, 1320, 1760 Hz, etc., excita porciones significativas del espectro audible y exige a los modelos conservar fase y amplitud frente a transiciones rápidas. Con Fs = 44.1 kHz, cada ciclo se representa con ≈ 100 muestras, lo que garantiza resolución temporal suficiente para el aprendizaje de la dinámica.

8.2.2. Otras señales de entrenamiento (complementarias)

Con el propósito de explorar la sensibilidad de los modelos frente a patrones espectrales diferentes, se utilizaron adicionalmente:

- **Onda diente de sierra**, cuyo contenido armónico denso permite analizar respuestas ante espectros más continuos y flancos pronunciados.
- **Ondas senoidales puras** en frecuencias puntuales del rango bajo–medio, útiles para verificar linealidad local y seguimiento de fase.

Estas señales complementarias se generaron bajo las mismas condiciones (Fs, duración y normalización) y se aplicaron en entrenamientos individuales por modelo.

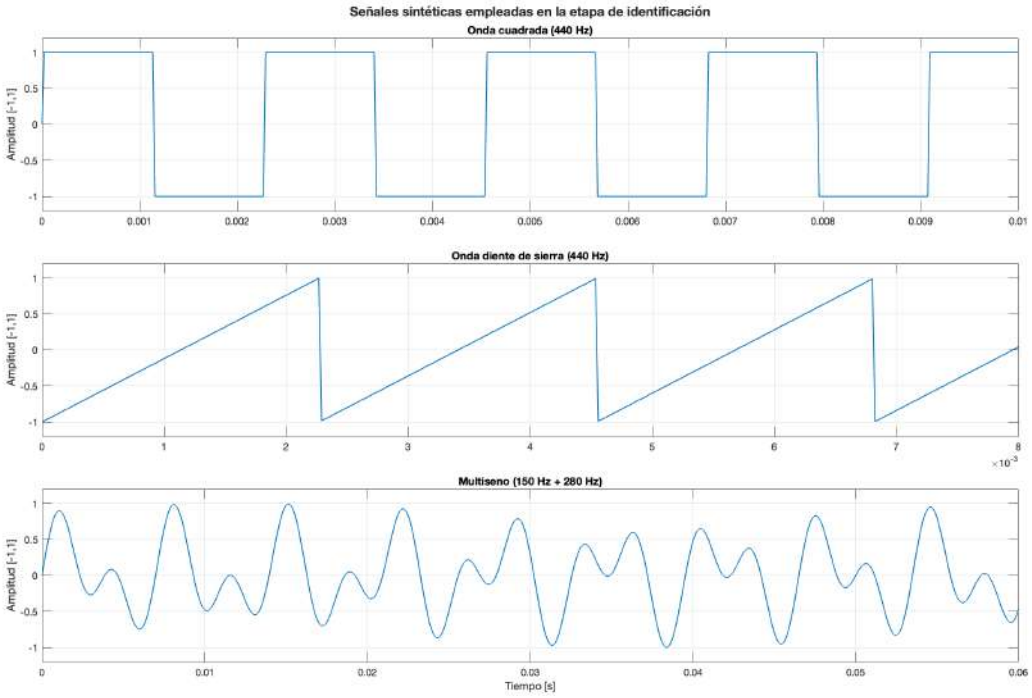
8.2.3. Señal de validación: multiseno

Tras el entrenamiento, la verificación se realizó con una señal *multiseno* compuesta por dos tonos no armónicos de 150 Hz y 280 Hz. Esta elección en banda media es más representativa para aplicaciones acústicas que pares subaudibles, y evita relaciones simples (p. ej., 2:1) que podrían facilitar excesivamente la tarea. La normalización $[-1, 1]$ se mantuvo también en esta señal.

8.2.4. Ejemplo de las señales utilizadas

La Figura 7 ilustra ejemplos representativos de las señales: (arriba) onda cuadrada a 440 Hz; (centro) onda diente de sierra; (abajo) multiseno de 150 Hz y 280 Hz, todas normalizadas a $[-1, 1]$.

Figura 7. Señales sintéticas empleadas en la etapa de identificación: cuadrada (440 Hz), diente de sierra y multiseno (150 Hz y 280 Hz)



Nota. Elaboración propia.

En conjunto, este diseño de estímulos provee un entorno controlado para medir la capacidad de los modelos de aprender mapeos dinámicos con contenido espectral relevante para audio, antes de abordar la tarea de deconvolución sobre grabaciones reales.

8.3. Metodología experimental

La fase de experimentación con identificación de sistemas se estructuró siguiendo un protocolo de preparación, entrenamiento y validación, diseñado para garantizar la comparabilidad entre los tres modelos evaluados: KLMS, NLARX y TCN. El objetivo de esta metodología fue mantener condiciones homogéneas en el tratamiento de los datos y en los criterios de evaluación, de modo que las diferencias observadas en el desempeño de los modelos respondieran exclusivamente a sus propiedades estructurales y no a factores externos de configuración o procesamiento.

8.3.1. Preparación de los datos de entrenamiento y validación

Todas las señales sintéticas descritas en la Sección 7 se organizaron en pares $(d(t), u(t))$, donde $d(t)$ representa la señal deseada o de referencia y $u(t)$ la señal observada o degradada, simulando la respuesta de un sistema desconocido.

Para emular un entorno no ideal y estudiar la capacidad de los modelos de reconstruir señales perturbadas, se inyectó ruido gaussiano aditivo a la salida del sistema. El ruido se generó con distribución normal de media cero y desviación estándar $\sigma = 2.5$, controlando así el nivel de degradación de la señal medida. De esta forma, cada observación quedó definida como:

$$y(t) = u(t) + n(t), \quad n(t) \sim \mathcal{N}(0, \sigma^2),$$

donde $n(t)$ representa el ruido gaussiano añadido. Este procedimiento permitió analizar la robustez de cada arquitectura frente a perturbaciones de distinta intensidad y su capacidad de aproximar correctamente la señal limpia.

Posteriormente, las secuencias (d, u) se dividieron en subconjuntos de entrenamiento y validación con una proporción del 70/30, siguiendo una segmentación temporal que preserva la continuidad de las señales. Este criterio se mantuvo uniforme en todos los modelos, tanto en los implementados en *MATLAB* (KLMS y NLARX) como en los desarrollados en Python (TCN).

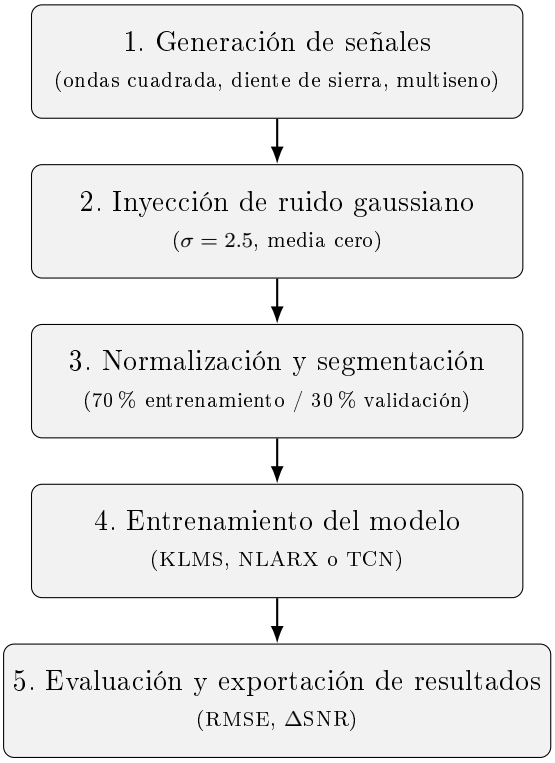
8.3.2. Flujo general de entrenamiento y validación

El procedimiento experimental se diseñó como un *pipeline* secuencial compuesto por cinco etapas principales:

1. **Generación e inyección de ruido:** creación de las señales sintéticas y adición de ruido gaussiano controlado ($\sigma = 2.5$).
2. **Normalización y estructuración:** ajuste de amplitud al rango $[-1, 1]$ y conversión a formato estructurado (`iddata` o tensores).
3. **Segmentación temporal:** división del conjunto completo en bloques de entrenamiento (70 %) y validación (30 %).
4. **Entrenamiento del modelo:** ajuste de parámetros mediante los algoritmos correspondientes (regla adaptativa, regresión no lineal o retropropagación).
5. **Evaluación y exportación:** estimación de la salida $\hat{y}(t)$, cálculo de métricas de desempeño y almacenamiento de resultados.

El flujo completo de este proceso se resume en la Figura 8.

Figura 8. Flujo general de entrenamiento y validación para los experimentos de identificación de sistemas



Nota. Elaboración propia.

8.3.3. Métricas de evaluación

Para cuantificar el desempeño de los modelos se emplearon tres métricas de uso común en tareas de identificación y filtrado de señales:

- **Raíz del error cuadrático medio (RMSE):**

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

  Expresa el error promedio en las mismas unidades de la señal, facilitando su interpretación perceptual.

- **Incremento de relación señal–ruido (ΔSNR):**

$$\Delta\mathrm{SNR} = \mathrm{SNR}_{\mathrm{salida}} - \mathrm{SNR}_{\mathrm{entrada}}$$

  Evalúa la mejora (o deterioro) de la calidad de la señal luego del procesamiento.

Estas métricas permiten realizar comparaciones directas entre modelos con estructuras y dimensiones diferentes, y fueron seleccionadas por su independencia del dominio de implementación.

8.4. Modelo 1: *Kernel Least Mean Squares* (KLMS)

El modelo KLMS se evaluó como primer enfoque dentro del proceso de identificación de sistemas. A diferencia de los otros dos modelos considerados, su operación es estrictamente adaptativa y en línea. Esto implica que, el filtro actualiza sus coeficientes a partir de cada nueva observación, sin una fase de entrenamiento fuera de línea ni una separación entre conjuntos de entrenamiento y validación. Por esta razón, el filtro se aplicó directamente sobre las señales sintéticas diseñadas para esta etapa, utilizándolas tanto para la adaptación como para la evaluación de desempeño.

8.4.1. Configuración del modelo

La implementación del filtro KLMS se realizó utilizando el *Kernel Adaptive Filtering Toolbox* de Van Vaerenbergh [20]. Para garantizar una comparación consistente con el resto de modelos, se fijaron parámetros comunes en las señales de entrada (frecuencia de muestreo de 44.1 kHz, duración de 10 s y amplitud normalizada en el rango $[-1, 1]$). La configuración base específica del filtro fue la siguiente:

- Kernel: gaussiano.
- $\eta = 10^{-4}$: tasa de aprendizaje.
- $\sigma = \sqrt{0.5}$: parámetro de dispersión del *kernel* gaussiano.
- $M = 100$: orden del filtro.

El parámetro M determina cuántas muestras pasadas de la señal ruidosa se utilizan para predecir la salida actual. Valores de M grandes permiten capturar dependencias temporales de mayor longitud, aunque incrementan de forma considerable el costo computacional y pueden dificultar la convergencia del algoritmo.

8.4.2. Resultados del entrenamiento

Durante la etapa de identificación de sistemas, el filtro KLMS mostró un comportamiento altamente dependiente del nivel de ruido agregado a las señales sintéticas. En la primera configuración evaluada (*kernel* gaussiano, $\eta = 10^{-4}$, $\sigma = \sqrt{0.5}$, $M = 100$), la onda cuadrada perturbada con ruido gaussiano de desviación estándar $\sigma_n = 2.5$ no pudo ser reconstruida. El modelo colapsó rápidamente hacia una salida nula, generando una señal prácticamente plana en torno a cero. Esta atenuación completa indica que el filtro no logró identificar la dinámica del sistema bajo niveles de ruido moderados.

La Figura 9 ilustra este comportamiento, donde la salida del modelo permanece suprimida a pesar de la variación abrupta de la onda cuadrada deseada. Las métricas obtenidas para este escenario reflejan esta incapacidad, con valores de error significativamente elevados: $\mathrm{RMSE}_{\mathrm{train}} = 1.00$ y $\mathrm{SNR}_{\mathrm{out}} = 0.00$ dB.

Ante este resultado, se evaluó una segunda configuración en la que se incrementó el orden del filtro a $M = 1000$ y se redujo la dispersión del *kernel* gaussiano a $\sigma = \sqrt{0.2}$. Sin embargo, el comportamiento observado fue esencialmente idéntico: el modelo volvió a colapsar hacia cero. La Figura 10 muestra que incluso con un mayor número de retardos, lo que teóricamente permite capturar relaciones temporales más largas, el filtro no logró aproximar la dinámica de la señal. Las métricas confirmaron nuevamente esta tendencia con $\mathrm{RMSE}_{\mathrm{train}} = 1.00$ y $\mathrm{SNR}_{\mathrm{out}} = 0.00$ dB.

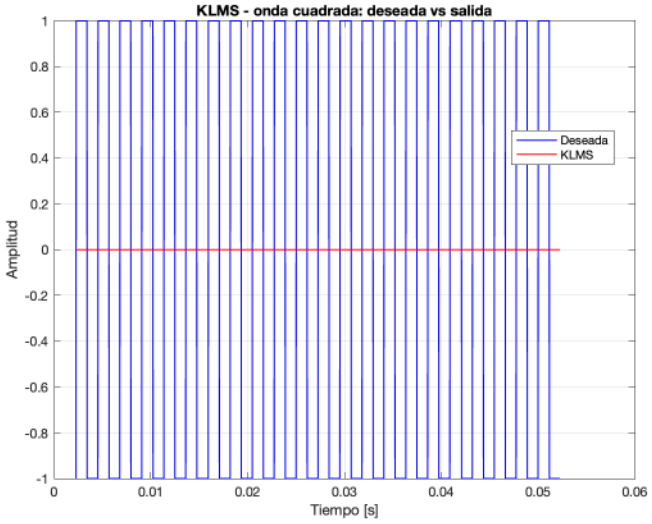
Posteriormente, se modificó el tipo de *kernel* a uno laplaciano manteniendo la configuración anterior ($\eta = 10^{-4}$, $\sigma = \sqrt{0.2}$, $M = 1000$). El objetivo de este ajuste era explorar si la métrica de distancia del *kernel* podía mejorar la sensibilidad del filtro ante discontinuidades pronunciadas. No obstante, el resultado fue nuevamente el mismo, el modelo anuló la señal durante el entrenamiento. La Figura 11 evidencia este patrón. Las métricas obtenidas volvieron a confirmar la ausencia de aprendizaje: $\mathrm{RMSE}_{\mathrm{train}} = 1.00$ y $\mathrm{SNR}_{\mathrm{out}} = 0.00$ dB.

Dado que todos los casos anteriores compartían un denominador común, la imposibilidad del modelo para adaptarse bajo ruido $\sigma_n = 2.5$, se redujo la desviación estándar a $\sigma_n = 0.2$. Bajo este nuevo escenario y utilizando nuevamente la configuración base, el filtro mostró por primera vez capacidad de reconstrucción parcial de la onda cuadrada. La Figura 12 muestra una aproximación ligeramente más cercana a la forma original, aunque aún con errores significativos alrededor de las transiciones y sin recuperar completamente la amplitud. Las métricas fueron coherentes con esta mejora: $\mathrm{RMSE}_{\mathrm{train}} = 0.95$ y $\mathrm{SNR}_{\mathrm{out}} = 0.21$ dB.

Finalmente, al reducir aún más el nivel de ruido a $\sigma_n = 0.1$, el filtro logró generar una reconstrucción visualmente más estable. Aunque el seguimiento de la onda cuadrada mejoró en las regiones constantes, las transiciones abruptas continuaron siendo problemáticas. La Figura 13 muestra este resultado. Las métricas nuevamente reflejaron una ligera mejora: $\mathrm{RMSE}_{\mathrm{train}} = 0.61$ y $\mathrm{SNR}_{\mathrm{out}} = 4.28$ dB.

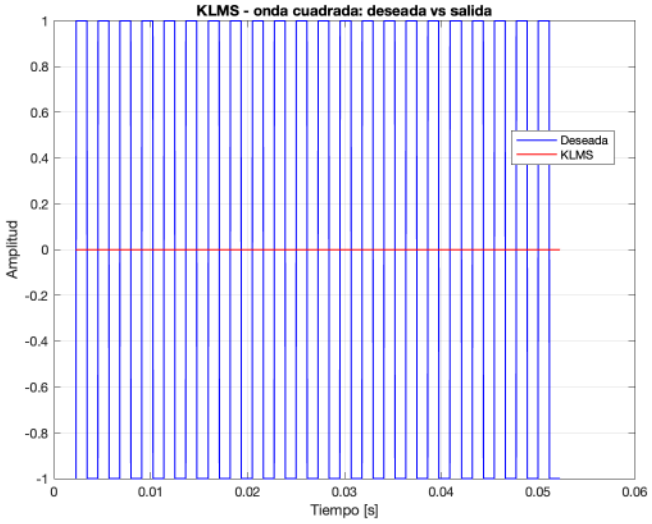
Estos resultados demuestran que el desempeño del filtro KLMS está fuertemente condicionado por la relación señal–ruido. Sólo bajo niveles de ruido muy bajos ($\sigma_n \leq 0.2$) el filtro logra capturar parcialmente la estructura de la señal, confirmando su sensibilidad extrema al ruido aditivo.

Figura 9. Resultado del entrenamiento del modelo KLMS con parámetros base y ruido $\sigma_n = 2.5$



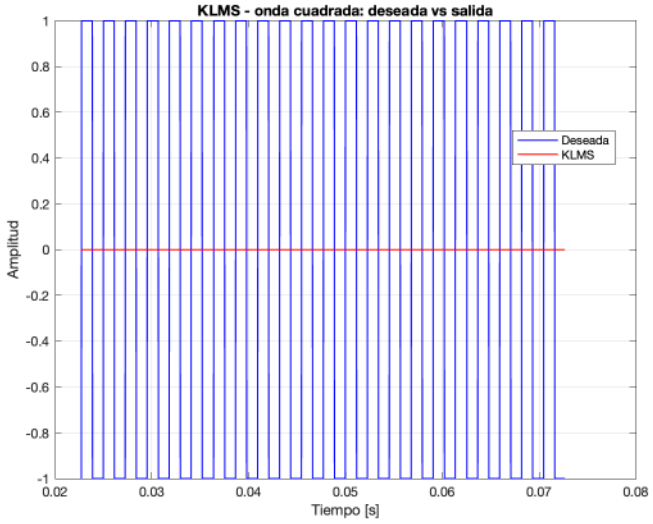
Nota. Elaboración propia.

Figura 10. Resultado del entrenamiento con $M = 1000$ y $\sigma = \sqrt{0.2}$, ruido $\sigma_n = 2.5$



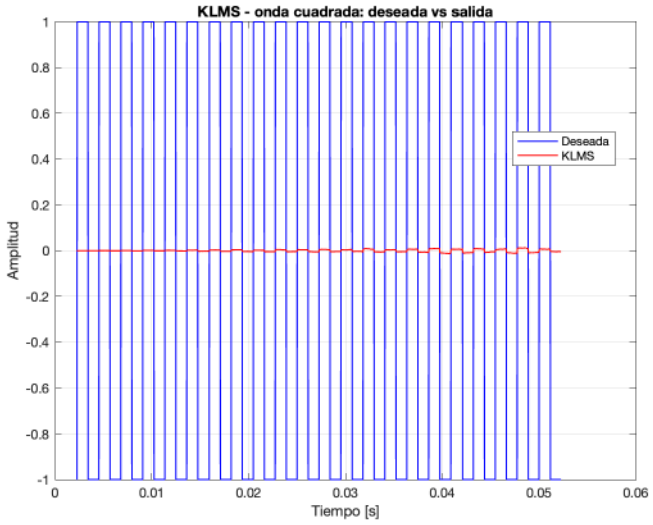
Nota. Elaboración propia.

Figura 11. Entrenamiento con kernel laplaciano, $\eta = 10^{-4}$, $M = 1000$, ruido $\sigma_n = 2.5$



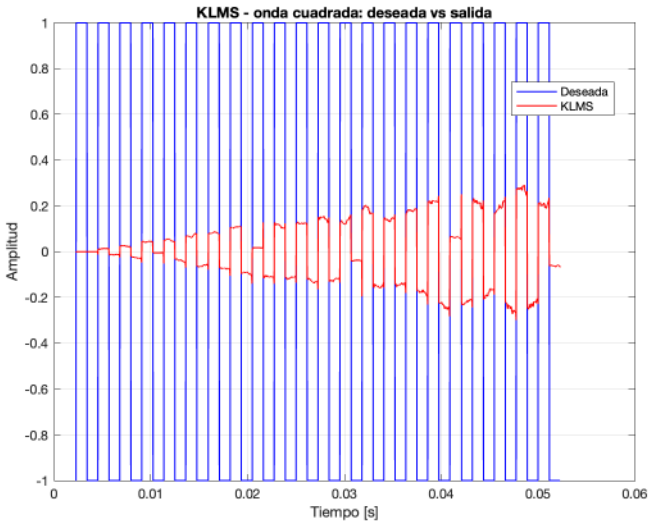
Nota. Elaboración propia.

Figura 12. Reconstrucción parcial con configuración base y ruido reducido $\sigma_n = 0.2$



Nota. Elaboración propia.

Figura 13. Mejor reconstrucción obtenida, con ruido $\sigma_n = 0.1$



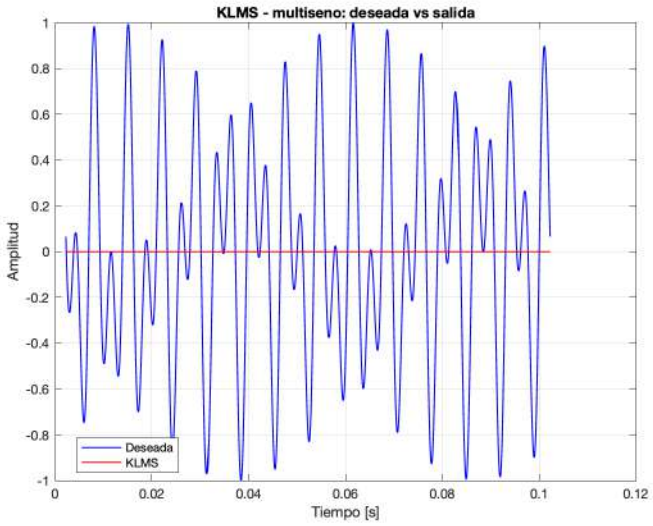
Nota. Elaboración propia.

8.4.3. Validación con señal multiseno

Una vez completado el entrenamiento con la onda cuadrada, se evaluó la capacidad del filtro KLMS para generalizar mediante una señal de validación de tipo multiseno compuesta por dos componentes sinusoidales de 150 Hz y 280 Hz. Esta prueba permite determinar si el modelo fue capaz de aprender una relación entrada–salida que trascienda el patrón específico utilizado durante la etapa de adaptación.

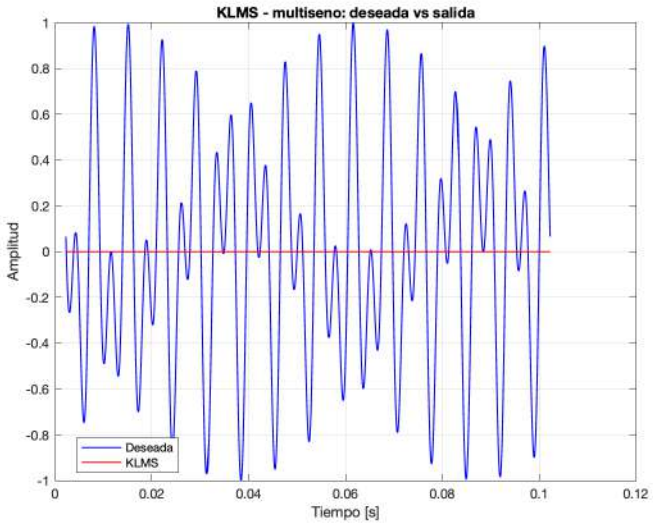
En todos los experimentos realizados, incluyendo aquellos donde la reconstrucción de la onda cuadrada fue parcialmente satisfactoria bajo condiciones de ruido reducido, el modelo colapsó nuevamente hacia cero durante la fase de validación. Este comportamiento se observa claramente en las Figuras 14 y 15, donde la salida del filtro carece de cualquier correspondencia con las componentes sinusoidales presentes en la señal deseada.

Figura 14. Validación con multiseno para el caso con ruido reducido $\sigma_n = 0.2$



Nota. La salida del filtro colapsa a cero a pesar del desempeño moderado en entrenamiento. Elaboración propia.

Figura 15. Validación con multiseno para el caso con ruido reducido $\sigma_n = 0.1$



Nota. El modelo nuevamente no generaliza, anulando completamente la señal. Elaboración propia.

Incluso en los escenarios más favorables, ruido $\sigma_n = 0.1$ y $\sigma_n = 0.2$, el modelo fue incapaz de reproducir alguna de las frecuencias presentes en la señal multiseno, evidenciando que la adaptación lograda durante el entrenamiento no se traduce en una representación

funcional del sistema. Las métricas obtenidas en esta etapa confirman este comportamiento: $\mathrm{RMSE}_{\mathrm{val}} = 0.49$, $\mathrm{SNR}_{\mathrm{in}} = 8.04$ dB y $\Delta\mathrm{SNR} = -8.04$ dB.

En conjunto, estos resultados muestran que, aunque el modelo KLMS puede aproximar parcialmente una onda cuadrada bajo niveles muy bajos de ruido, su capacidad de generalización es prácticamente nula. Esto sugiere que la representación interna generada por el filtro durante la fase adaptativa es altamente dependiente del nivel de ruido, del tipo de estímulo y de la estructura espectral de la señal de entrada, limitando severamente su aplicabilidad en tareas más complejas.

8.4.4. Discusión

Los resultados obtenidos muestran que el filtro KLMS, bajo las configuraciones evaluadas, no es adecuado para la tarea de identificación dinámica planteada. Uno de los problemas más evidentes es su alta susceptibilidad al ruido: únicamente logra aproximar la señal cuando el nivel de perturbación es extremadamente bajo ($\sigma_n \leq 0.2$). En escenarios más realistas, el algoritmo colapsa rápidamente hacia una salida cercana a cero, lo que impide cualquier reconstrucción útil de la señal original.

Otro aspecto que limita su desempeño es la fuerte dependencia de los parámetros del *kernel*. La modificación de la dispersión o incluso del tipo de *kernel* no produjo mejoras sustanciales, lo que sugiere que el problema no radica únicamente en la función núcleo seleccionada, sino en la forma en que el algoritmo adapta sus coeficientes muestra a muestra. Esta naturaleza estrictamente adaptativa, si bien es útil en otros contextos, dificulta la estabilidad cuando la señal ruidosa presenta variaciones abruptas o contenido espectral amplio.

Finalmente, incluso en los casos en que consiguió reconstruir parcialmente la onda cuadrada bajo ruido reducido, el filtro no logró generalizar hacia la señal multiseno empleada en la validación. La salida volvió a colapsar, indicando que lo aprendido durante la adaptación no constituye una representación funcional del sistema. Esta falta de generalización limita severamente su aplicabilidad como identificador dinámico.

En conjunto, los resultados permiten concluir que, aunque KLMS puede capturar algunas estructuras simples en condiciones ideales, su utilidad en entornos ruidosos es muy baja. La sensibilidad extrema al ruido, la dependencia crítica de los parámetros del *kernel* y la ausencia de generalización lo posicionan como el método menos robusto entre los evaluados en esta etapa de experimentación.

8.5. Modelo 2: *Temporal Convolutional Networks* (TCN)

En esta etapa se evaluó un modelo basado en *Temporal Convolutional Networks* (TCN) para la tarea de identificación de sistemas. A diferencia del filtro KLMS, la TCN se entrena de forma fuera de línea a partir de ventanas de la señal, utilizando optimización basada en gradiente y un esquema explícito de entrenamiento y validación. El objetivo fue aprender un mapeo entre una entrada ruidosa y su correspondiente salida limpia, bajo condiciones controladas de ruido gaussiano aditivo.

El modelo se implementó en *PyTorch* y se entrenó utilizando las señales de entrenamiento y validación generadas según se describió en la Sección 8.2 (onda cuadrada y señal multiseno con ruido gaussiano). La arquitectura y los hiperparámetros se definieron con el propósito de equilibrar capacidad de representación, estabilidad numérica y costo computacional.

8.5.1. Configuración de la arquitectura

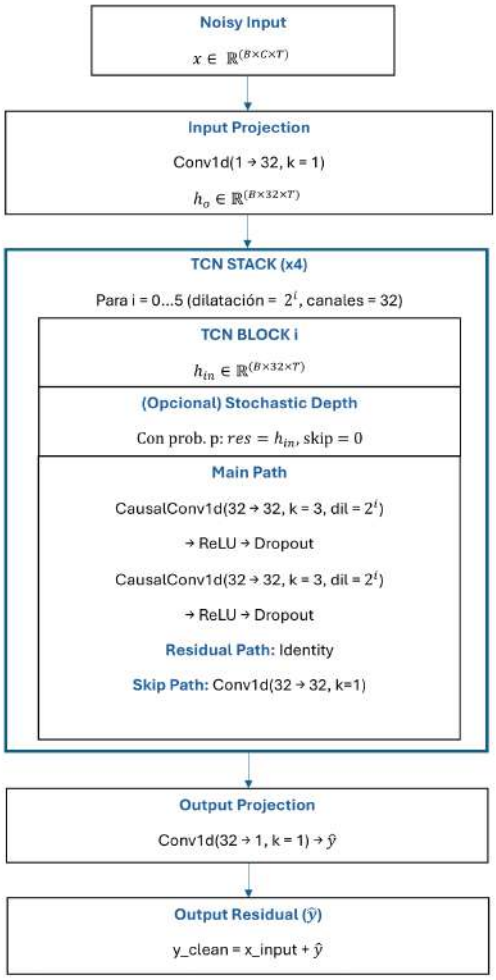
La red TCN utilizada recibe como entrada secuencias unidimensionales de longitud $L = 2048$ muestras, organizadas con forma $[B, 1, L]$, donde B es el tamaño del lote (*batch*). La arquitectura está compuesta por seis bloques convolucionales causales con dilataciones crecientes, seguidos de una capa de proyección final. En términos generales, el modelo se configuró de la siguiente forma:

- Canal de entrada: $c_{\text{in}} = 1$.
- Canal de salida: $c_{\text{out}} = 1$.
- Número de bloques TCN: $n_{\text{blocks}} = 4$.
- Canales internos: $c = 32$ filtros por bloque.
- Tamaño de kernel: $k = 5$.
- Dilataciones: $d_i = 2^i$ para $i = 0, \ldots, 5$, es decir $d \in \{1, 2, 4, 8, 16, 32\}$.
- Función de activación: *ReLU*.
- Regularización: *dropout* con probabilidad $p = 0.1$.
- Normalización de pesos: *weight normalization* aplicada a las convoluciones internas.

Cada bloque TCN está formado por dos convoluciones causales en serie, ambas con el mismo tamaño de *kernel* y dilatación. La salida de estas convoluciones se combina mediante una conexión residual con la entrada del bloque, seguida de una activación *ReLU*. Cuando el número de canales de entrada y salida difiere, se utiliza una convolución 1×1 como atajo residual para igualar dimensiones. La operación es estrictamente causal gracias al uso de ceros a la izquierda (*left padding*), lo que garantiza que la predicción en el tiempo n dependa únicamente de muestras anteriores o iguales a n.

La Figura 16 ilustra la arquitectura general empleada para la identificación de sistemas.

Figura 16. Arquitectura general de la red TCN utilizada para la identificación de sistemas



Nota. Elaboración propia.

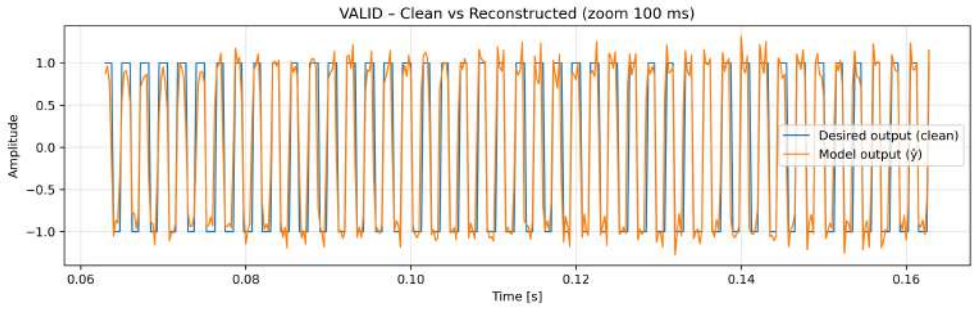
8.5.2. Resultados del entrenamiento

Se realizaron cuatro experimentos para evaluar el comportamiento del modelo TCN bajo diferentes configuraciones de hiperparámetros y diferentes condiciones de generación de señales. En los Experimentos 1 y 2, el entrenamiento se llevó a cabo utilizando una onda cuadrada tal como se describe en la Sección 8.2. Para los Experimentos 3 y 4, se modificó deliberadamente la señal de entrenamiento, utilizando una onda cuadrada mucho más lenta (8 Hz) y de mayor amplitud (± 5), con el fin de evaluar la robustez del modelo frente a una distribución distinta a la utilizada inicialmente.

En todos los casos, la TCN logró aprender patrones relevantes de la onda cuadrada, aunque con diferencias en precisión y estabilidad según el conjunto de entrenamiento empleado.

Experimento 1 (LR $= 10^{-3}$, *kernel* = 3, 40 épocas). El modelo registró una mejora significativa en la relación señal–ruido durante la validación, pasando de $\mathrm{SNR}_{\mathrm{in}} = -7.99$ dB a $\mathrm{SNR}_{\mathrm{out}} = 9.68$ dB, equivalente a una ganancia de 17.67 dB. El error obtenido fue $\mathrm{RMSE}_{\mathrm{valid}} = 0.3281$, indicando una reconstrucción estable de la onda cuadrada.

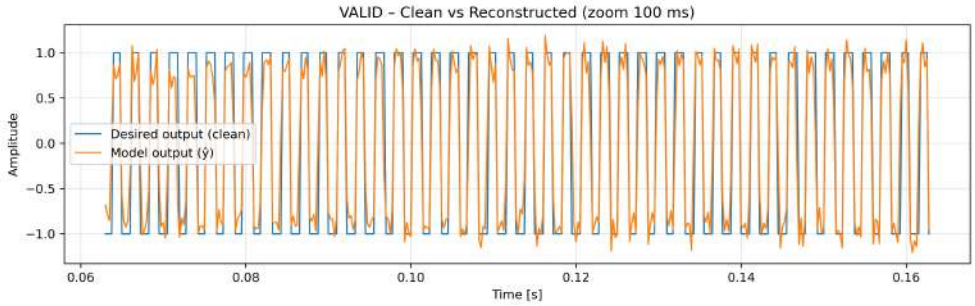
Figura 17. Reconstrucción de la onda cuadrada en validación, Experimento 1



Nota. Elaboración propia.

Experimento 2 (LR $= 10^{-4}$, *kernel* = 5, 60 épocas). Mostró un desempeño similar, aunque ligeramente inferior al del Experimento 1. La mejora lograda fue $\Delta\mathrm{SNR} = 17.48$ dB con $\mathrm{RMSE}_{\mathrm{valid}} = 0.3334$. El incremento en el tamaño del *kernel* suaviza las transiciones, lo cual coincide con la forma de la salida reconstruida.

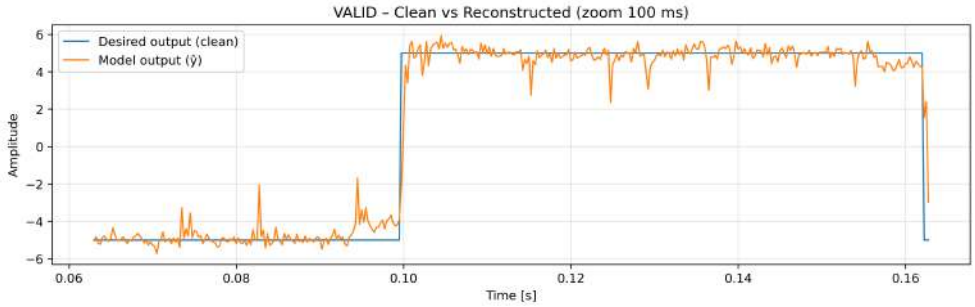
Figura 18. Reconstrucción de la onda cuadrada en validación, Experimento 2



Nota. Elaboración propia.

Experimento 3 (LR $= 10^{-3}$, *kernel* = 3, 40 épocas). En este experimento se entrenó la TCN con una onda cuadrada distinta, de 8 Hz y amplitud 5, mucho más lenta y con mayor separación entre transiciones. Esta modificación metodológica incrementó la dificultad para la red, reflejándose en una menor ganancia de $\Delta\mathrm{SNR} = 11.39$ dB y un error $\mathrm{RMSE}_{\mathrm{valid}} = 0.8098$.

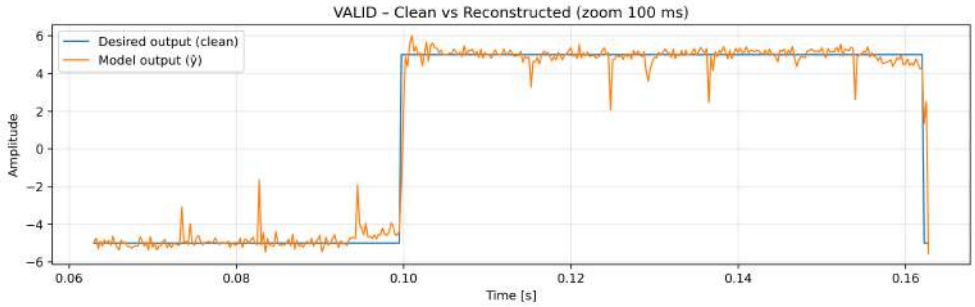
Figura 19. Reconstrucción de la onda cuadrada en validación, Experimento 3



Nota. Elaboración propia.

Experimento 4 (LR = 10^{-4}, *kernel* = 5, 60 épocas). Con el mismo conjunto de entrenamiento modificado del Experimento 3, la TCN logró ΔSNR = 12.36 dB y $\text{RMSE}_{\text{valid}}$ = 0.7244, mostrando una ligera mejora respecto al modelo previo aunque manteniendo la misma tendencia general.

Figura 20. Reconstrucción de la onda cuadrada en validación, Experimento 4



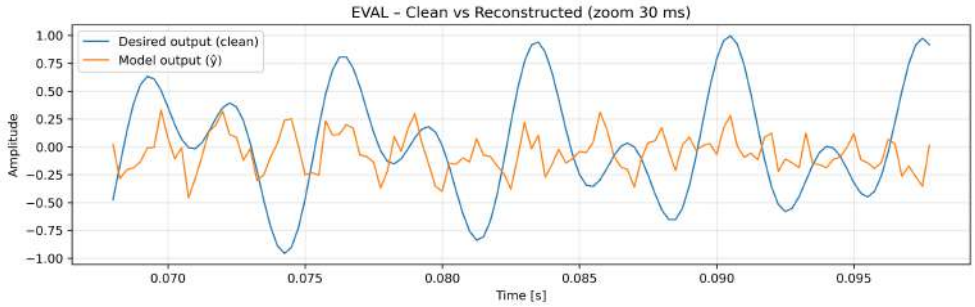
Nota. Elaboración propia.

8.5.3. Validación con señal multiseno

Para analizar la capacidad de generalización de los modelos entrenados, se evaluó cada experimento utilizando una señal multiseno contaminada con ruido gaussiano. En los Experimentos 1 y 2, dicha señal fue la combinación de dos tonos (150 Hz y 280 Hz). En los Experimentos 3 y 4, se modificó el procedimiento de generación, utilizando un multiseno aleatorio compuesto por cuatro tonos distribuidos entre 100 y 200 Hz, con amplitudes y fases aleatorias. Este cambio intencional expone al modelo a un contenido espectral más complejo y diferente del observado durante entrenamiento.

Experimento 1. El modelo alcanzó una mejora de ΔSNR = 4.97 dB, con un error $\text{RMSE}_{\text{eval}} = 0.5643$. La TCN logró reducir el ruido, aunque la forma de onda reconstruida muestra distorsión en comparación con el multiseno original.

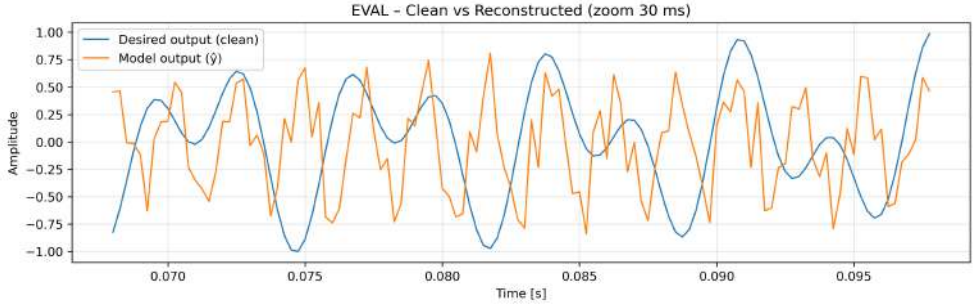
Figura 21. Reconstrucción de la señal multiseno en evaluación, Experimento 1



Nota. Elaboración propia.

Experimento 2. La ganancia observada fue de $\Delta\text{SNR} = 11.66$ dB, mientras que el error fue $\text{RMSE}_{\text{eval}} = 0.6536$. A pesar de la mayor ganancia en SNR, la reconstrucción presenta suavizado excesivo y pérdida parcial de componentes armónicas.

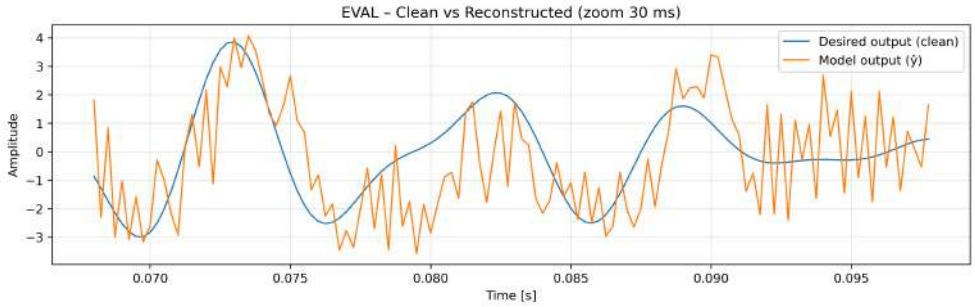
Figura 22. Reconstrucción de la señal multiseno en evaluación, Experimento 2



Nota. Elaboración propia.

Experimento 3. Con la nueva señal multiseno aleatoria, la mejora fue moderada: $\Delta\text{SNR} = 4.22$ dB, acompañada de un error elevado ($\text{RMSE}_{\text{eval}} = 1.8455$). Esto indica que el modelo no logró extrapolar con precisión hacia señales fuera de la distribución vista en entrenamiento.

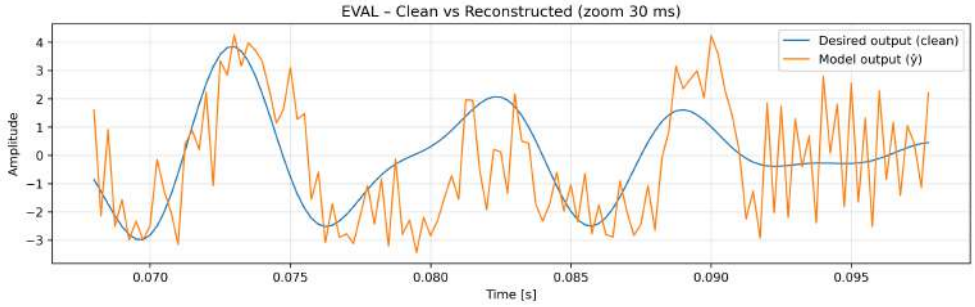
Figura 23. Reconstrucción de la señal multiseno en evaluación, Experimento 3



Nota. Elaboración propia.

Experimento 4. El comportamiento fue similar al anterior, con $\Delta\text{SNR} = 3.61$ dB y $\text{RMSE}_{\text{eval}} = 1.9807$. La mayor distorsión se asocia a la complejidad del multiseno aleatorio y a la fuerte diferencia entre las señales de entrenamiento y validación.

Figura 24. Reconstrucción de la señal multiseno en evaluación, Experimento 4



Nota. Elaboración propia.

8.5.4. Discusión

Los resultados obtenidos muestran que la TCN es capaz de aprender y reconstruir con buena fidelidad señales periódicas con transiciones abruptas, como la onda cuadrada utilizada en los experimentos iniciales. La red produjo mejoras significativas de SNR y valores bajos de RMSE en la fase de validación, especialmente cuando la señal de entrenamiento poseía alta frecuencia y amplitud moderada.

Sin embargo, la capacidad de generalización se vio limitada al evaluar los modelos con señales multiseno. Aunque la TCN logró aumentar la SNR en todos los casos, los errores RMS fueron considerablemente mayores, reflejando la dificultad del modelo para capturar dinámicas suaves y múltiples componentes frecuenciales no observadas durante entrenamiento. La degradación fue más pronunciada en los Experimentos 3 y 4, donde tanto la onda cuadrada como el multiseno fueron modificados, alterando sustancialmente la distribución de las señales de entrada.

En conjunto, estos resultados indican que, si bien la TCN presenta buen desempeño como identificador de sistemas para señales periódicas y altamente estructuradas, su capacidad de extrapolación hacia señales con espectro más complejo es limitada.

8.6. Modelo 3: modelo AutoRegresivo No Lineal con Entrada Exógena (NLARX)

El tercer enfoque explorado en esta investigación corresponde a los modelos AutoRegresivos No Lineales con Entrada Exógena (NLARX). A diferencia de los modelos KLMS y TCN, cuyo entrenamiento requiere conjuntos explícitos de entrada ruidosa y salida limpia, el modelo NLARX se ajusta directamente a partir de las señales disponibles mediante los algoritmos internos de la *System Identification Toolbox*. En este enfoque, la dinámica del sistema se modela mediante regresores construidos a partir de retardos de entrada y salida, combinados con una función no lineal que actúa como aproximador universal.

El objetivo de esta etapa fue evaluar la capacidad del modelo NLARX para reconstruir la señal limpia a partir de una observación contaminada, bajo diferentes configuraciones de regresores y funciones de no linealidad. Dado que su formulación permite capturar relaciones dinámicas no lineales de forma explícita, este modelo constituye un punto de comparación importante frente a los métodos basados en filtros adaptativos y redes neuronales convolucionales temporales.

8.6.1. Configuración base del modelo NLARX

El modelo NLARX utilizado en esta etapa se configuró empleando un conjunto de datos generado de forma controlada mediante una onda cuadrada de 200 Hz y amplitud 3, contaminada con ruido gaussiano aditivo ($\sigma = 2.5$). Esta señal se dividió siguiendo una proporción 70/30 para conformar los conjuntos de entrenamiento y validación. Las secuencias fueron incorporadas al entorno de trabajo utilizando objetos `iddata` del *System Identification Toolbox* de MATLAB, lo que permitió mantener un control estricto sobre las características espectrales y estadísticas de los estímulos aplicados al sistema.

La estructura base del NLARX incluyó retardos de entrada y salida comprendidos entre 1 y 6 muestras. Esta configuración permite capturar dependencias dinámicas de corto plazo entre la señal ruidosa y la respuesta esperada del sistema. El Cuadro 3 resume esta configuración inicial, utilizada como punto de partida antes de explorar variaciones adicionales en los experimentos posteriores.

Cuadro 3. Retardos base utilizados en el modelo NLARX

Componente	Tipo	Retardos
Entrada	$u(t)$	1:6
Salida	$y(t)$	1:6

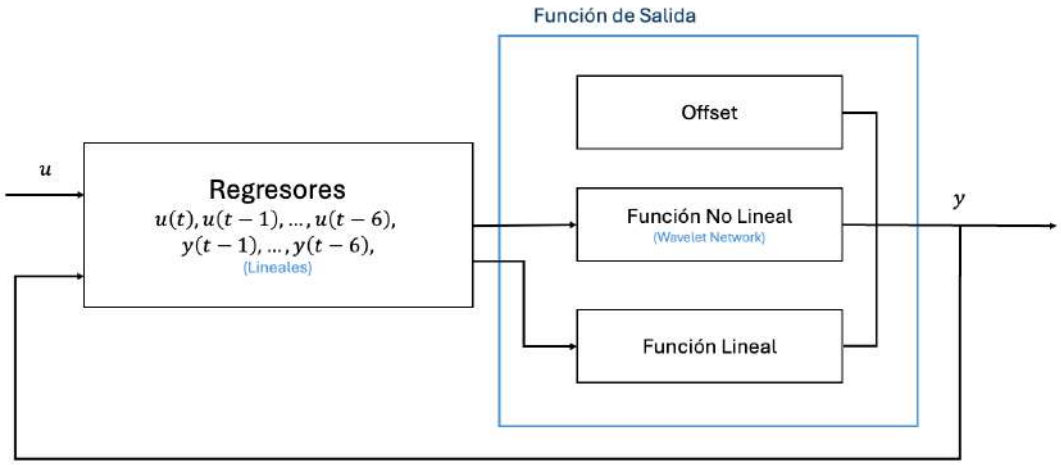
Nota. Elaboración propia.

Para modelar la no linealidad del sistema se seleccionó una red *wavelet*, la cual es adecuada para capturar relaciones no lineales localizadas y suavemente cambiantes. El número de unidades de esta red se determinó mediante la opción de selección automática incluida en el *System Identification Toolbox* de MATLAB, habilitando mecanismos internos de regularización que ajustan la complejidad del modelo en función de los datos.

En los experimentos 3, 4 y 5 se modificaron las características de las señales de entrada para analizar la sensibilidad del modelo. La onda cuadrada de entrenamiento se ajustó a 10 Hz, mientras que la señal de validación fue una multiseno compuesta por 5 Hz y 10 Hz. Esta configuración resultó ser la que mejor correspondió con la capacidad de reconstrucción del modelo bajo condiciones de ruido elevado.

La Figura 25 muestra un esquema conceptual de la arquitectura empleada en los experimentos.

Figura 25. Diagrama conceptual de la arquitectura base del modelo NLARX



Nota. Elaboración propia.

8.6.2. Resultados del entrenamiento

A diferencia de los modelos KLMS y TCN, cuyo proceso de entrenamiento fue implementado manualmente en *MATLAB* y *Python*, respectivamente, el modelo NLARX fue estimado utilizando directamente la interfaz gráfica del *System Identification Toolbox* de *MATLAB*. En este entorno, la selección de regresores, el ajuste de parámetros y la calibración de la función no lineal se realizan mediante los algoritmos internos de la herramienta, sin exponer al usuario curvas de aprendizaje, pérdidas por época o métricas intermedias del proceso de optimización.

Como consecuencia, no es posible obtener una gráfica de "entrenamiento" comparable a las utilizadas en KLMS o TCN. En su lugar, el flujo estándar de NLARX consiste en evaluar

el modelo únicamente a través de la comparación entre la respuesta simulada y la señal real, empleando métricas como el error de simulación y la coherencia entre la dinámica estimada y la señal de referencia.

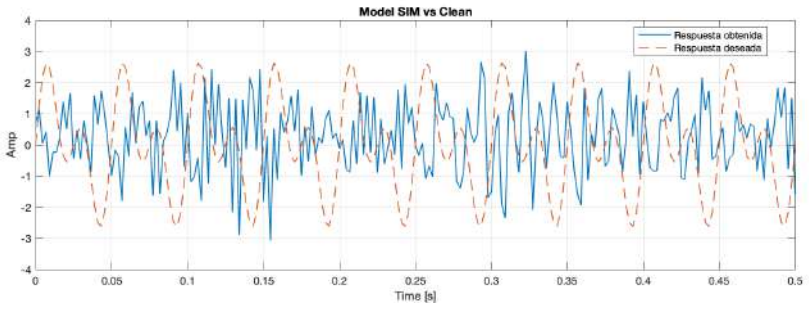
Por esta razón, en esta sección se presentan directamente los resultados producidos por el modelo NLARX frente a la señal limpia, lo cual corresponde a la forma habitual de evaluación en este tipo de modelos. Es importante destacar que la ausencia de curvas explícitas de entrenamiento no afecta la validez del análisis, ya que el método NLARX no utiliza un proceso iterativo visible para el usuario, sino una estimación basada en mínimos cuadrados y funciones *wavelet* ajustadas localmente. El criterio principal de desempeño es, por tanto, la capacidad del modelo estimado para reproducir adecuadamente el comportamiento del sistema.

8.6.3. Resultados del modelo NLARX

A continuación se presentan los cinco experimentos realizados, acompañados de las métricas de error cuadrático medio (RMSE) y relación señal–ruido (SNR), evaluadas tanto sobre la señal ruidosa inicial como sobre la salida simulada del modelo. Las figuras asociadas muestran la comparación entre la respuesta obtenida y la señal limpia.

Experimento 1: configuración base (200 Hz, amplitud 3, regresores 1:6). En la configuración base los resultados muestran una mejora modesta respecto a la señal ruidosa. Aunque el modelo atenúa parte del ruido, la simulación conserva distorsiones significativas, reflejadas en un RMSE de 2.2268 y una SNR de -3.43 dB, apenas superiores al caso ruidoso.

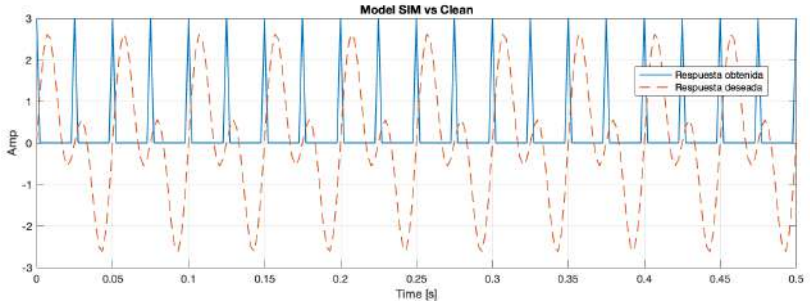
Figura 26. NLARX — Resultados experimento 1



Nota. Elaboración propia.

Experimento 2: aumento a 10 regresores (observación de sobreajuste). Al incrementar los retardos de entrada y salida a diez muestras, el modelo muestra una reducción en RMSE (1.7551) y una mejora moderada en SNR (-1.36 dB). Sin embargo, la simulación evidencia un sobreajuste notable a la forma de la onda cuadrada, perdiendo capacidad de generalización. Esto coincide con las oscilaciones abruptas observadas en la respuesta simulada dentro de la *System Identification App*.

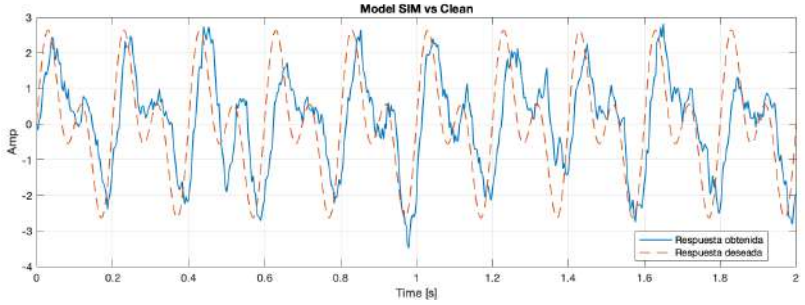
Figura 27. NLARX — Resultados experimento 2



Nota. Elaboración propia.

Experimento 3: bajas frecuencias con regresores 1:6 (10 Hz y multiseno 5–10 Hz). Esta configuración resultó ser la más efectiva. El modelo logra una reconstrucción más fiel, con un RMSE de 1.3503 y una SNR positiva de 0.91 dB, indicando que la simulación consigue superar el nivel de ruido presente en la señal de entrada. La variabilidad más lenta de la onda cuadrada y la multiseno de validación parece favorecer la capacidad del modelo para capturar la dinámica no lineal sin incurrir en sobreajuste.

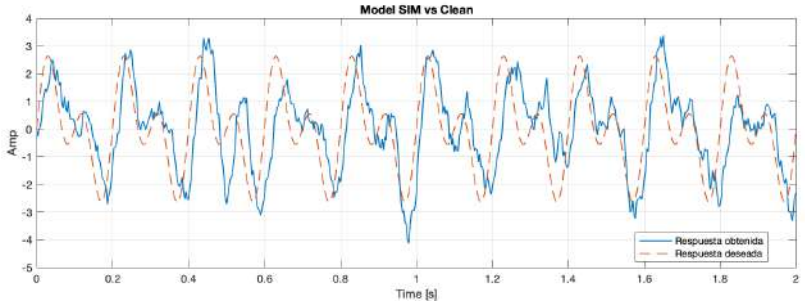
Figura 28. NLARX — Resultados experimento 3



Nota. Elaboración propia.

Experimento 4: bajas frecuencias con regresores reducidos (salida 1:4, entrada 1:2). Con un número menor de regresores el modelo mantiene una mejora significativa respecto al caso ruidoso, con un RMSE de 1.3738 y SNR de 0.76 dB. Aunque el desempeño es ligeramente inferior al del experimento 3, el resultado confirma que un modelo más compacto puede ser suficiente para capturar la dinámica relevante, evitando la complejidad innecesaria.

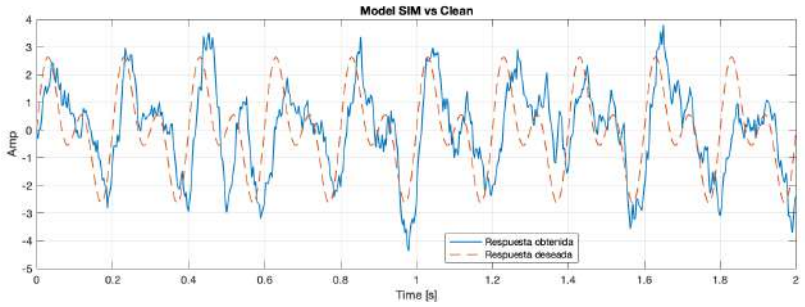
Figura 29. NLARX — Resultados experimento 4



Nota. Elaboración propia.

Experimento 5: bajas frecuencias con regresores 1:10 (modelo más complejo). El aumento de regresores no produjo mejoras adicionales. Por el contrario, el RMSE se incrementó a 1.4176 y la SNR se redujo a 0.49 dB. El modelo tiende a sobredimensionarse sin capturar mejor la dinámica subyacente, lo que sugiere que la complejidad excesiva no aporta beneficios cuando la estructura del sistema es relativamente simple y el ruido domina el comportamiento observado.

Figura 30. NLARX — Resultados experimento 5



Nota. Elaboración propia.

En conjunto, los cinco experimentos confirman que el modelo NLARX presenta una capacidad robusta para reconstruir la señal limpia bajo condiciones de ruido elevado. El mejor rendimiento se obtuvo con señales de baja frecuencia y una configuración equilibrada de regresores (experimento 3), mientras que los modelos excesivamente grandes no aportaron mejoras y, en algunos casos, redujeron la capacidad de generalización. Estos resultados posicionan al NLARX como el método con mejor desempeño dentro del conjunto de modelos evaluados en este trabajo.

8.7. Discusión comparativa de los modelos

La etapa de identificación de sistemas se concibió como un banco de pruebas controlado para contrastar el comportamiento de tres enfoques representativos: filtros adaptativos con *kernel* (KLMS), redes convolucionales temporales (TCN) y modelos AutoRegresivos No Lineales con Entrada Exógena (NLARX). Aunque esta fase no forma parte de los objetivos específicos del trabajo de graduación, los resultados obtenidos permiten extraer conclusiones relevantes sobre la capacidad de cada modelo para reconstruir señales degradadas por ruido gaussiano y sobre su potencial de generalización hacia estímulos distintos a los usados en entrenamiento.

En primer lugar, el filtro KLMS mostró el desempeño más limitado. Bajo niveles de ruido realistas ($\sigma_n = 2.5$), el algoritmo colapsó sistemáticamente hacia una salida cercana a cero, sin lograr aprender la dinámica de la onda cuadrada. Sólo cuando el ruido se redujo drásticamente ($\sigma_n \leq 0.2$) el modelo consiguió una reconstrucción parcial, con mejoras modestas en las métricas ($\text{SNR}_{\text{out}} \approx 4.28$ dB) y errores todavía elevados. Más crítico aún, en la validación con la señal multiseno el filtro volvió a anular la salida, produciendo una degradación neta de la calidad ($\Delta\text{SNR} \approx -8.04$ dB). Estos resultados indican que la representación aprendida por KLMS es extremadamente sensible al ruido y al tipo de estímulo, con una capacidad de generalización prácticamente nula en el contexto considerado.

El modelo basado en *Temporal Convolutional Networks* ofreció un salto cualitativo respecto a KLMS. En los experimentos de la fase de validación, entrenados con ondas cuadradas contaminadas con ruido gaussiano, la TCN alcanzó ganancias de relación señal–ruido del orden de $\Delta\text{SNR} \approx 17{-}19$ dB y errores bajos ($\text{RMSE}_{\text{valid}} \approx 0.28{-}0.33$) en las configuraciones con mayor tasa de aprendizaje y kernels pequeños. Estas cifras reflejan una capacidad sólida para aprender mapeos no lineales causales cuando la distribución de entrenamiento y validación es similar. Sin embargo, al evaluar con señales multiseno la mejora fue más modesta: en el mejor caso se obtuvo $\Delta\text{SNR} \approx 11.66$ dB con $\text{RMSE}_{\text{eval}} \approx 0.65$, mientras que en los experimentos con señales de bajas frecuencias y multiseno aleatorio, el error aumentó de forma notable ($\text{RMSE}_{\text{eval}} > 1.8$). Las formas de onda reconstruidas muestran suavizado excesivo, evidenciando que la TCN generaliza razonablemente dentro de la distribución vista en entrenamiento, pero presenta dificultades para extrapolar a contenidos espectrales más complejos o distribuciones claramente distintas.

Por su parte, el modelo NLARX se posicionó como el enfoque con comportamiento más robusto dentro de este entorno de experimentación. Trabajando directamente con las secuencias completas a través de la *System Identification Toolbox* de MATLAB, y utilizando una función de salida no lineal basada en redes *wavelet* con selección automática de unidades, el NLARX logró mejorar de forma consistente tanto la RMSE como la SNR en todos los experimentos. El caso más representativo corresponde al Experimento 3, donde, bajo ruido elevado y señales de baja frecuencia, la salida simulada pasó de una $\text{SNR}_{\text{noisy}} = -4.49$ dB a una $\text{SNR}_{\text{sim}} = 0.91$ dB, con una reducción del error desde $\text{RMSE}_{\text{noisy}} = 2.5152$ hasta $\text{RMSE}_{\text{sim}} = 1.3503$. Aunque la ganancia en SNR es numéricamente menor a la reportada por la TCN en algunos casos, la inspección visual de las formas de onda muestra una reconstrucción más estable y coherente con la dinámica original, especialmente cuando se utilizan configuraciones equilibradas de regresores (1:6) y señales de prueba de baja frecuencia. Además, los experimentos con modelos más complejos (regresores 1:10) evidenciaron

que el aumento indiscriminado de parámetros no aporta beneficios e incluso puede degradar el desempeño, lo que refuerza la importancia de un diseño parsimonioso.

La fase de identificación de sistemas confirma que los modelos basados en estructuras no lineales explícitas y regresores temporales (NLARX y, en menor medida, TCN) son más adecuados que los filtros adaptativos *kernelizados* tradicionales para abordar tareas de reconstrucción en entornos ruidosos. Esta evidencia respalda la decisión de dar mayor peso a enfoques no lineales de tipo NLARX y TCN en la siguiente etapa de deconvolución acústica con grabaciones reales.

CAPÍTULO 9

Deconvolución acústica con grabaciones reales de audio

Este capítulo presenta la evaluación de los tres métodos considerados en esta investigación, KLMS, TCN y NLARX, aplicados sobre grabaciones reales. A diferencia del capítulo anterior, donde se emplearon señales sintéticas para comprender el comportamiento base de cada modelo bajo condiciones controladas, aquí se analiza su desempeño frente a escenarios propios de una grabación musical real, incluyendo reverberación, interacción acústica con el recinto y variabilidad interpretativa. Esta etapa es esencial para determinar la capacidad práctica de cada método para recuperar señales limpias a partir de material capturado en entornos no ideales.

La evaluación de los modelos no siguió un esquema uniforme, debido a que cada enfoque presentó requerimientos y comportamientos particulares que demandaron estrategias de análisis distintas. En el caso del filtro KLMS, una sola grabación fue suficiente para mostrar que el método no se comporta de manera estable en señales largas, ya que sus ajustes internos cambian con el tiempo y terminan degradando la señal recuperada, lo cual resulta incompatible con un modelo de sistema acústico estático. La arquitectura TCN se evaluó utilizando un conjunto coherente de grabaciones que permitiera una comparación directa con resultados previos dentro de esta línea de investigación. Finalmente, el modelo NLARX se sometió al mismo conjunto de pruebas que la TCN, lo que permitió identificar que su respuesta operaba principalmente como un filtro pasabajas y no como un mecanismo efectivo de deconvolución.

Este capítulo ofrece un análisis crítico del desempeño real de los tres métodos, destacando sus fortalezas, sus limitaciones y el papel que cada uno podría desempeñar en el futuro de esta investigación.

9.1. Modelo 1: *Kernel Least Mean Square* KLMS

9.1.1. Configuración del modelo

El primer enfoque evaluado con grabaciones reales fue el filtro KLMS (*kernel least-mean-square*), implementado en MATLAB utilizando la *Kernel Adaptive Filtering Toolbox* (KAFbox) de Van Vaerenbergh [20]. Este modelo corresponde a un filtro adaptativo no lineal que opera en línea y actualiza sus coeficientes muestra a muestra en un espacio de características inducido por un núcleo de tipo Gaussiano o Laplaciano, tal como se describió en la metodología general de filtros adaptativos no lineales.

Para el análisis con grabaciones reales se trabajó con la base de datos consolidada creada en esta investigación, empleando como caso representativo el par de señales `Ja_AtlanticLimited_Recorded.wav` (entrada reverberada) y `Ja_AtlanticLimited_Original.wav` (referencia limpia). Ambas señales se alinearon temporalmente por correlación cruzada y se recortaron a una longitud común, siguiendo el mismo criterio aplicado en el capítulo de metodología.

La configuración del filtro consideró un paso de adaptación fijo $\eta = 10^{-4}$ y órdenes del regresor en el rango $M \in \{10, 100, 1000\}$, con longitudes efectivas de procesamiento hasta $N = 100{,}000$ muestras. En el espacio de características se exploraron principalmente núcleos Gaussianos y Laplacianos con diferentes valores de dispersión σ, priorizando combinaciones representativas como:

- Kernel Gaussiano con $\sigma = 1$, $M = 10$ y $N = 10{,}000$ muestras.
- Kernel Gaussiano con $\sigma \approx 0.707$, $M = 100$ y $N = 100{,}000$ muestras.
- Kernel Laplaciano con $\sigma = 1$, $M = 10$ y $N = 10{,}000$ muestras.
- Kernel Laplaciano con $\sigma \approx 0.707$, $M = 100$ y $N = 100{,}000$ muestras.

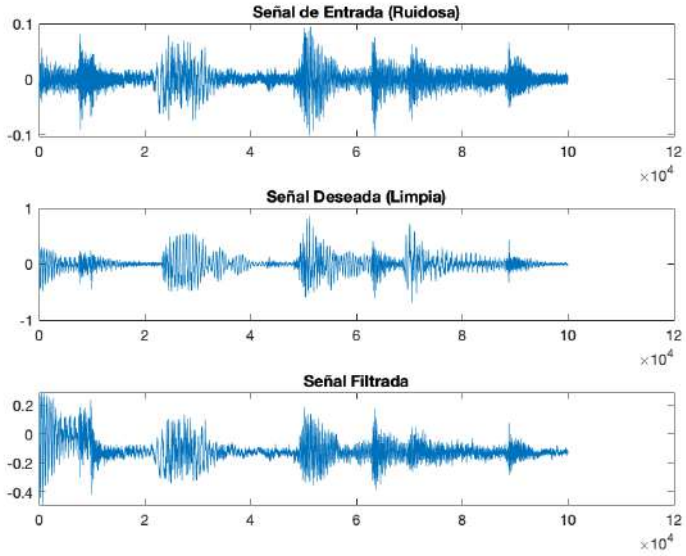
Para cada configuración seleccionada se generaron tres tipos de salidas: una gráfica de amplitud en el dominio del tiempo, un panel comparativo entre las señales deseada, de entrada y estimada, y un conjunto de espectrogramas para la entrada reverberada, la referencia limpia y la salida del filtro. Estas figuras se emplean en esta sección únicamente de forma ilustrativa, con el fin de mostrar el comportamiento típico del KLMS ante una grabación musical real.

9.1.2. Resultados

A continuación se presentan los tres casos representativos seleccionados para ilustrar el comportamiento del filtro KLMS con grabaciones reales. Estos resultados sintetizan las tendencias observadas en la exploración inicial, priorizando configuraciones que permiten comparar estabilidad, calidad de reconstrucción y respuesta temporal ante señales de larga duración.

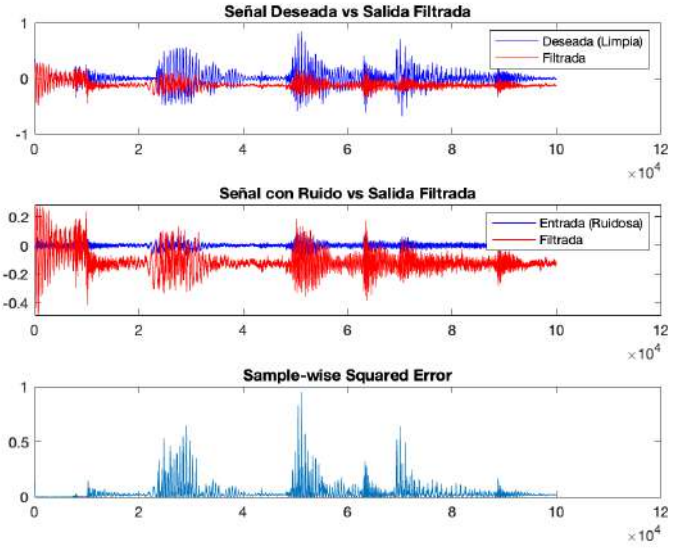
Kernel Gaussiano ($\sigma = 0.707$), $M = 100$, $N = 100{,}000$

Figura 31. Salida temporal del filtro KLMS para kernel Gaussiano con $\sigma = 0.707$, $M = 100$ y $N = 100{,}000$ muestras



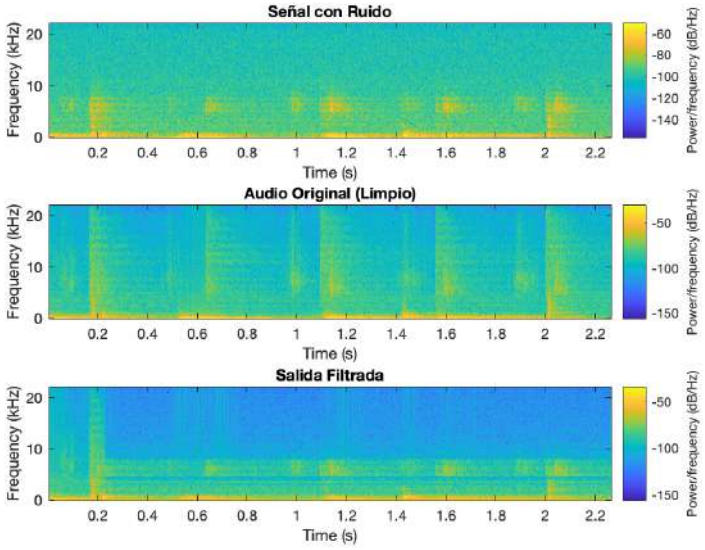
Nota. Elaboración propia.

Figura 32. Comparación entre referencia limpia, entrada reverberada y salida KLMS con kernel Gaussiano ($\sigma = 0.707$, $M = 100$, $N = 100{,}000$)



Nota. Elaboración propia.

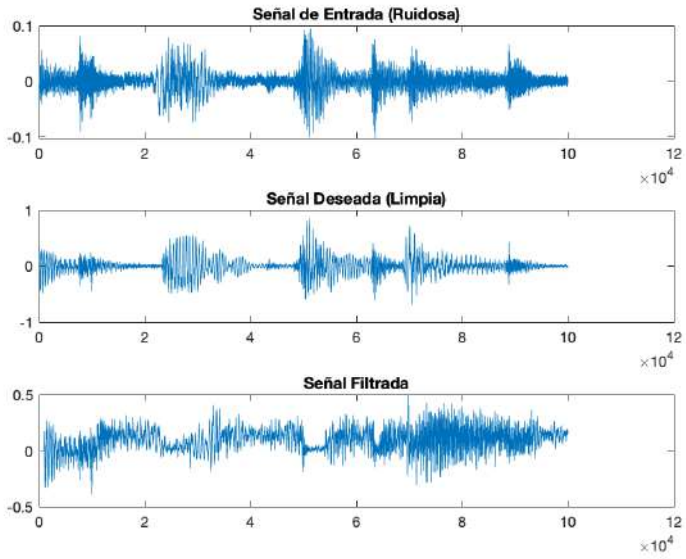
Figura 33. Espectrogramas de entrada, referencia y salida KLMS con kernel Gaussiano ($\sigma = 0.707$, $M = 100$ y $N = 100{,}000$)



Nota. Elaboración propia.

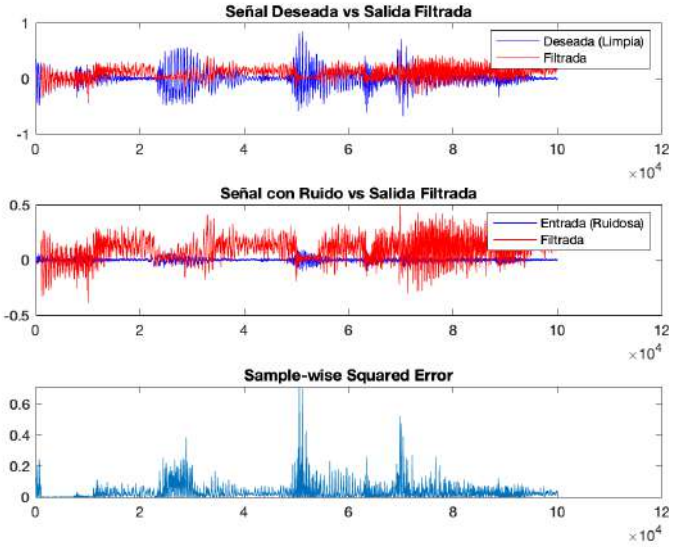
Kernel Gaussiano ($\sigma = 0.447$), $M = 1000$, $N = 100{,}000$

Figura 34. Salida temporal del filtro KLMS para kernel Gaussiano con $\sigma = 0.447$, $M = 1000$ y $N = 100{,}000$ muestras



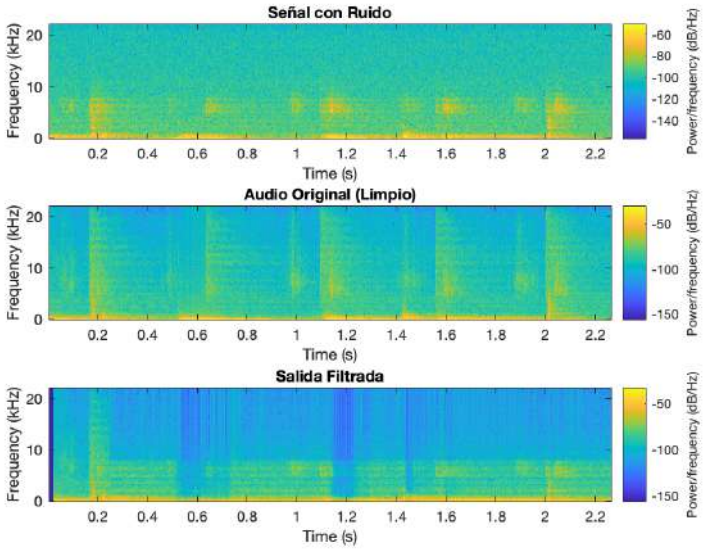
Nota. Elaboración propia.

Figura 35. Comparación entre referencia limpia, entrada reverberada y salida KLMS con kernel Gaussiano ($\sigma = 0.447$, $M = 1000$ y $N = 100{,}000$)



Nota. Elaboración propia.

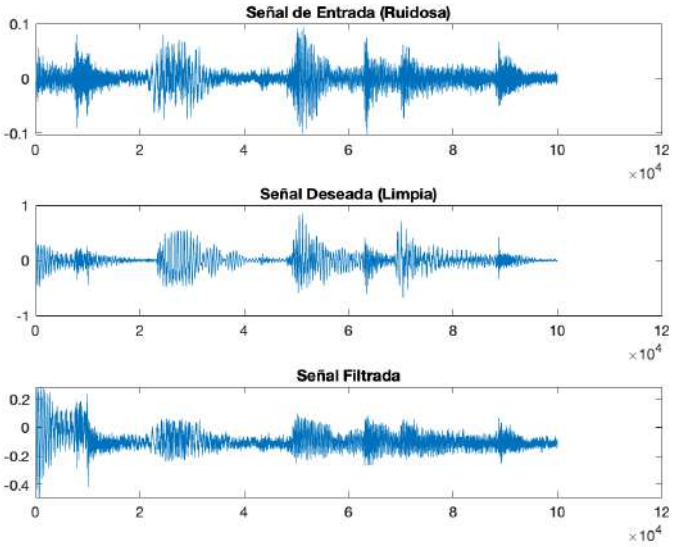
Figura 36. Espectrogramas de entrada, referencia y salida KLMS con kernel Gaussiano ($\sigma = 0.447$, $M = 1000$, $N = 100{,}000$)



Nota. Elaboración propia.

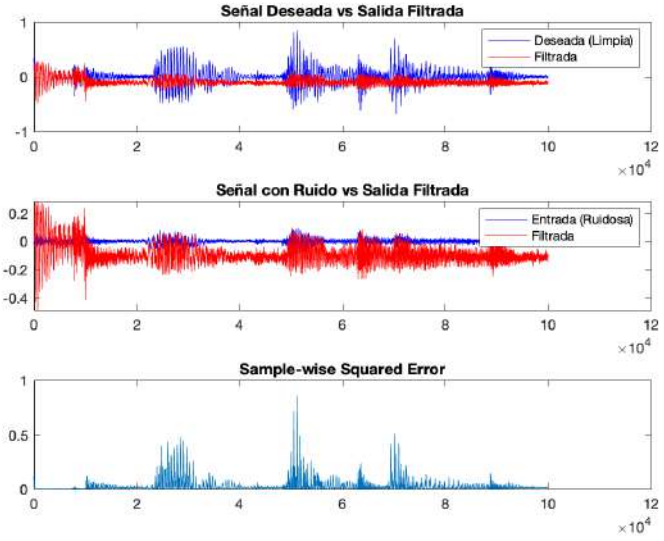
Kernel Laplaciano ($\sigma = 0.707$), $M = 100$, $N = 100{,}000$

Figura 37. Salida temporal del filtro KLMS para kernel Laplaciano con $\sigma = 0.707$, $M = 100$ y $N = 100{,}000$ muestras



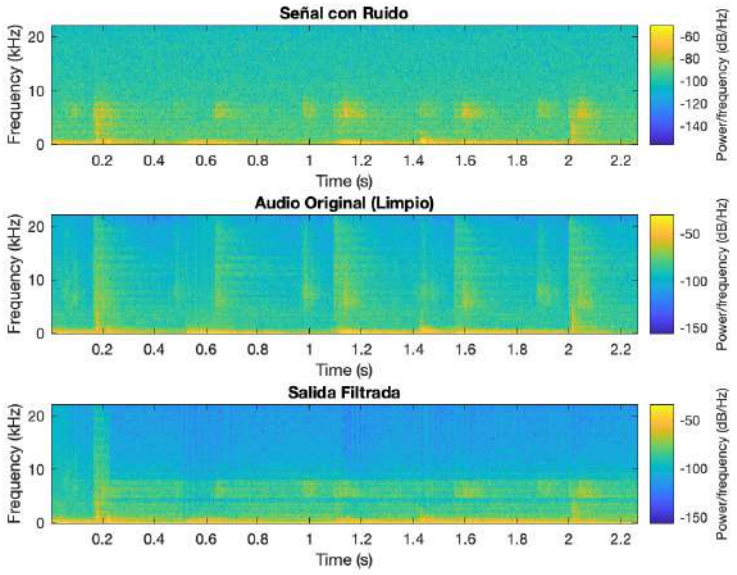
Nota. Elaboración propia.

Figura 38. Comparación entre referencia limpia, entrada reverberada y salida KLMS con kernel Laplaciano ($\sigma = 0.707$, $M = 100$ y $N = 100{,}000$)



Nota. Elaboración propia.

Figura 39. Espectrogramas de entrada, referencia y salida KLMS con kernel Laplaciano ($\sigma = 0.707$, $M = 100$, $N = 100{,}000$)



Nota. Elaboración propia.

9.1.3. Discusión

Los resultados obtenidos con el filtro KLMS permitieron identificar con claridad sus limitaciones para la tarea de deconvolución acústica con grabaciones reales. Aunque el método logró capturar ciertas relaciones no lineales y mejorar parcialmente la señal reverberada en ventanas temporales cortas, su comportamiento no resultó compatible con la naturaleza del problema abordado en esta investigación.

En primer lugar, KLMS es un algoritmo adaptativo en línea cuyos coeficientes continúan actualizándose a lo largo de toda la señal procesada. Esto implica que el modelo nunca converge hacia un conjunto estable de parámetros, sino que sigue modificándose en respuesta a los cambios dinámicos del contenido musical. Este comportamiento es coherente con la filosofía de los filtros adaptativos, pero entra en conflicto directo con la necesidad de estimar un modelo estático del recinto acústico, el cual debe representar una relación fija entre entrada y salida para ser reutilizada en nuevas grabaciones.

En segundo lugar, las configuraciones de mayor orden y longitud de procesamiento presentaron un costo computacional elevado. En particular, el tiempo de ejecución creció de manera significativa conforme aumentó el número de muestras procesadas, lo cual limita la viabilidad del KLMS para experimentos extensivos o para entornos donde se requiere explorar múltiples configuraciones de forma iterativa. Este comportamiento es resultado de la expansión creciente del diccionario de *kernels* y del cálculo repetitivo de productos internos en el espacio de características.

A pesar de que el *kernel* Gaussiano ofreció un desempeño superior al Laplaciano, las diferencias observadas en estabilidad y contenido espectral no modificaron el comportamiento global del método. En todos los casos persiste una degradación progresiva de la señal reconstruida, así como una falta de estacionariedad en el error, lo cual refuerza que las limitaciones identificadas son estructurales y no se corrigen mediante ajustes en el tipo de núcleo o en sus parámetros.

Por estas razones, el filtro KLMS se consideró únicamente como un punto de partida dentro de la fase exploratoria del proyecto. Sus resultados motivaron la transición hacia métodos de aprendizaje profundo, los cuales permiten aprender una representación fija del recinto sin requerir adaptación continua, abordando de forma más adecuada la naturaleza estática del problema de deconvolución acústica.

9.2. Modelo 2: *Temporal Convolutional Networks* (TCN)

9.2.1. Configuración del modelo

El segundo enfoque evaluado corresponde a una red neuronal convolucional temporal (*temporal convolutional network*, TCN) desarrollada en *PyTorch* y basada en los principios introducidos por la arquitectura WaveNet [14]. Esta TCN fue diseñada para operar directamente sobre señales de audio en el dominio temporal, aprovechando convoluciones causales y dilatadas que permiten capturar dependencias de largo alcance sin recurrir a estructuras recurrentes.

La red recibe una señal de entrada de un solo canal y aplica una proyección inicial mediante una convolución 1×1 que expande la dimensionalidad a 64 canales. Sobre esta representación se apilan seis bloques TCN, cada uno compuesto por dos convoluciones causales de anchura $k = 3$ y dilataciones crecientes de 2^i para $i = 0, \ldots, 5$. Cada convolución en el camino principal incorpora normalización de pesos, normalización por lotes, función de activación ReLU y *dropout*. Adicionalmente, cada bloque incluye rutas residual y de omisión (*skip connections*) que facilitan el flujo del gradiente durante el entrenamiento y permiten una integración eficiente de información a través de diferentes escalas temporales.

Cada bloque cuenta también con la opción de *stochastic depth*, de modo que, con una probabilidad fija, el bloque puede ignorar su transformación interna y transmitir directamente su entrada a las rutas residual y de omisión. Esta característica introduce una forma de regularización estructural que reduce la coadaptación entre capas profundas y mejora la estabilidad del entrenamiento cuando se trabaja con señales de larga duración.

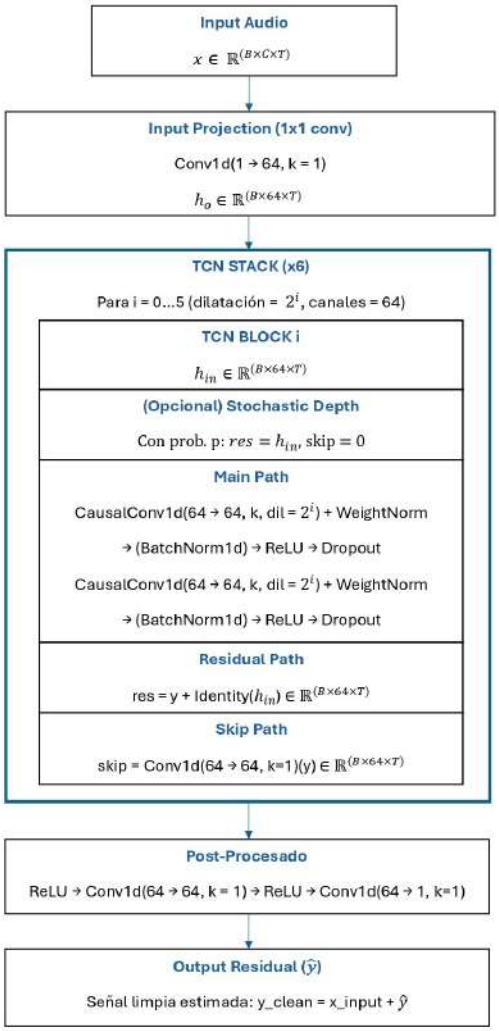
Finalmente, las contribuciones de las rutas de omisión se integran y procesan mediante un conjunto de convoluciones 1×1 con activaciones ReLU, lo que produce una estimación residual $\hat{r}(t)$ que representa la diferencia entre la señal reverberada y la señal limpia. La salida final del modelo se obtiene mediante aprendizaje residual, de modo que la señal limpia estimada se expresa como

$$\hat{y}(t) = x(t) + \hat{r}(t),$$

donde $x(t)$ es la señal de entrada reverberada. Este esquema permite que la red aprenda

únicamente las correcciones necesarias para eliminar la contribución del recinto acústico, lo cual mejora la estabilidad y facilita la generalización a nuevas grabaciones.

Figura 40. Diagrama general de la arquitectura TCN utilizada en esta investigación



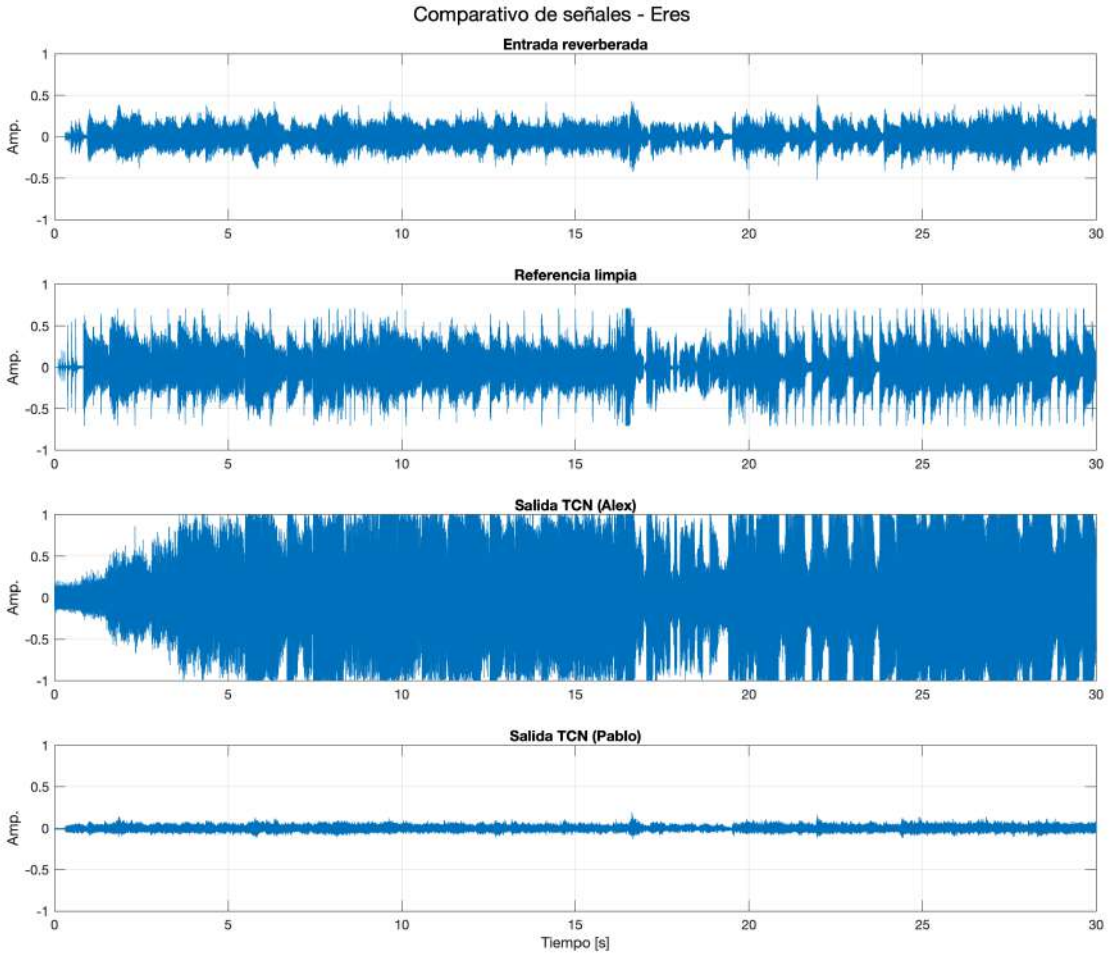
Nota. Elaboración propia.

9.2.2. Resultados

Esta sección presenta los resultados obtenidos por la red neuronal convolucional temporal (TCN) al procesar cuatro fragmentos musicales de distinta naturaleza espectral y dinámica. Para cada canción se incluyen dos figuras: un panel comparativo en el dominio del tiempo y un conjunto de espectrogramas que permite evaluar la reducción de reverberación, la conservación de componentes armónicas y la estabilidad temporal de la red. Finalmente, se resume el desempeño cuantitativo mediante las métricas RMSE, MAE y SNR.

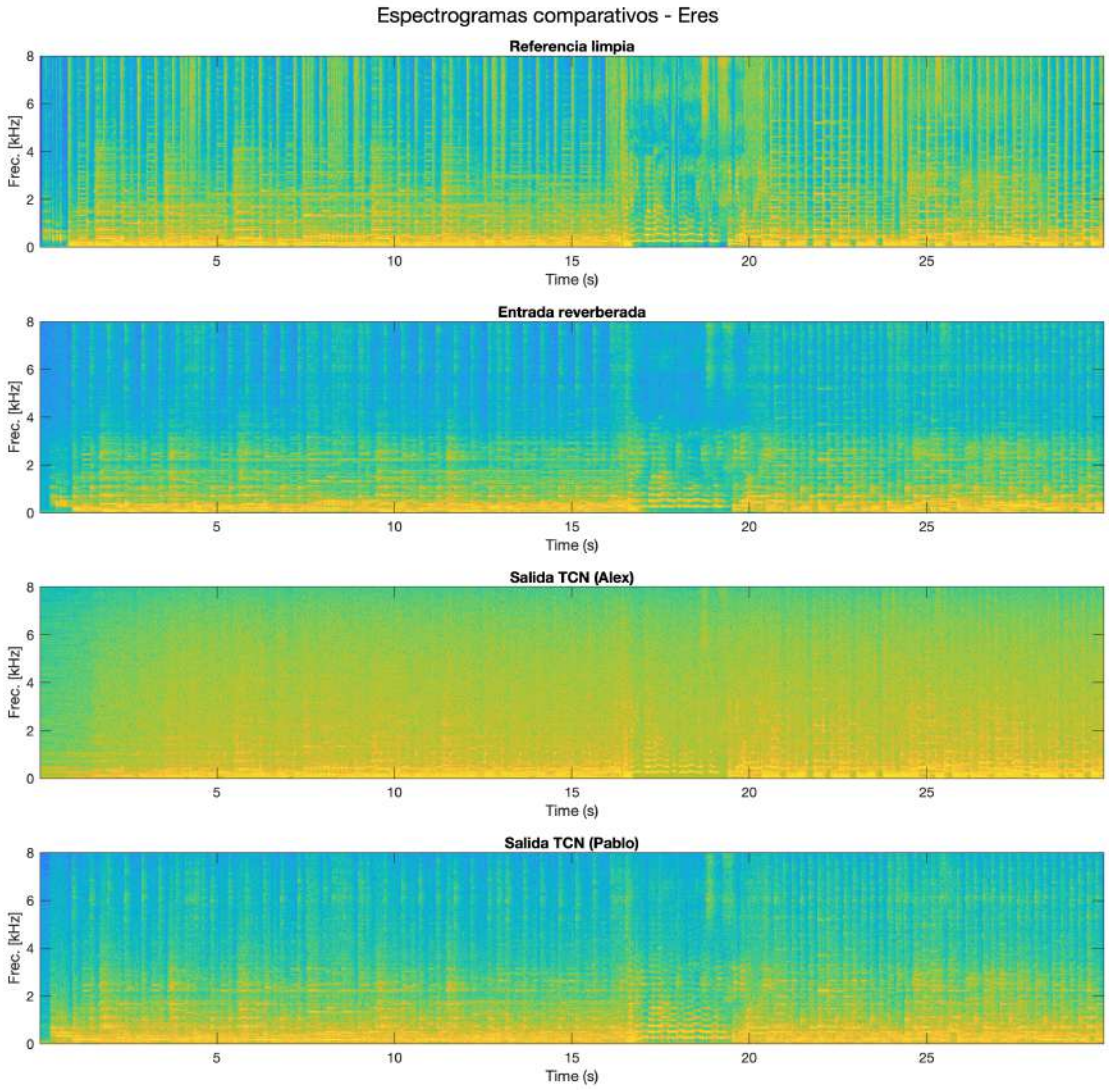
Canción 1: "Eres".

Figura 41. Comparativo temporal para la canción "Eres": entrada reverberada, señal limpia, salida TCN de Alex y salida TCN propuesta



Nota. Elaboración propia.

Figura 42. Espectrogramas comparativos para "Eres": referencia limpia, entrada reverberada, salida TCN de Alex y salida TCN propuesta

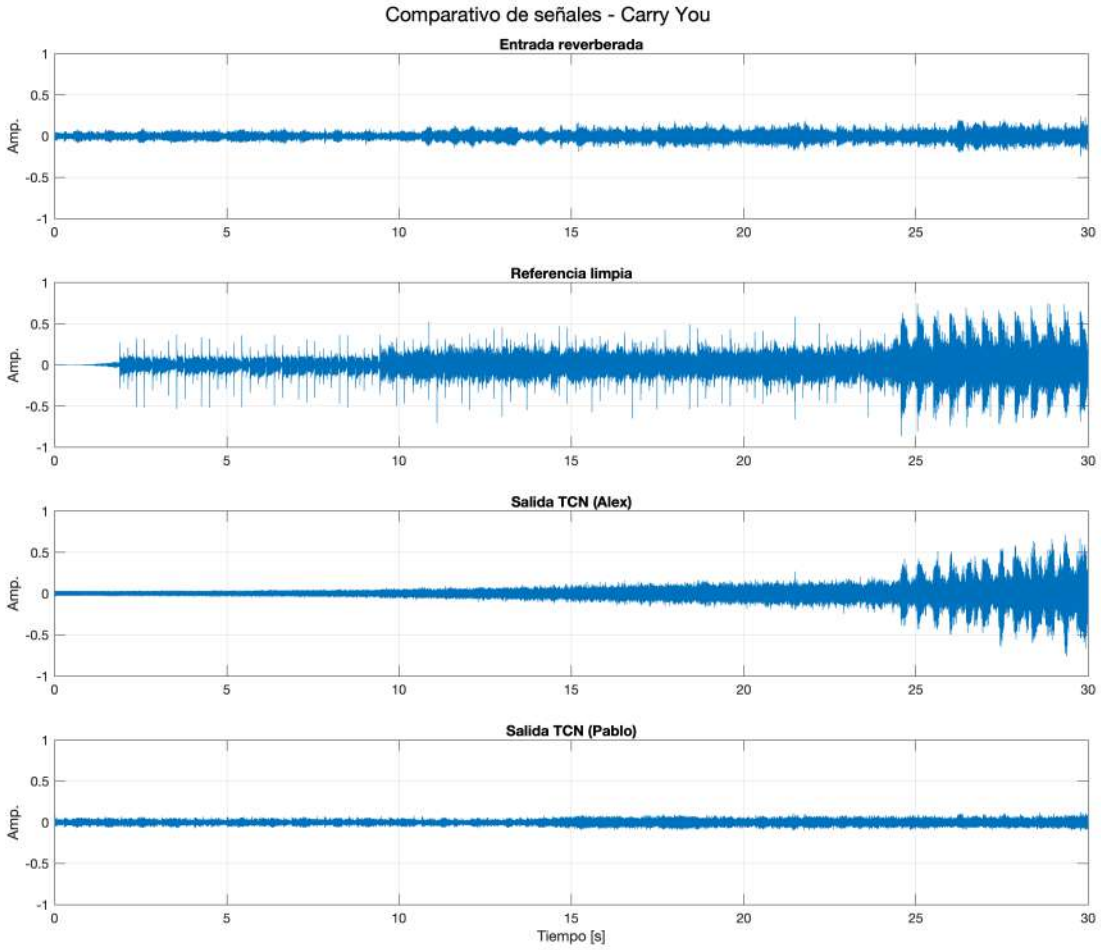


Nota. Elaboración propia.

Para esta canción, la TCN propuesta obtuvo un RMSE de 0.172, MAE de 0.133 y SNR de -0.08 dB, lo cual representa una mejora sustancial respecto al modelo previo, que presentó un RMSE de 0.318 y un SNR de -5.41 dB. Las gráficas evidencian una recuperación más estable de los transitorios y una menor energía residual de reverberación en bandas medias y altas.

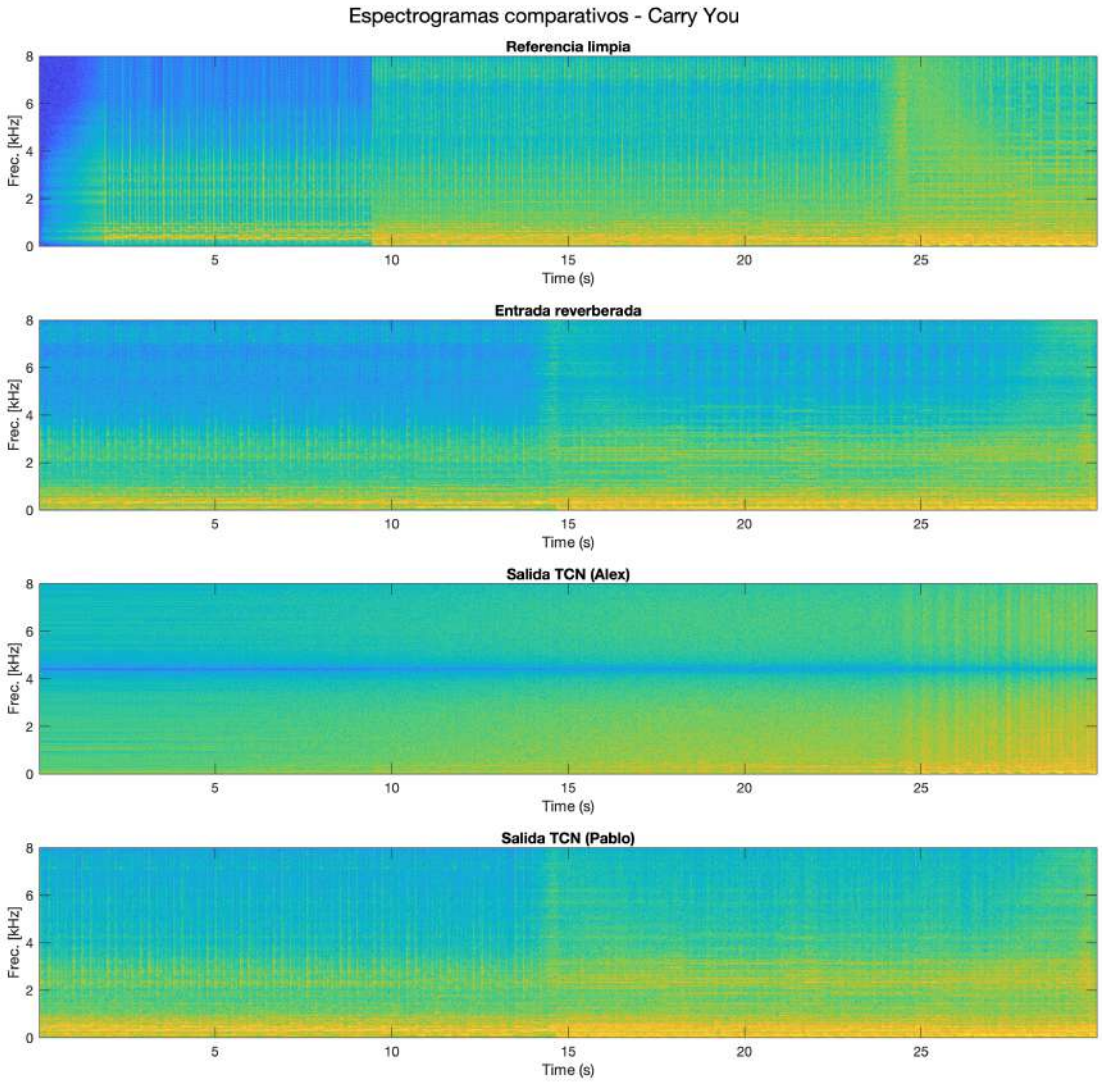
Canción 2: "Carry You".

Figura 43. Comparativo temporal para "Carry You"



Nota. Elaboración propia.

Figura 44. Espectrogramas comparativos para “Carry You”

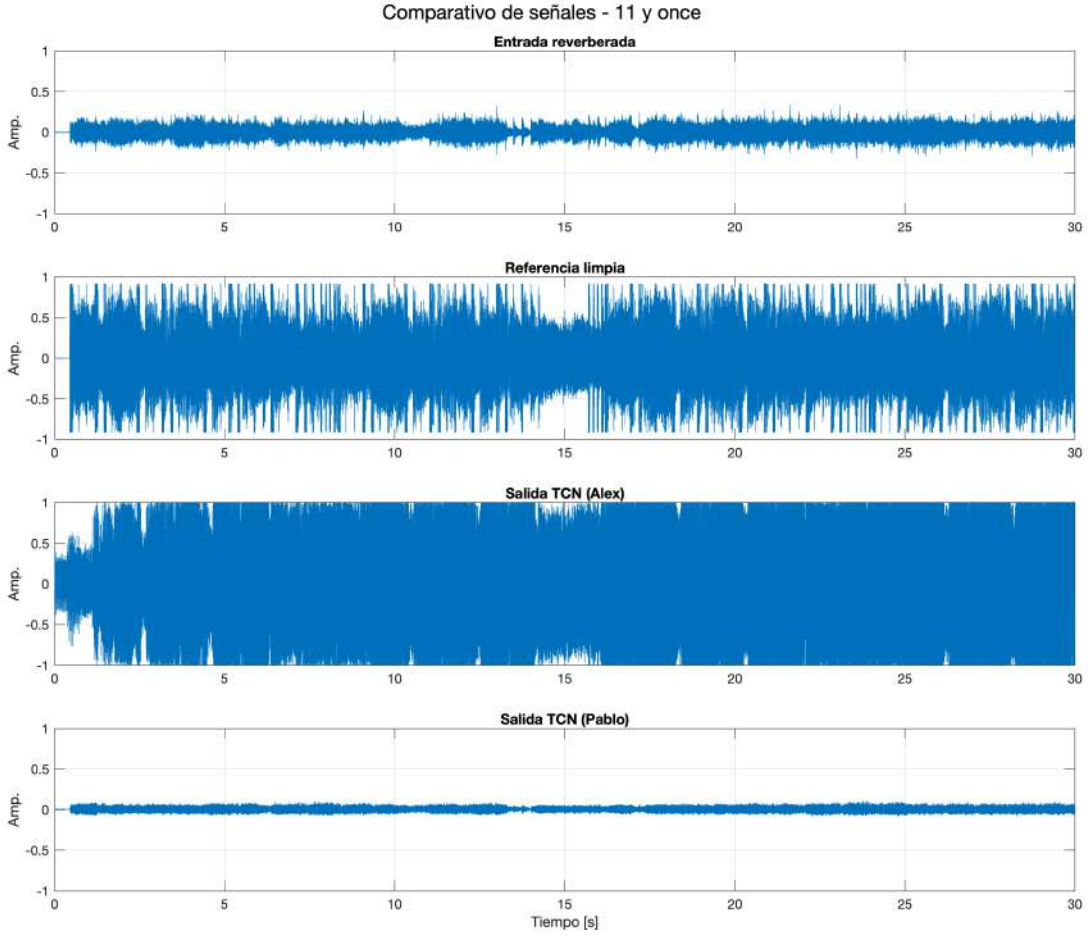


Nota. Elaboración propia.

En esta canción, caracterizada por menor densidad espectral y presencia de eventos suaves, el modelo previo obtuvo mejores resultados cuantitativos, con un RMSE de 0.070 y un SNR de 3.79 dB frente al RMSE de 0.112 y SNR de -0.25 dB alcanzados por la TCN propuesta. Este comportamiento sugiere que el modelo previo conserva mejor la energía de pasajes con baja ocupación armónica.

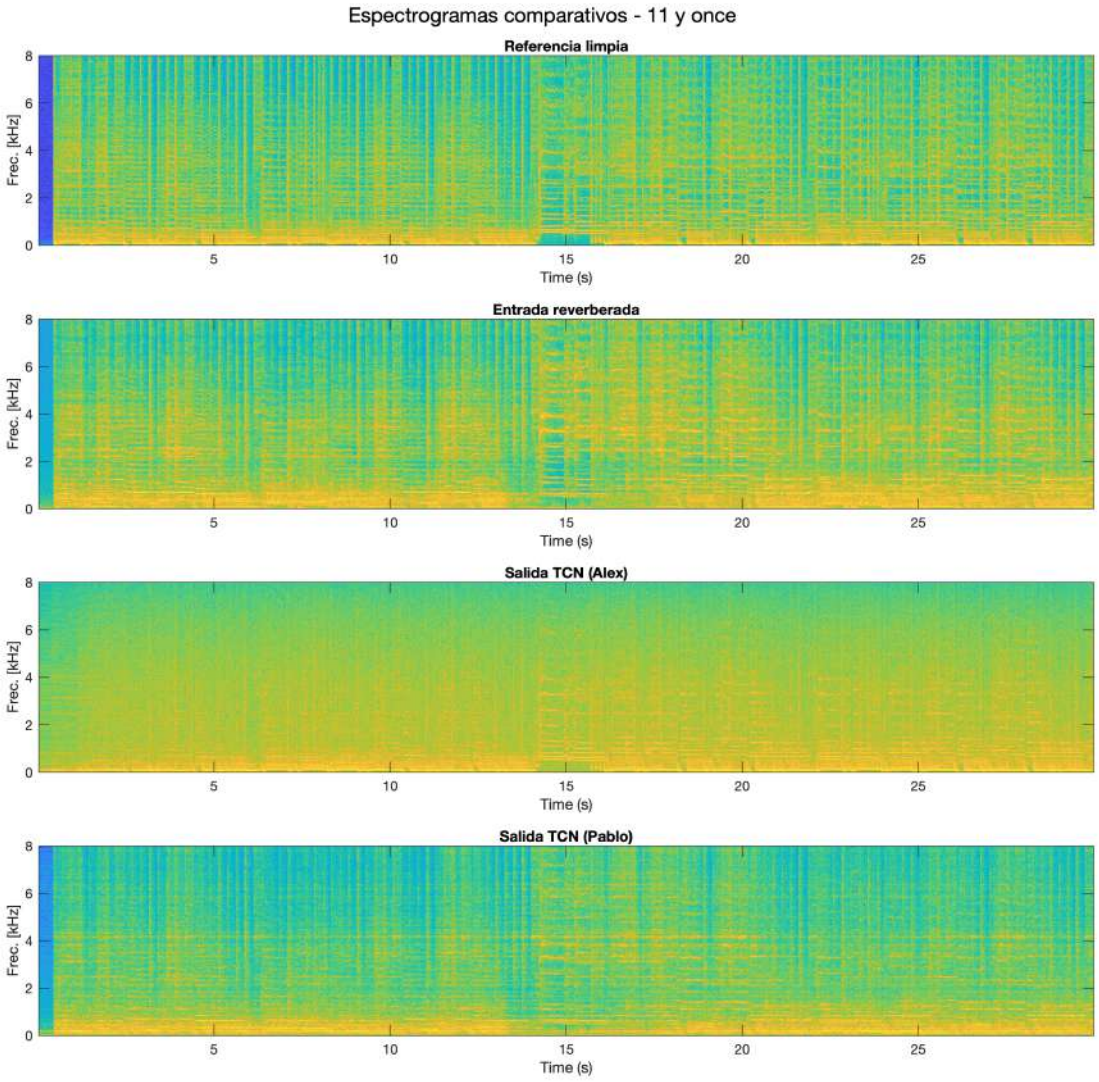
Canción 3: "11 y once".

Figura 45. Comparativo temporal para "11 y once"



Nota. Elaboración propia.

Figura 46. Espectrogramas comparativos para “11 y once”

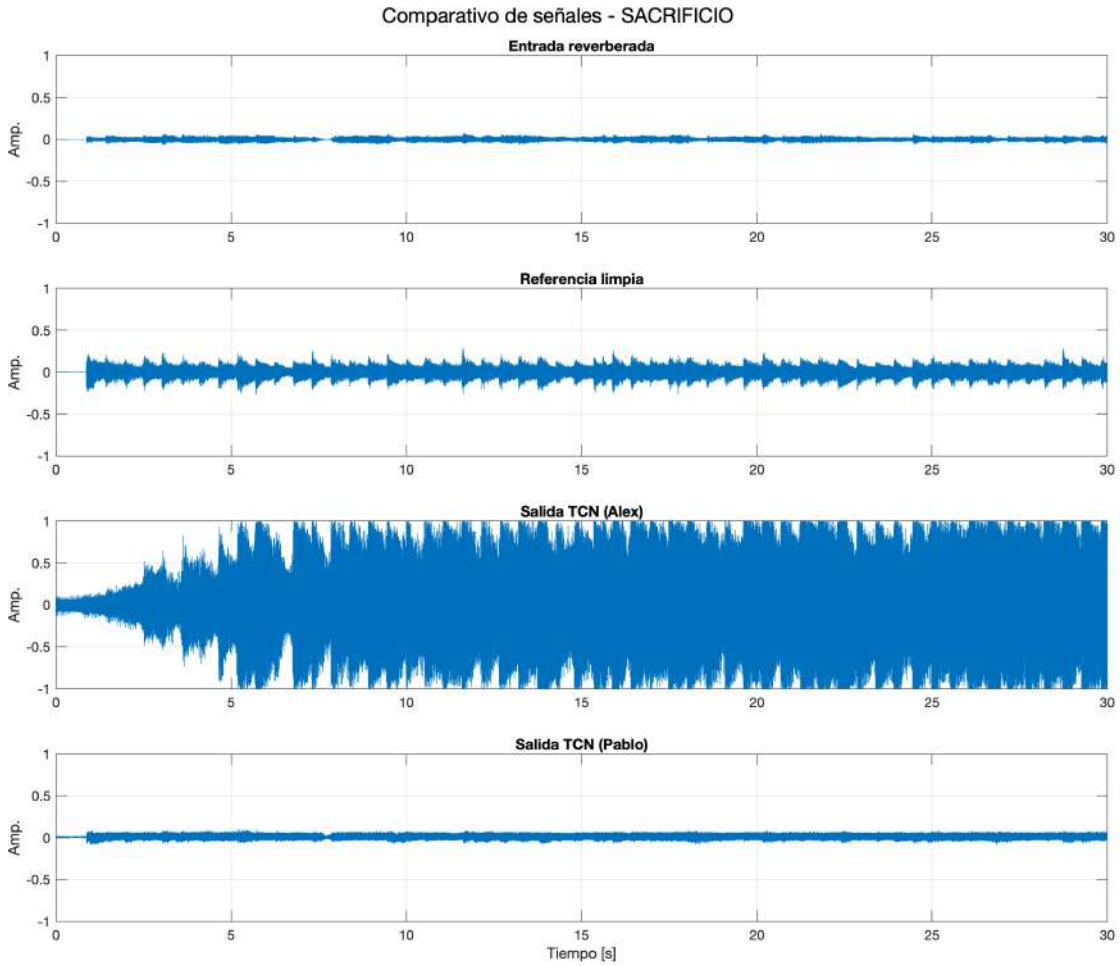


Nota. Elaboración propia.

La TCN propuesta mostró un mejor desempeño, con un RMSE de 0.322 y un SNR cercano a 0 dB, frente al RMSE de 0.360 y SNR de -0.97 dB del modelo previo. Se observa una menor persistencia de energía residual en las colas de reverberación y una reconstrucción más fiel de los transitorios vocales.

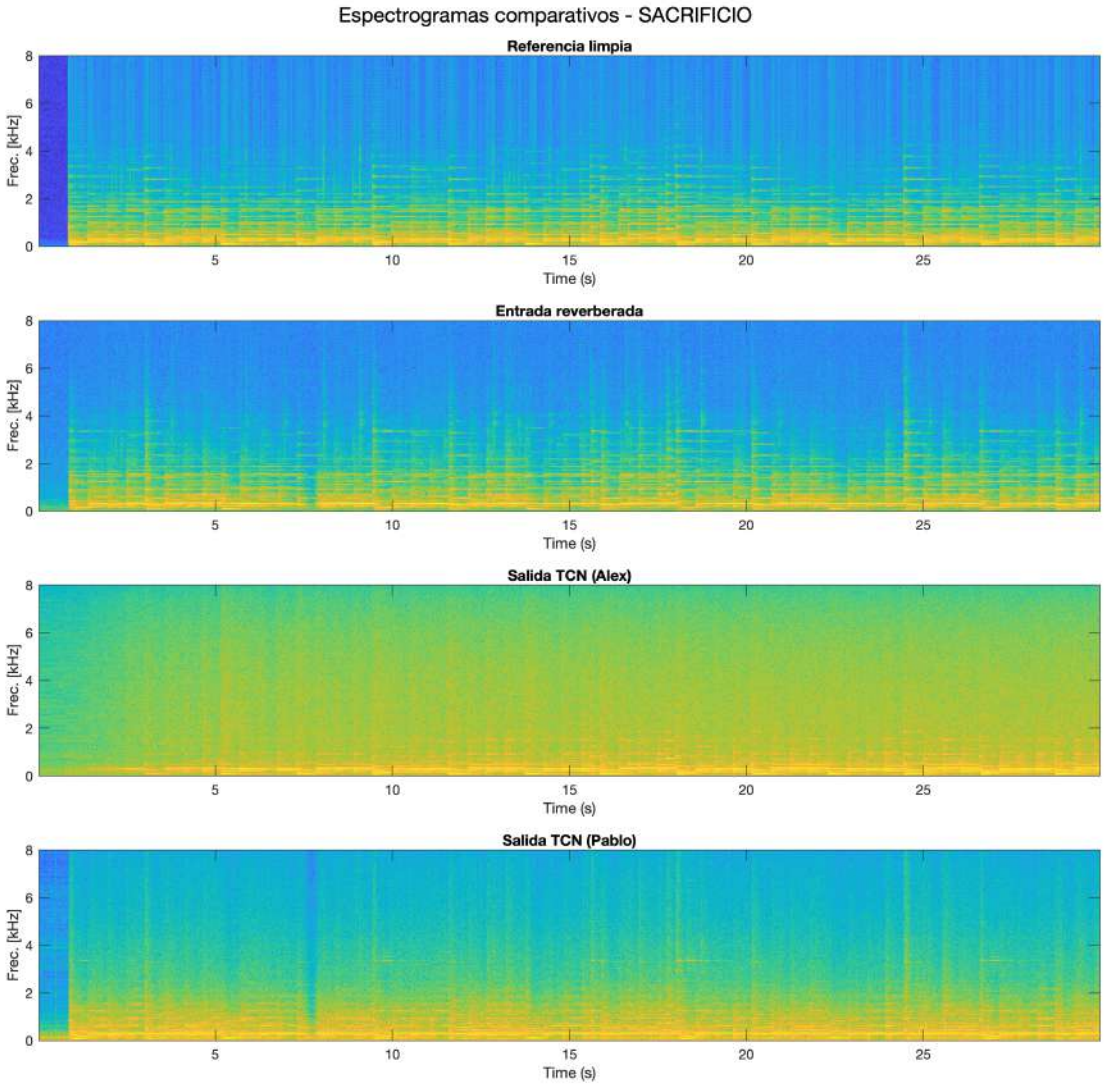
Canción 4: "SACRIFICIO".

Figura 47. Comparativo temporal para "SACRIFICIO"



Nota. Elaboración propia.

Figura 48. Espectrogramas comparativos para "SACRIFICIO"



Nota. Elaboración propia.

En esta canción, que presenta alta densidad espectral y mezcla compleja, la TCN propuesta superó de manera marcada al modelo previo. Con un RMSE de 0.055 y un MAE de 0.042, frente al RMSE de 0.369 y SNR de -16.63 dB del modelo anterior, los resultados indican una mejor conservación de la estructura harmónica y una reducción de reverberación significativamente superior.

Resumen cuantitativo

El Cuadro 4 presenta un resumen de las métricas RMSE, MAE y SNR obtenidas para las cuatro canciones consideradas en esta evaluación, permitiendo observar de manera

condensada el desempeño del modelo en cada caso.

Cuadro 4. Resumen de métricas RMSE, MAE y SNR para las cuatro canciones analizadas

Canción	Modelo	RMSE	MAE	SNR (dB)
Eres	TCN propuesta	0.172	0.133	−0.08
	TCN Calí	0.318	0.254	−5.41
Carry You	TCN propuesta	0.113	0.078	−0.25
	TCN Calí	0.071	0.053	3.79
11 y once	TCN propuesta	0.322	0.249	0.02
	TCN Calí	0.361	0.308	−0.97
Sacrificio	TCN propuesta	0.055	0.042	−0.16
	TCN Calí	0.369	0.291	−16.63

Nota. Elaboración propia.

9.2.3. Discusión

Los resultados muestran que la arquitectura TCN desarrollada en esta investigación ofrece un desempeño estable en la mayoría de los fragmentos musicales evaluados. En tres de las cuatro canciones analizadas ("Eres", "11 y once" y "Sacrificio") el modelo propuesto alcanzó errores más bajos y valores de SNR más favorables que la versión previa (Calí). Esto se aprecia tanto en las métricas como en las figuras, donde la salida de la TCN propuesta conserva mejor la energía útil de la señal y reduce la presencia de reverberación en comparación con el modelo anterior.

Un aspecto que resalta es la capacidad del modelo para manejar canciones con mayor complejidad sonora, como "11 y once", donde la mezcla presenta una superposición notable de instrumentos y componentes armónicas. En este tipo de material, el modelo previo pierde estabilidad y deja pasar una cantidad considerable de reverberación, mientras que la TCN propuesta mantiene una salida más clara y definida. Esto sugiere que la arquitectura diseñada responde bien cuando la señal tiene mucha información distribuida en distintas frecuencias.

Por otro lado, en la canción "Carry You", que tiene menor densidad espectral y una estructura más limpia, el modelo previo presentó mejores resultados. En este caso, la TCN propuesta tiende a suavizar demasiado la señal, lo que afecta la recuperación precisa de algunos detalles. Este comportamiento indica que, en señales menos complejas, un modelo más conservador puede preservar mejor las características originales del audio.

En conjunto, los resultados sugieren que la TCN desarrollada generaliza de forma más consistente ante señales densas y variadas, que son las que suelen presentar mayores desafíos en tareas de deconvolución acústica. Aunque el modelo previo mostró ventajas en escenarios simples, la propuesta de esta investigación demuestra un mejor equilibrio entre reducción de reverberación y conservación de la estructura general de la señal, especialmente en condiciones más exigentes.

9.3. Modelo 3: modelo AutoRegresivo No Lineal con Entrada Exógena (NLARX)

9.3.1. Configuración del modelo

El tercer enfoque evaluado corresponde a un modelo autorregresivo no lineal con entrada exógena (*nonlinear autoregressive model with exogenous input*, NLARX) utilizando como regresor una red neuronal de tipo *wavelet*. La configuración del modelo siguió la estructura base utilizada previamente en la etapa de identificación de sistemas. En particular, se emplearon seis retardos de entrada y seis retardos de salida, lo cual proporciona una ventana temporal suficiente para capturar la relación entre la señal afectada por el recinto acústico y la señal original. Esta elección buscó mantener un balance entre capacidad de representación y estabilidad numérica durante el entrenamiento.

Se intentó explorar configuraciones de mayor orden para aumentar la capacidad del modelo y permitirle representar estructuras temporales más largas. Sin embargo, al superar diez regresores tanto en la entrada como en la salida, la aplicación *System Identification* de *MATLAB* dejó de responder o presentó tiempos de cómputo excesivamente prolongados. Esta limitación impidió evaluar modelos NLARX de mayor complejidad, por lo que el análisis se centró en la configuración descrita anteriormente.

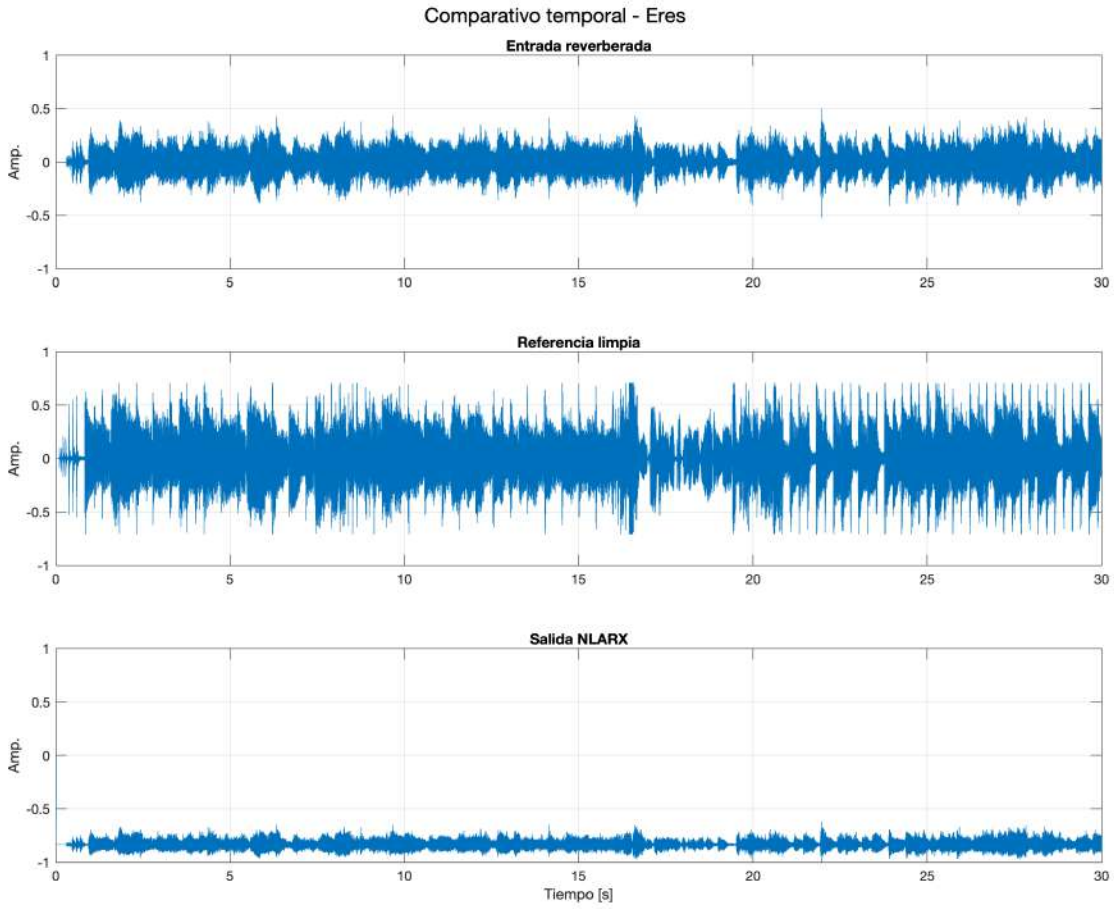
Esta arquitectura permitió examinar el comportamiento del NLARX en un contexto de deconvolución acústica, evaluando su capacidad para recuperar la estructura general de la señal limpia a partir de sus versiones reverberadas.

9.3.2. Resultados

Esta sección presenta el desempeño del modelo NLARX basado en funciones *wavelet* aplicado a las cuatro canciones empleadas en las evaluaciones previas. Para cada fragmento musical se incluyen dos figuras: un comparativo temporal entre la señal reverberada, la referencia limpia y la salida del NLARX, así como un panel de espectrogramas que permite observar la distribución de energía en tiempo y frecuencia. Al final, se resumen las métricas de error en una tabla comparativa.

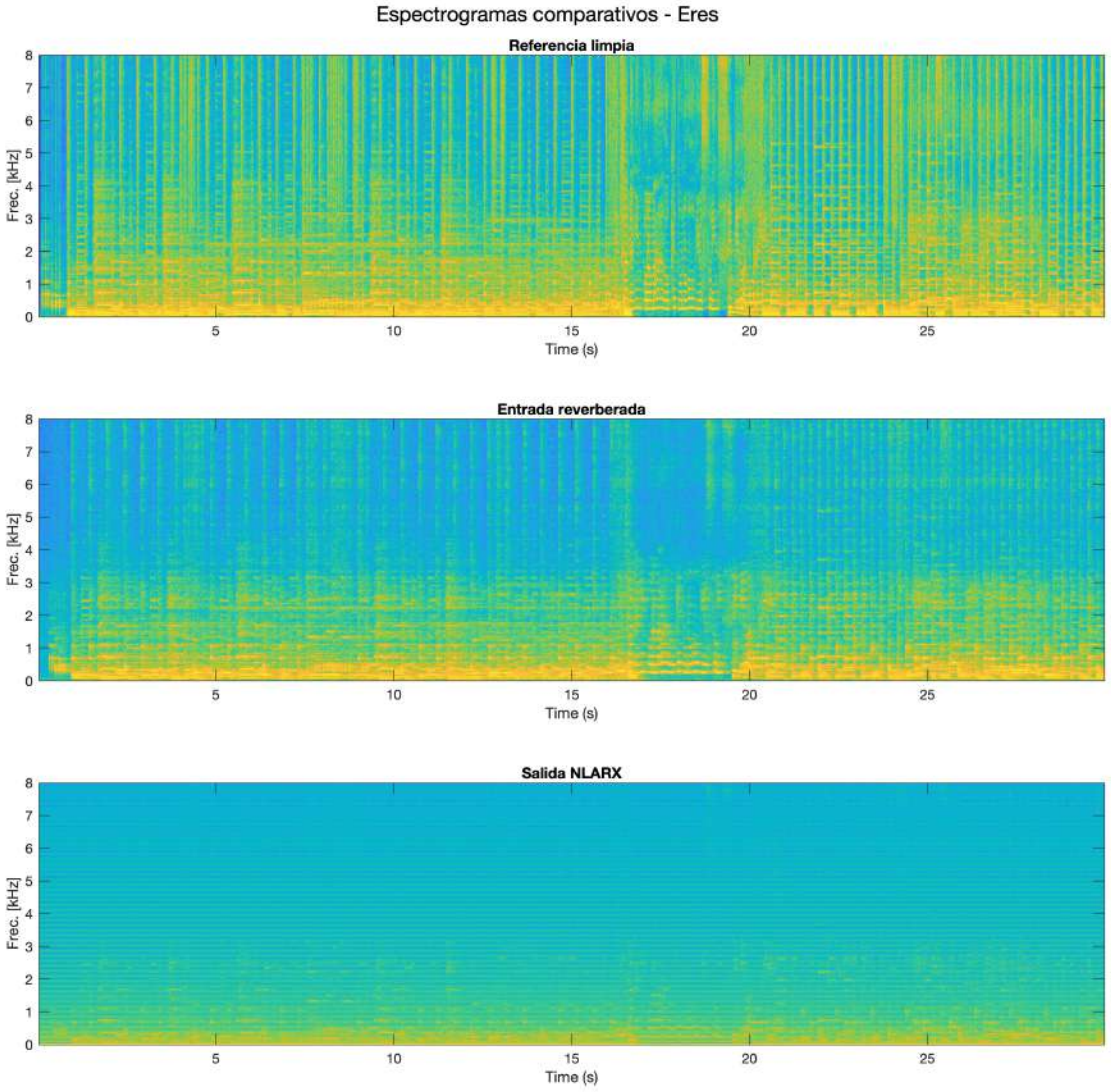
Canción 1: "Eres".

Figura 49. Comparativo temporal para "Eres": entrada reverberada, señal limpia y salida NLARX



Nota. Elaboración propia.

Figura 50. Espectrogramas comparativos para "Eres": referencia limpia, entrada reverberada y salida NLARX

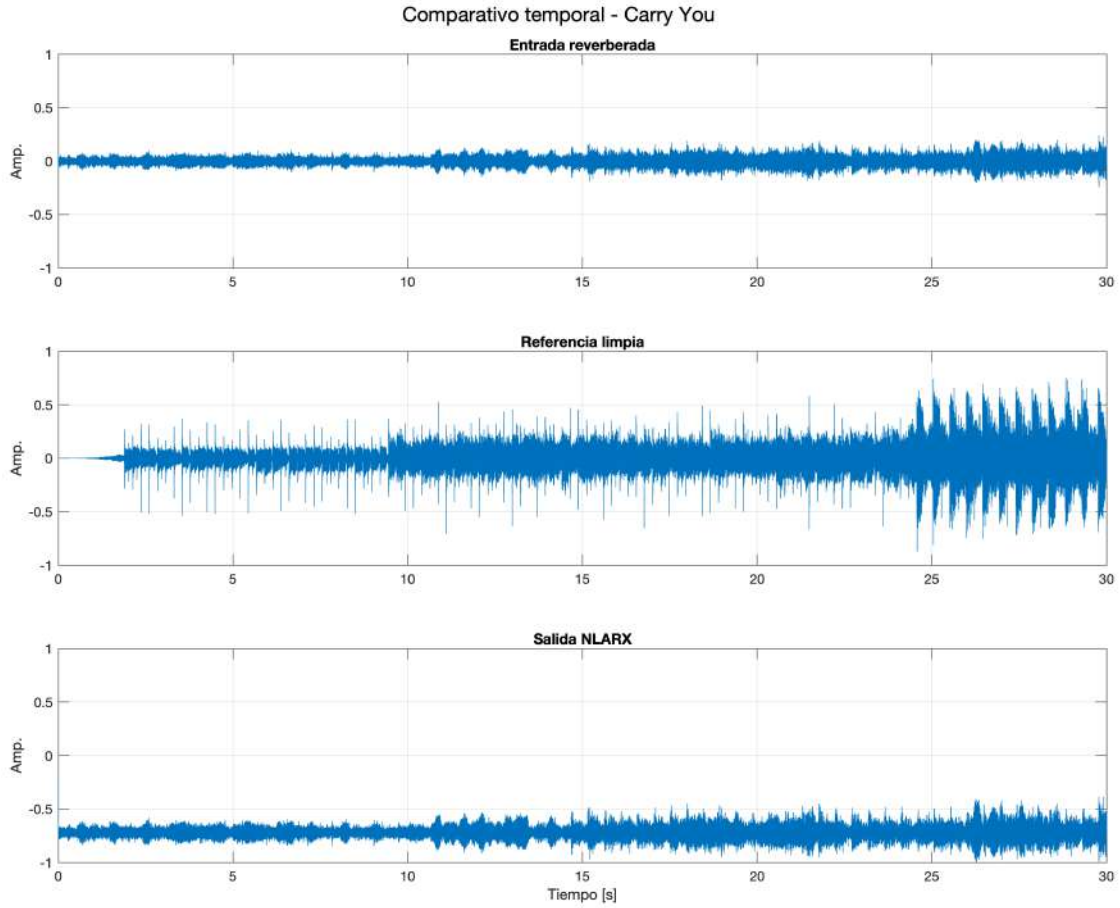


Nota. Elaboración propia.

En esta canción, el modelo NLARX mostró un rendimiento limitado, con un RMSE de 0.851 y un SNR de -13.95 dB. Las gráficas reflejan una atenuación significativa de la señal, acompañada de un desplazamiento hacia valores negativos, lo cual afecta la reconstrucción de las componentes armónicas y dificulta la recuperación de la señal limpia.

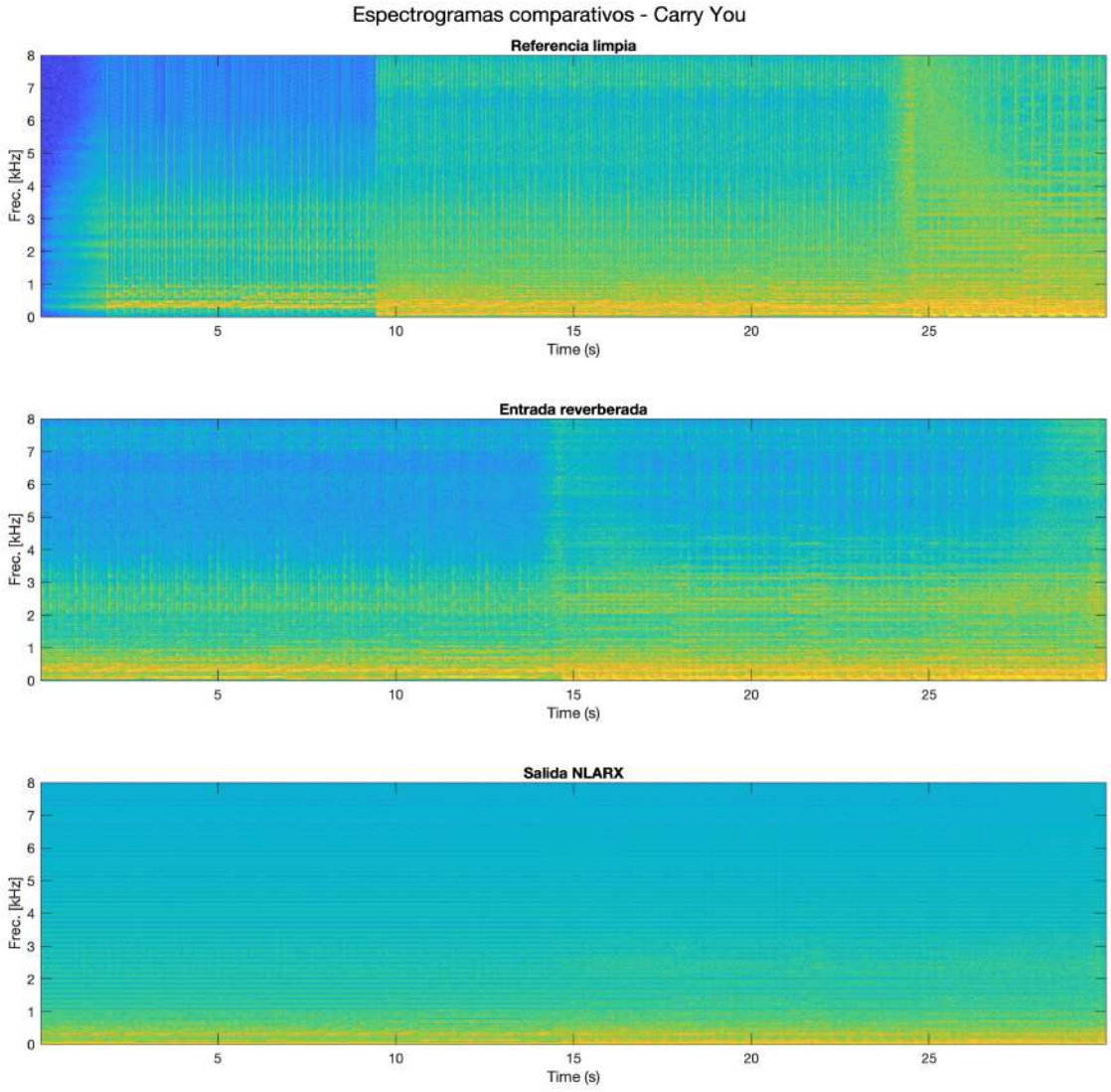
Canción 2: "Carry You".

Figura 51. Comparativo temporal para "Carry You"



Nota. Elaboración propia.

Figura 52. Espectrogramas comparativos para “Carry You”

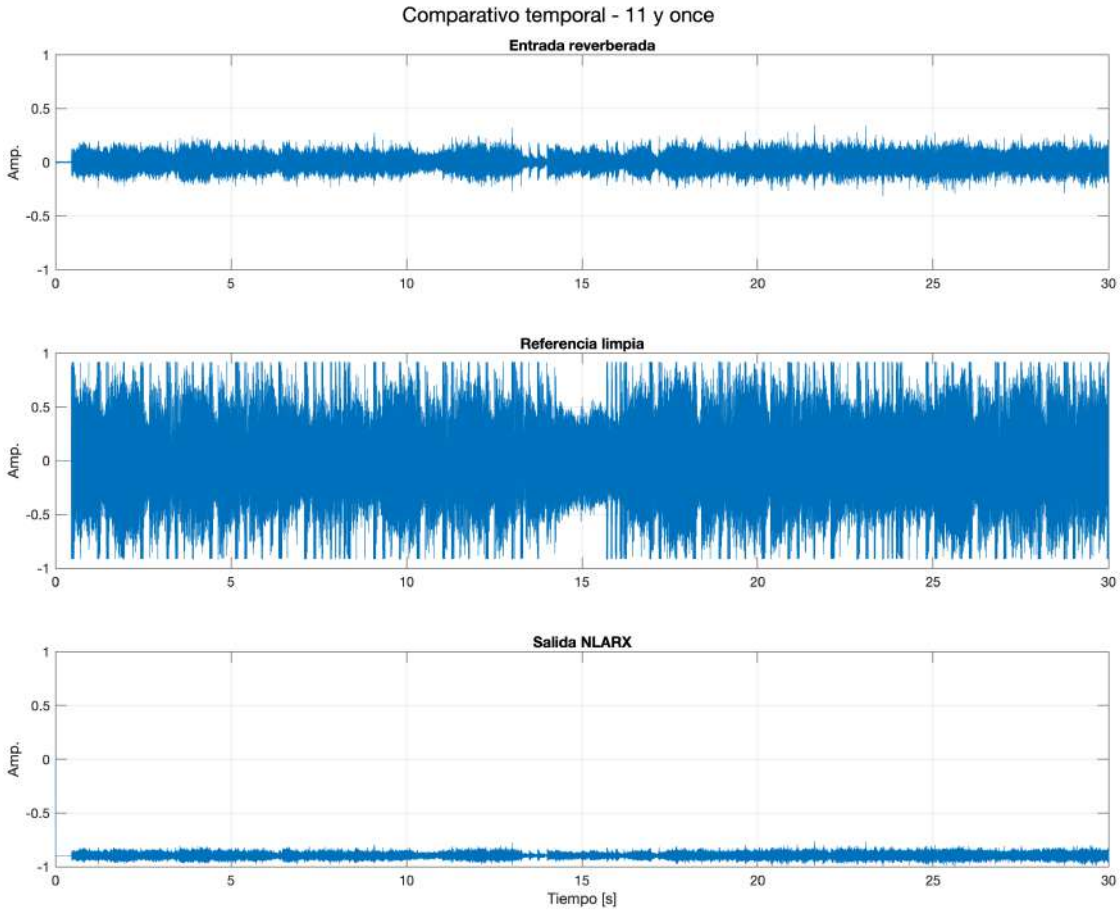


Nota. Elaboración propia.

El desempeño del NLARX fue similar al caso anterior, con valores de RMSE de 0.731 y un SNR de -16.48 dB. Se observa una salida suavizada en exceso y con pérdida notable de energía en bandas medias y altas, lo cual indica que el modelo actúa de manera semejante a un filtro pasabajas.

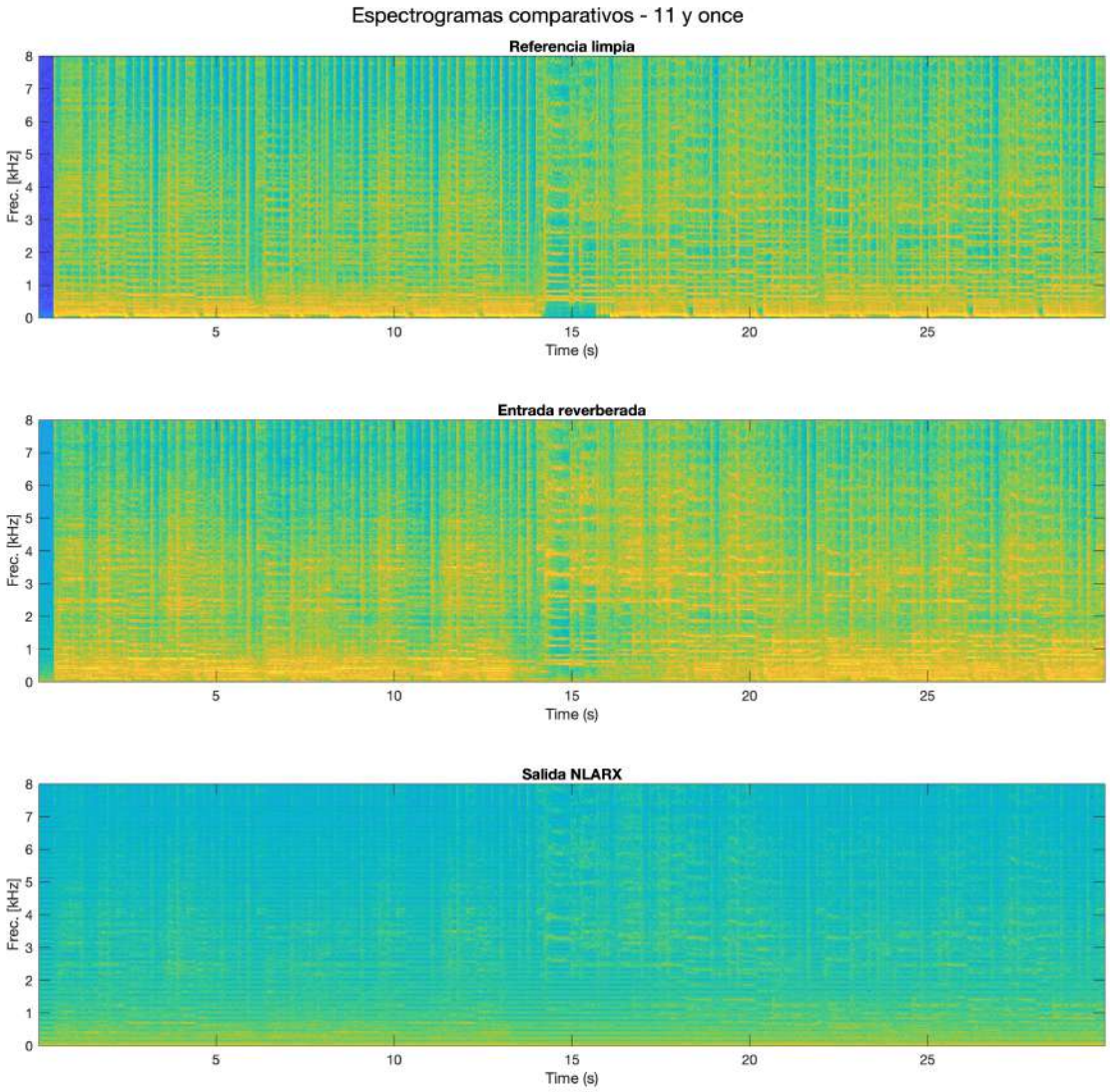
Canción 3: "11 y once".

Figura 53. Comparativo temporal para "11 y once"



Nota. Elaboración propia.

Figura 54. Espectrogramas comparativos para “11 y once”

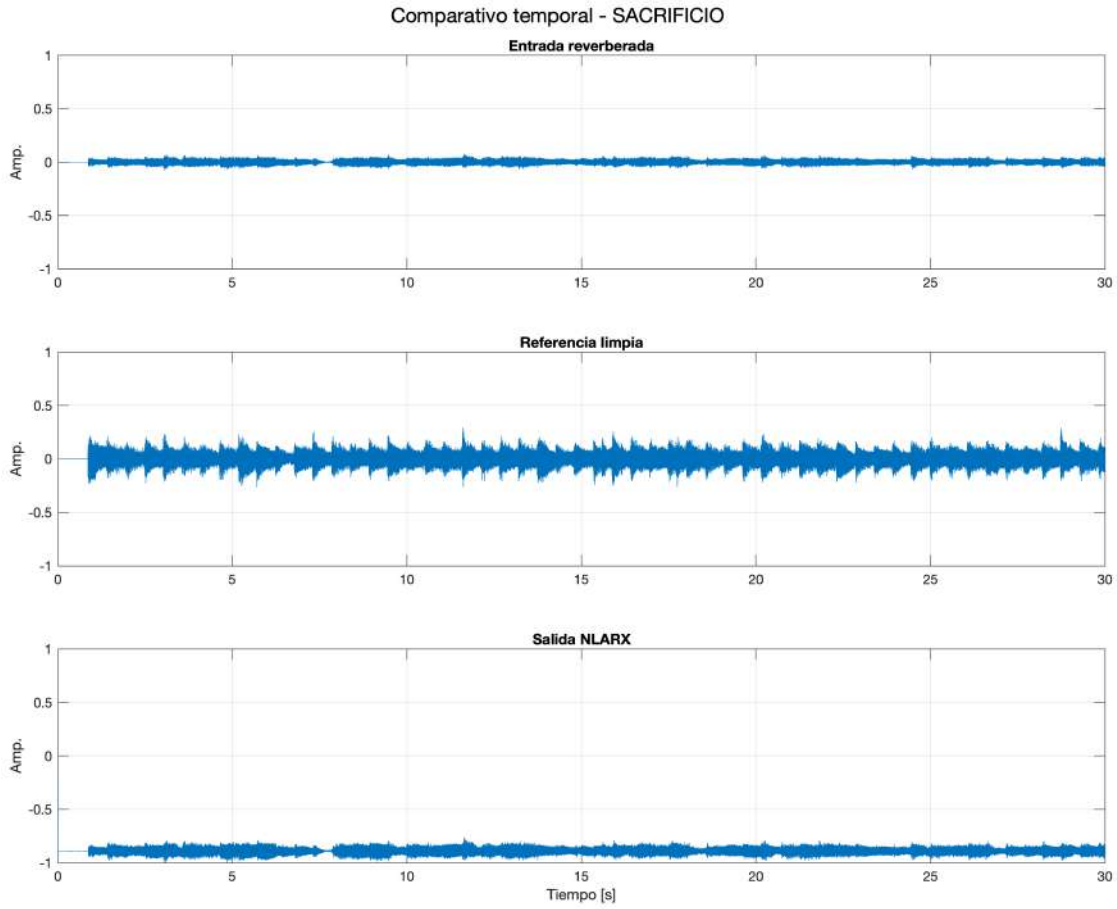


Nota. Elaboración propia.

En esta canción, la salida del NLARX muestra nuevamente una atenuación pronunciada, con un RMSE de 0.954 y un SNR de -9.42 dB. El espectrograma evidencia pérdida de contenido armónico y reducción de la energía directa, lo cual afecta la estructura general de la señal reconstruida.

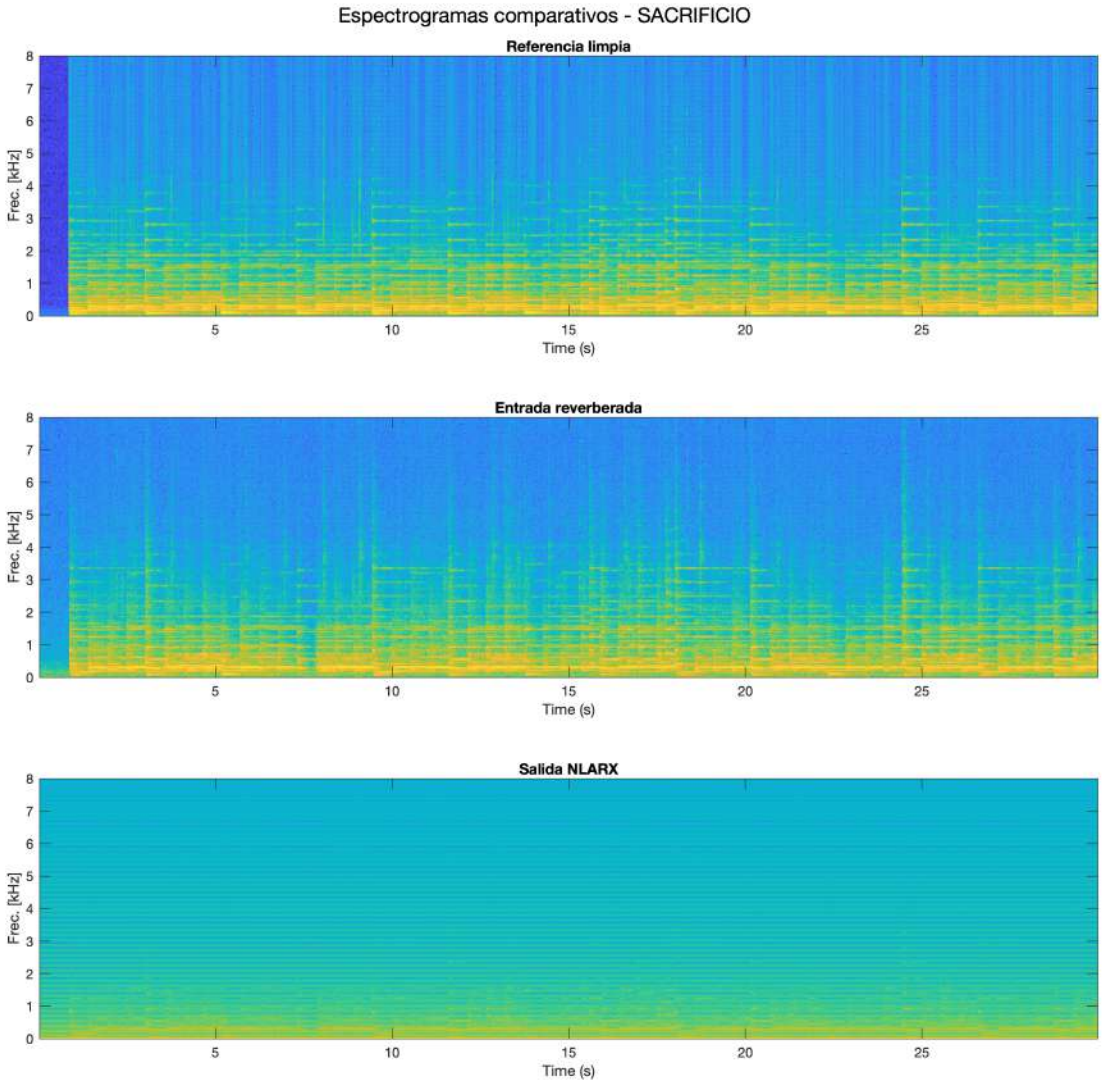
Canción 4: "Sacrificio".

Figura 55. Comparativo temporal para "SACRIFICIO"



Nota. Elaboración propia.

Figura 56. Espectrogramas comparativos para "SACRIFICIO"



Nota. Elaboración propia.

El comportamiento observado en "Sacrificio" fue consistente con los casos anteriores. El NLARX obtuvo un RMSE de 0.893 y un SNR de -24.30 dB, evidenciando una salida dominada por atenuación excesiva y desplazamiento vertical hacia valores negativos, lo cual impide la recuperación adecuada de la señal original.

Resumen cuantitativo

Tal como se muestra en el Cuadro 5, el modelo NLARX obtuvo errores elevados (RMSE y MAE) y valores de SNR negativos para las cuatro canciones evaluadas. Este comportamiento evidencia la pérdida sistemática de contenido de alta frecuencia y su limitada capacidad de

reconstrucción en escenarios acústicos reales.

Cuadro 5. Resumen de métricas RMSE, MAE y SNR para NLARX en las cuatro canciones

Canción	RMSE	MAE	SNR (dB)
Eres	0.851	0.833	−13.95
Carry You	0.731	0.720	−16.48
11 y once	0.954	0.898	−9.42
Sacrificio	0.893	0.891	−24.30

Nota. Elaboración propia.

9.3.3. Discusión

Los resultados obtenidos con el modelo NLARX muestran un patrón consistente en las cuatro canciones evaluadas. En todos los casos, la salida del modelo presentó una atenuación considerable de la señal, acompañada de un desplazamiento progresivo hacia valores negativos. Este comportamiento indica que el NLARX no logró modelar adecuadamente la relación entre la señal reverberada y la señal limpia, y en su lugar respondió como un filtro pasabajas que reduce de forma indiscriminada la energía del contenido de alta frecuencia.

El análisis temporal revela que la forma de onda resultante conserva únicamente la estructura más gruesa de la señal original, pero pierde transitorios, armónicos y detalles dinámicos esenciales para la restauración de la grabación. Esta pérdida también se observa en los espectrogramas, donde la energía por encima de las bandas bajas disminuye de manera pronunciada, lo que evidencia que el modelo no compensó la reverberación ni recuperó la respuesta original antes del efecto del recinto acústico.

Desde el punto de vista computacional, el NLARX también presentó limitaciones. Se intentó incrementar el número de regresores con el fin de otorgar mayor capacidad temporal al modelo, sin embargo, al superar diez retardos en la entrada y la salida, la aplicación *System Identification* mostró inestabilidad y tiempos de respuesta excesivamente altos, lo que impidió evaluar configuraciones más complejas. Como consecuencia, el modelo quedó restringido a configuraciones de menor orden que no fueron suficientes para la tarea de deconvolución acústica.

En conjunto, los resultados sugieren que, aunque el NLARX puede capturar relaciones no lineales en otros contextos, la combinación de su regresor basado en *wavelets* y el número limitado de retardos no permitió modelar adecuadamente la interacción temporal que caracteriza a la reverberación. Su salida suavizada y su tendencia a desplazar la señal hacia valores negativos limitan severamente su utilidad práctica en este tipo de aplicaciones, especialmente cuando se compara con los métodos evaluados en las secciones anteriores.

9.4. Discusión comparativa entre modelos

La evaluación conjunta de los tres métodos permite identificar diferencias claras en su comportamiento y en la utilidad real que cada uno ofrece para el problema de deconvolución acústica. Aunque todos fueron aplicados a las mismas grabaciones, sus resultados muestran que cada modelo opera bajo principios distintos y responde de manera diferente a las características de las señales musicales.

El filtro KLMS evidenció desde el inicio limitaciones estructurales importantes. Su naturaleza dinámica provocó que los pesos cambiaran continuamente durante procesamientos largos, lo que afectó la estabilidad del método y degradó la señal recuperada. Esto, sumado a su costo computacional creciente con la longitud del audio, lo vuelve poco adecuado para aplicaciones prácticas donde se busca un modelo estático que pueda generalizar a distintas grabaciones sin recalibrarse cada vez.

Por otro lado, el modelo NLARX mostró un desempeño insuficiente para la tarea. Su salida presentó una atenuación excesiva y un desplazamiento hacia valores negativos, lo que afectó tanto la estructura temporal como la energía del contenido armónico. En la práctica, el modelo actuó de forma similar a un filtro pasabajas, lo que impidió recuperar detalles relevantes de las grabaciones originales. Además, las limitaciones computacionales de la herramienta utilizada no permitieron explorar configuraciones de mayor complejidad, lo cual restringió su capacidad para capturar relaciones temporales largas, que son esenciales para modelar la reverberación.

En contraste, la arquitectura TCN desarrollada en esta investigación mostró el comportamiento más estable y consistente. Su diseño basado en convoluciones dilatadas permitió capturar dependencias temporales amplias sin comprometer la eficiencia computacional. Los resultados fueron favorables en la mayoría de las canciones analizadas, especialmente en aquellas con mayor densidad espectral, donde la red mantuvo la claridad de los transitorios y logró reducir la reverberación de manera más efectiva que los demás métodos. Sin embargo, el entrenamiento de redes más profundas o con mayor número de canales estuvo limitado por la capacidad de cómputo del equipo utilizado. Las pruebas se realizaron en una *MacBook Pro* con chip *Apple M4 Pro*, equipada con una *CPU* de 14 núcleos, *GPU* de 20 núcleos, *Neural Engine* de 16 núcleos y 48 GB de memoria unificada. Aunque este hardware ofrece un rendimiento considerable para procesamiento de audio, resultó insuficiente para entrenar configuraciones TCN de mayor escala, debido al incremento exponencial en el uso de memoria y en el tiempo por época requerido. Esta restricción impidió evaluar arquitecturas más grandes que podrían haber permitido analizar si una capacidad adicional del modelo mejoraba aún más la reconstrucción de señales.

En conjunto, los resultados comparativos indican que los métodos basados en aprendizaje profundo, en particular la TCN desarrollada, ofrecen una solución más confiable y flexible para la deconvolución acústica en grabaciones reales. Mientras que KLMS y NLARX enfrentan limitaciones inherentes a su estructura y capacidad de modelado, la TCN logra un equilibrio adecuado entre calidad de reconstrucción, estabilidad y capacidad para manejar señales complejas, convirtiéndose en el enfoque más prometedor para continuar esta línea de investigación.

CAPÍTULO 10

Conclusiones

La presente investigación evaluó la eficacia de tres enfoques de aprendizaje automático aplicados a la deconvolución acústica: *kernel least mean squares* (KLMS), el modelo no lineal autoregresivo con entrada exógena (NLARX) y las redes neuronales convolucionales temporales (TCN). Para ello se consolidó una base de datos unificada compuesta por treinta y cuatro pares de señales limpia y degradada, estandarizadas en duración, alineadas mediante correlación cruzada y organizadas bajo una misma convención de nombres. El análisis se llevó a cabo empleando la *Kernel Adaptive Filtering Toolbox* para el método KLMS, la aplicación *System Identification* de MathWorks para el modelo NLARX y un entorno de entrenamiento en *PyTorch* para las TCN, lo que permitió una comparación coherente entre los tres enfoques.

Antes de abordar la deconvolución, los modelos se sometieron a una tarea de identificación de sistemas sintéticos para verificar su implementación. En este escenario controlado, el modelo NLARX mostró el mejor desempeño, lo que confirmó su capacidad para representar relaciones no lineales y proporcionó una referencia previa al análisis con señales reales.

En la deconvolución acústica aplicada a las grabaciones reales de la base consolidada, los resultados evidenciaron diferencias importantes entre los métodos. El KLMS presentó dificultades de convergencia y alcanzó una SNR de 0 dB, reflejando limitaciones tanto en estabilidad como en generalización. El modelo NLARX, pese a su solidez en la evaluación sintética, actuó como un filtro pasa bajas y obtuvo una SNR de –13.95 dB, sin recuperar adecuadamente los componentes de alta frecuencia. En contraste, las redes neuronales convolucionales temporales alcanzaron el mejor desempeño global, con un RMSE de 0.1723 y una SNR de –0.08 dB, preservando de forma más completa la estructura temporal y espectral de las señales degradadas.

En conjunto, los resultados evidencian un cambio claro en la metodología de esta línea de investigación. Si bien los filtros adaptativos pueden funcionar en escenarios controlados, muestran limitaciones importantes frente a grabaciones reales. En contraste, los métodos basados en aprendizaje profundo ofrecen una mejor capacidad para manejar la complejidad acústica y se perfilan como la ruta más prometedora para las próximas iteraciones, especialmente al explorar arquitecturas neuronales de mayor capacidad.

CAPÍTULO 11

Recomendaciones

A partir de los resultados obtenidos, se recomienda que las siguientes iteraciones de esta línea de investigación exploren arquitecturas de aprendizaje profundo de mayor capacidad. En particular, el uso de equipos con mejores prestaciones computacionales permitiría entrenar redes neuronales convolucionales temporales (TCN) más profundas y con tamaños de núcleo mayores, lo que ampliaría el campo receptivo efectivo y facilitaría la modelación de dependencias acústicas de larga duración. Esta mejora en capacidad de cómputo es relevante, dado que en el presente estudio el *hardware* disponible constituyó un cuello de botella que limitó la complejidad de las TCN empleadas.

Asimismo, se sugiere aprovechar infraestructura de cómputo especializada para realizar búsquedas más amplias de hiperparámetros y evaluar variantes arquitectónicas modernas, tales como mecanismos de atención o modelos secuenciales avanzados. Estas alternativas podrían aumentar la robustez del proceso de reconstrucción y mejorar la generalización en recintos acústicos no controlados.

De igual forma, se recomienda profundizar en la implementación del modelo NLARX utilizando herramientas más eficientes que la aplicación *System Identification* de MathWorks. Replicar estos experimentos con librerías dedicadas o con técnicas de optimización más avanzadas permitiría determinar si las limitaciones observadas provienen del propio modelo o de la herramienta utilizada para su estimación.

Finalmente, se sugiere que futuras investigaciones continúen consolidando y ampliando la base de datos empleada en este trabajo. Incorporar nuevas clases de señales degradadas, distintos escenarios acústicos y grabaciones distribuidas en el tiempo permitiría entrenar modelos más representativos y evaluar con mayor precisión la capacidad real de las arquitecturas profundas frente a condiciones de captura diversas.

CAPÍTULO 12

Referencias

[1] J. L. M. Cárdenas, «Deconvolución acústica para grabación en ambientes no tratados,» Trabajo de graduación, Universidad del Valle de Guatemala, Facultad de Ingeniería, Guatemala, 2020. dirección: https://repositorio.uvg.edu.gt/handle/123456789/3795.

[2] C. M. L. Flores, «Deconvolución acústica basada en filtros adaptativos y redes neuronales regresivas,» Trabajo de graduación, Universidad del Valle de Guatemala, Facultad de Ingeniería, Guatemala, 2021. dirección: https://repositorio.uvg.edu.gt/handle/123456789/3851.

[3] A. Calí, «Exploración de métodos de aprendizaje de máquina para mejorar aplicaciones enfocadas en deconvolución acústica en entornos no ideales,» Tesis inédita. Universidad del Valle de Guatemala, 2024, 2024.

[4] J. C. Principe, W. Liu y S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*. Hoboken, NJ: John Wiley & Sons, 2010, ISBN: 978-0-470-44753-6.

[5] D. S. Reay, *Digital Signal Processing Using the ARM Cortex-M4*. Wiley, 2015, ISBN: 978-1-118-85904-9.

[6] B. Ghojogh, A. Ghodsi, F. Karray y M. Crowley, «Reproducing Kernel Hilbert Space, Mercer's Theorem, Eigenfunctions, Nyström Method, and Use of Kernels in Machine Learning: Tutorial and Survey,» *arXiv preprint arXiv:2106.08443*, 2021. arXiv: 2106.08443 [stat.ML].

[7] OpenAI, *ChatGPT (GPT-5 Thinking)*, https://chat.openai.com/, [Herramienta de IA conversacional], OpenAI, 2025. visitado 28 de sep. de 2025.

[8] S. J. Prince, *Understanding Deep Learning*. The MIT Press, 2023. dirección: http://udlbook.com.

[9] G. Cybenko, «Approximation by superpositions of a sigmoidal function,» *Mathematics of Control, Signals and Systems*, vol. 2, n.º 4, págs. 303-314, 1989.

[10] K. Hornik, «Approximation capabilities of multilayer feedforward networks,» *Neural Networks*, vol. 4, n.º 2, págs. 251-257, 1991.

[11] F. Rosenblatt, «The Perceptron: A probabilistic model for information storage and organization in the brain,» *Psychological Review*, vol. 65, n.º 6, págs. 386-408, 1958.

[12] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016. dirección: https://www.deeplearningbook.org/.

[13] S. Bai, J. Z. Kolter y V. Koltun, «An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling,» *arXiv preprint arXiv:1803.01271*, 2018.

[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves et al., «WaveNet: A Generative Model for Raw Audio,» *arXiv preprint arXiv:1609.03499*, 2016.

[15] K. He, X. Zhang, S. Ren y J. Sun, «Deep Residual Learning for Image Recognition,» en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, págs. 770-778. DOI: 10.1109/CVPR.2016.90.

[16] I. J. Leontaritis y S. A. Billings, «Input-output parametric models for non-linear systems part I: deterministic non-linear systems,» *International Journal of Control*, vol. 41, n.º 2, págs. 303-328, 1985.

[17] MathWorks, *What Are Nonlinear ARX Models?* Disponible en: https://www.mathworks.com/help/ident/ug/what-are-nonlinear-arx-models.html, 2024.

[18] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. John Wiley & Sons, 2013.

[19] J. Deng, S. Zhong y A. Ordys, «Robustness and Accuracy Test of Particular Matter Prediction Based on Neural Networks,» *Communications and Network*, vol. 5, págs. 53-59, ene. de 2013. DOI: 10.4236/cn.2013.52B010.

[20] S. V. Vaerenbergh, *Kernel Adaptive Filtering Toolbox (KAFbox)*, ver. 2.0.0, MathWorks File Exchange. Accessed: 2025-09-29, 2014. dirección: https://www.mathworks.com/matlabcentral/fileexchange/46747-kernel-adaptive-filtering-toolbox.

CAPÍTULO 13

Anexos

En esta sección se incluyen los recursos complementarios que respaldan el desarrollo de la investigación y permiten la reproducción de los experimentos realizados. Estos materiales contienen tanto la base de datos consolidada empleada para las pruebas como el repositorio de código fuente donde se documentan las implementaciones utilizadas en las tareas de identificación de sistemas y de deconvolución acústica.

A.1 Base de datos consolidada

La base de datos unificada, compuesta por los treinta y cuatro pares de señales limpia–degradada utilizados en los experimentos, puede descargarse en el siguiente enlace:

- **Base de datos consolidada:**
  `https://uvggt.sharepoint.com/:f:/r/sites/Test399/Documentos%20compartidos/15%20-%20Robotat/otros/deconvoluciÃşn_dl/maldonado_2025/multimedia/audios/2%20-%20BDD%20Unificada?csf=1&web=1&e=N09WAK`

A.2 Repositorio de código fuente

El código empleado para la implementación de los filtros adaptativos, el modelo NLARX y las redes neuronales convolucionales temporales se encuentra disponible públicamente en GitHub:

- **Repositorio del código:**
  `https://github.com/PabloMaldo/Acoustic-Deconvolution`