
**UNIVERSIDAD AUTÓNOMA METRPOLITANA**

UNIDAD IZTAPALAPA

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

POSGRADO EN CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

# SÍNTESIS DE VOZ BASADA EN MODELOS OCULTOS DE MARKOV Y ALGORITMOS DE APRENDIZAJE PROFUNDO

TESIS QUE PRESENTA:

MARVIN COTO JIMÉNEZ

PARA OBTENER EL GRADO DE:

**DOCTOR EN CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN**

ASESOR: DR. JOHN GODDARD CLOSE

CIUDAD DE MÉXICO, NOVIEMBRE 2017



**UNIVERSIDAD AUTÓNOMA METROPOLITANA – IZTAPALAPA**
**DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA**

# SÍNTESIS DE VOZ BASADA EN MODELOS OCULTOS DE MARKOV Y ALGORITMOS DE APRENDIZAJE PROFUNDO

Tesis que presenta:
**Marvin Coto Jiménez**
Para obtener el grado de:
**Doctor en Ciencias y Tecnologías de la Información**

Asesor: Dr. John Goddard Close

Jurado calificador:

| | |
|---|---|
| Presidente: | Dr. John Goddard Close |
| Secretario: | Dr. Pedro Lara Velázquez |
| Vocal: | Dr. Luis Martín Rojas Cárdenas |
| Vocal: | Dr. René MacKinney Romero |
| Vocal: | Dr. Hugo Leonardo Rufiner Di Persia |

Ciudad de México, Noviembre 2017

# Síntesis de voz basada en Modelos Ocultos de Markov y Algoritmos de Aprendizaje Profundo


## Breve resumen:

El propósito principal de esta tesis es desarrollar sistemas basados en algoritmos de aprendizaje profundo para mejorar los resultados obtenidos con la síntesis de voz basada en Modelos Ocultos de Markov. La tesis está estructurada con base en las publicaciones realizadas durante su desarrollo, en los cuales se extendió la aplicación de sistemas de post-filtros hacia múltiples arquitecturas en paralelo. También se incluyen aplicaciones afines donde la propuesta ha mostrado su utilidad.



**Palabras clave:** Síntesis de voz, LSTM, Post-filtros, HMM, Aprendizaje profundo

# Lista de contribuciones

Durante el desarrollo de esta tesis doctoral se produjeron las siguientes publicaciones:

1. Artículos en revistas:

    *a*) Coto-Jiménez, M., Goddard-Close, J. "LSTM Deep Neural Networks Postfiltering for Enhancing Synthetic Voices." In: International Journal of Pattern Recognition and Artificial Intelligence. Vol. 32.3 (2018), 23 pages (en prensa). Doi: https://doi.org/10.1142/S021800141860008X. En este trabajo se desarrolla la propuesta de utilización de colecciones de redes LSTM como post-filtros para mejorar los resultados de la síntesis de voz basada en HMM, lo cual constituye el fundamento del Capítulo 4 de la presente tesis y la base de los sistemas en los capítulos 5,6,9 y 10.

2. Artículos en congresos internacionales:

    *a*) Coto-Jiménez, M, Goddard-Close, J. "Speech Synthesis Based on Hidden Markov Models and Deep Learning." Advances in Pattern Recognition: 19 (2016) pp. 19-28. En esta publicación se describió el problema de la síntesis de voz basada en HMM, sus limitaciones y las perspectivas de los algoritmos de aprendizaje profundo para solventarlas. Los resultados de esta publicación han sido integrados principalmente en el Capítulo 3.

    *b*) Coto-Jiménez, M, Goddard-Close, J. "LSTM Deep Neural Networks Postfiltering for Improving the Quality of Synthetic Voices". Pattern Recognition: Proceedings of the 8th Mexican Conference, MCPR'16, Guanajuato, Mexico, Jun. 22-25, 2016. 9703, (2016) pp. 280-289. Este artículo contiene la primer propuesta de utilización de una colección de redes profundas (*autoencoders*) para mejorar los resultados de la síntesis de voz, de forma preliminar a la publicación 1.a).

    *c*) Coto-Jiménez M., Goddard-Close J., Martínez-Licona F.M. "Quality Assessment of HMM-Based Speech Synthesis Using Acoustical Vowel Analysis." In: Proceedings of the International Conference on Speech and Computer SPECOM'14, Oct. 5, (2014) pp. 368-375. Springer International Publishing. Este artículo presentó la posibilidad de analizar únicamente las vocales del habla sintetizada para predecir la calidad del resultado, de forma independiente a medidas objetivas y subjetivas. La investigación en las medidas de evaluación ha enriquecido los capítulos 4, 5 y 6 de la presente tesis, y especialmente el Capítulo 8 en la evaluación del cambio de acento.

*d*) Coto-Jiménez M, Martínez-Licona F.M., Goddard-Close J. "Acoustic Vowel Analysis in a Mexican Spanish HMM-based Speech Synthesis." Spanish HMM-based Speech Synthesis". Research in Computing Science, 86 (2014) pp. 53-62. Semejante al anterior, se presentaron opciones a los procedimientos de evaluación tradicional basada en análisis acústico de las vocales.

*e*) Coto-Jiménez, M., Goddard-Close, J. "Hidden Markov Models for Artificial Voice Production and Accent Modification." Proceedings of the Ibero-American Conference on Artificial Intelligence. Springer International Publishing (2016) pp. 415-426. Este trabajo mostró la aplicación de la técnica de creación de voces con la técnica de adaptación en la modificación de acento de voces en castellano de México y europeo. Los resultados han sido integrados en el Capítulo 8 de la presente tesis.

*f*) Coto-Jiménez, M, Goddard-Close, J., Martínez-Licona, F.M. "Improving Automatic Speech Recognition Containing Additive Noise Using Deep Denoising Autoencoders of LSTM Networks." International Conference on Speech and Computer. Springer International Publishing (2016) pp. 354-361. Este artículo mostró la utilidad de utilizar un conjunto de redes LSTM para eliminar ruido en el habla, y su ventaja sobre otros modelos de redes neuronales. Estos resultados han sido ampliados, junto con una propuesta adicional, en los Capítulos 9 y 10 de esta tesis.

3. Otras publicaciones (artículos de divulgación):

*a*) Coto-Jiménez M., Goddard-Close, J. Martínez-Licona, F.M. "Producción artificial del habla: Síntesis de voz". Komputer Sapiens. Revista de la Sociedad Mexicana de Inteligencia Artificial. 7.3 (2015), pp. 6-10.

*b*) Coto-Jiménez M., Goddard-Close, J. "Producción de habla artificial en español con acento mexicano ContactoS". ContacOS, Revista de educación en ciencias e ingeniería (2016).

*c*) Coto-Jiménez, M. Goddard-Close, J. "LSTM: Redes neuronales con memoria a corto y largo plazo". ContacOS, Revista de educación en ciencias e ingeniería. (2015).

*d*) Coto-Jiménez, M. Goddard-Close, J. "Las tecnologías del habla como recurso para la preservación de lenguas en peligro de extinción. ContacOS, Revista de educación en ciencias e ingeniería (2016).

4. Software desarrollado:

*a*) HTS-Parallel: Modificación de HTS para producir una versión sintetizada de la base de datos, alineada ventana a ventana. Con esta variante del HTS fueron producidas las voces basadas en Modelos Ocultos de Markov de los capítulos 4, 5 y 6. Disponible en: `https://github.com/mcoto/HTS-ParallelTraining`

# Contenidos

# Lista de Figuras

# Lista de Tablas

# Lista de Algoritmos

# Acrónimos

$f_0$ Frecuencia fundamental

AE *Autoencoder*

ANOVA Análisis de varianza

ASR Reconocedor automático de voz *(Automatic Speech Recognizer)*

BAM Memoria asociativa bidireccional *(Bidirectional Associative Memory)*

CMLLR Regresión lineal de máxima probabilidad restringida *(Constrained Maximum Likelihood Linear Regression)*

CS Acento castellano europeo

DBN Redes de creencia profunda *(Deep Belief Networks)*

DLSTM-N Sistema de eliminación de ruido basado en LSTM

DNN Red neuronal profunda *(Deep Neural Network)*

DTW Adaptación de tiempo dinámico *(Dynamic Time Warping)*

GMM Mezcla de gaussianas

GPU Unidad de procesamiento gráfico

HMM Modelos Ocultos de Markov

HSD Prueba estadística de diferencia significativa honesta *(Honest Significant Difference)*

HTS Sistema de creación de voces basado en HMM

klt *Generalized Subspace Approach*

logmmse *Log-Spectral Amplitude Estimator*

LSTM Modelo de memoria de corto y largo plazo

LSTM-N Sistema de post-filtro basado en LSTM

MAA Memoria Auto-asociativa

MAD Media de la diferencia espectral

MFCC Coeficientes Cepstrales en la escala de Mel *(Mel-Frequency Cepstral Coefficients)*

MLLR Regresión lineal de máxima probabilidad *(Maximum Likelihood Linear Regression)*

MLSA Filtro de reconstrucción de señal a partir del espectro en escala de Mel *(Mel Log Spectrum Approximation)*

mmse *Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator*

MS Acento castellano de México

MSD Distribución de probabilidad multi-estado *(Multi-State Distribution)*

mss *Multi-band Spectral Subtraction*

PESQ Evaluación de la calidad vocal por percepción *(Perceptual Evaluation of Speech Quality)*

PLP Predicción perceptual lineal *(Perceptual Linear Predictive)*

RNN Redes neuronales recurrentes *(Recurrent Neural Networks)*

SAMPA Alfabeto fonético legible por computadora *(Speech Assessment Methods Phonetic Alphabet)*

$\text{SegSNR}_{\text{f}}$ Relación señal a ruido segmental

SNR Relación señal a ruido *(Signal to Noise Ratio)*

specsub *Spectral subtraction*

STSA Amplitud espectral de tiempo corto *(Short-time spectral amplitude)*

WER Tasa de error de reconocimiento de palabras

WSS Pendientes del espectro con pesos *(Weighted Spectral Slope)*

## Resumen

En esta tesis se aborda el problema de mejorar los resultados de la síntesis estadística paramétrica de voz basada en Modelos Ocultos de Markov (HMM), utilizando algoritmos de aprendizaje profundo. El tema ha cobrado mayor importancia en épocas recientes debido a la presencia cada vez mayor de voces artificiales en diversos dispositivos y aplicaciones, en los cuales existe la necesidad de perfeccionar los resultados de manera que el sonido de la voz sintetizada se acerque a la naturalidad y expresividad del habla humana.

La síntesis de voz basada en HMM se difundió durante la segunda mitad de la década de 2000, gracias a su probada capacidad para generar voces con menos recursos y mayor flexibilidad que otras técnicas. Por esta razón, el interés de los principales grupos de investigación del mundo en este tema se volvió hacia perfeccionar sus resultados, a partir de la década de 2010.

En la presente tesis se realizan tres propuestas para mejorar estos resultados: la primera utiliza post-filtros basados en redes neuronales de memoria a corto y largo plazo (LSTM), la segunda una combinación con filtros Wiener, y la tercera un nuevo enfoque discriminativo. A diferencia de las propuestas preliminares que se encuentran en la literatura, las de esta tesis tienen como base colecciones de diversas arquitecturas, tales como autocodificadores (*autoencoders*) y memorias auto-asociativas, las cuales se entrenan y aplican de acuerdo con subconjuntos de parámetros del habla. De esta manera, los resultados alcanzados superan intentos previos en los que se considera un único modelo, principalmente enfocado a las componentes espectrales de las voces.

Adicionalmente, se presentan dos aplicaciones donde la propuesta de utilización de síntesis de voz basada en HMM y los sistemas de post-filtros basados en algoritmos de aprendizaje profundo muestran buenos resultados. La primera es el cambio de acento en voces, área poco explorada para variantes de la lengua castellana. La segunda es la reducción de ruido en señales degradadas tanto con ruidos naturales como artificiales.

Tanto los sistemas de post-filtros para la síntesis de voz, como las aplicaciones adicionales, incluyen combinaciones de los algoritmos de aprendizaje profundo con otros clásicos en el tema de mejoramiento de señales de habla. El trabajo permite vislumbrar nuevas líneas de investigación en el tema de síntesis de voz y de mejora de señales de habla en presencia de ruido.

## Abstract

This thesis addresses the problem of improving the results of statistical parametric speech synthesis using deep learning algorithms. The subject has become more important in recent times due to the increasing presence of artificial voices in several devices and applications. In these, there is a need to refine the results so that the sound of a synthetic voice approaches the naturalness and expressiveness of human speech.

HMM-based voice synthesis became a hot topic after the second half of the 2000s thanks to its proven ability to generate speech with small amounts of data as well as its increased flexibility compared to other techniques. For this reason, the interest of the world's leading research groups in this area turned to refine their results.

In this work, three proposals are made to improve those results using post-filters based on LSTM deep neural networks. Unlike the preliminary proposals found in the literature, are based on collections of various architectures, such as auto-encoders and auto-associative memories, which are trained and applied according to subsets of speech parameters. In this way, the results achieved surpass previous attempts in which it is considered a single model that is mainly focused on the spectrum of voices.

Also, two applications are presented where the use of HMM-based speech synthesis and post-filter systems based on deep learning algorithms show good results. The first is the change of accent in voices, a little-explored area for the variants of the Castilian Spanish. The second is noise reduction in signals with both natural and artificial noise.

Both post-filter systems for speech synthesis, as well as the additional applications, include combinations of algorithms with other classical speech signal improvement. The work presented here allows us to glimpse new areas of research in the topic of speech synthesis and enhancement of speech signals in the presence of noise.

# Prefacio

Esta tesis fue presentada en el Posgrado en Ciencias y Tecnologías de la Información de la Universidad Autónoma Metropolitana, en la Ciudad de México, como requisito parcial para obtener el grado de Doctor en Ciencias (Ciencias y Tecnologías de la Información), en la especialidad de Sistemas Inteligentes. El trabajo realizado se extendió entre los años 2014 a 2017 en el Laboratorio de Habla y Reconocimiento de Patrones de esta universidad.

El tema es continuación directa de la tesis de maestría "Síntesis estadística paramétrica de voz", realizada en esta misma casa de estudios entre los años 2012 a 2014. A partir de esa primera experiencia generando habla de forma artificial, se determinó la necesidad de mejorar los resultados que se habían obtenido hasta el momento con voces artificiales basadas en Modelos Ocultos de Markov.

Para esto se ha desarrollado un conjunto de algoritmos, software y sistemas, que se combinan a manera de post-filtros en la salida de la síntesis basada en HMM para mejorar sus resultados. Durante la investigación se produjeron artículos de revista, publicaciones de conferencia y artículos de divulgación en torno al tema. Parte de estos resultados han sido incluidos en los capítulos de la tesis.

Con el avance de los sistemas desarrollados se vieron oportunidades de aplicación en áreas afinas a la síntesis de voz, especialmente en la mejora de habla en presencia de ruido. Este es un área de investigación muy activa y que genera gran interés en la comunidad de investigación en tecnologías del habla. Es por esto que se ha incluido una segunda parte de la tesis para mostrar las propuestas y resultados obtenidos en esta línea.

## Objetivos de la tesis

El objetivo principal de la tesis es diseñar y evaluar estrategias de implementación de algoritmos de aprendizaje profundo que permitan mejorar los resultados de la síntesis de voz basada en Modelos Ocultos de Markov. Para esto se tienen como objetivos particulares: 1) Estudiar y analizar las propuestas para mejora del habla sintetizada presentadas en la literatura. 2) Identificar

algoritmos de aprendizaje profundo aplicables a la mejora de señales de habla. 3) Diseñar propuestas basadas en aprendizaje profundo para mejorar señales de voz sintetizada. 4) Establecer áreas de oportunidad para aplicar las propuestas en nuevos contextos relacionados con síntesis de voz y mejora de señales de voz.

## Estructura del documento

La tesis reúne una amplia gama de resultados, organizados en dos partes, con dos capítulos de introducción y antecedentes. La primera parte (Post-filtros basados en algoritmos de aprendizaje profundo para la mejora de la síntesis de voz basada en HMM) contiene tres capítulos con una propuesta, experimentación y análisis de resultados independientes. La segunda parte (Otras aplicaciones de los sistemas propuestos) contiene tres capítulos que cuentan también con independencia en su experimentación y resultados. Cada una de estas partes tiene un capítulo introductorio con los principios teóricos y antecedentes. En el Capítulo 11 se presentan las conclusiones y las líneas de investigación que puede seguir la investigación futura.

*Dedicada a Alejandro, Gabriel y Andrea.*

## Agradecimientos

Al **Consejo Nacional de Ciencia y Tecnología (CONACyT)** por haber otorgado el financiamiento que ha hecho posible la realización de este proyecto de investigación, así como a la **Universidad de Costa Rica** por el apoyo económico y administrativo brindado, gracias al cual ha sido posible desarrollar el gran proyecto de estudios de cinco años en México.

A mi asesor, **Dr. John Goddard Close** por la confianza depositada en mi persona, su decisivo impulso intelectual y personal, y sus sabios consejos a lo largo de todo el proceso de estudio e investigación. Mi total admiración hacia su persona

A **Andrea**, **Gabriel** y **Alejandro** por todo el amor, apoyo y armonía ideales para poder luchar y entregarme a este proyecto de estudio.

A mi mamá **Mayra** y a mis hermanos **Kattia** y **Jorge**, quienes han creído y apoyado este proyecto de estudios, y con su amor fraterno inspiran cada día, junto a sus hijos **Adrián** y **Sebastián**.

Al **Dr. Alfonso Prieto Guerrero** y al **Dr. Sergio de los Cobos Silva** que ofrecieron el apoyo académico, administrativo y personal que me abrió las puertas a la Universidad Autónoma Metropolitana y a México.

A los sinodales **Dr. Hugo Leonardo Rufiner**, **Dr. Pedro Lara Velásquez**, **Dr. Luis Martín Rojas Cárdenas** y **Dr. René MacKinney Romero** por darme el honor de formar parte del jurado revisor y calificador para la disertación pública de este trabajo, y por haberlo enriquecido con sus valiosas aportaciones.

Al **Dr. Leandro Di Persia** por su aporte en la teoría y en el software utilizados, ambos de vital importancia para el desarrollo de la tesis.

A los funcionarios del **Departamento de Asuntos Internacionales y Cooperación Externa** de la Universidad de Costa Rica, por su apoyo y orientación en tantos trámites y procesos administrativos.

Al **Dr. Javier Trejos Zelaya** y **Dr. Jorge Romero Chacón** por haber creído en mi persona y apoyado desde la Universidad de Costa Rica este proyecto de estudios.

Al **Dr. Enrique Rodríguez de la Colina**, coordinador del posgrado, a los **profesores y compañeros de generación** por los conocimientos, orientación y ayuda brindada en tantos aspectos.

Al **Dr. Orlando Arrieta Orozco**, director de la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica por su orientación, apoyo y voluntad que permitió la conclusión del proyecto de estudios.

A la comunidad costarricense en México, por sus muestras de solidaridad y apoyo en los momentos más difíciles.

# 1

# INTRODUCCIÓN

*En el presente capítulo se describen los principales elementos que sustentan el problema de investigación, así como una visión general de los métodos y objetivos planteados para abordarlo.*

## Índice

## 1.1 Presentación

El habla es la forma de comunicación más natural entre seres humanos. Por esta razón, trasladar la capacidad humana de utilizar la voz para intercambiar información con los dispositivos tecnológicos que forman parte de la sociedad actual es de gran pertinencia, y ha despertado el interés de investigadores en las últimas décadas [3], aunque los primeros intentos de generar dispositivos de habla datan de hace varios siglos. La síntesis de voz equivale, en el proceso de comunicación, a la producción de habla de forma artificial.

Dada la proliferación e interconexión de dispositivos como teléfonos celulares, tabletas y sistemas de navegación, con información que por su longitud o por su naturaleza cambiante no puede ser pregrabada, la necesidad de utilizar síntesis de voz con características de inteligibilidad y naturalidad son crecientes. Los sistemas de síntesis de voz actuales han logrado importantes avances en cuanto a la inteligibilidad de sus resultados, sin embargo, la naturalidad es aún un obstáculo para su utilización más extendida y la creación de nuevas áreas de oportunidad para su aplicación.

Cuando los niveles de naturalidad permitan a las voces artificiales asemejarse más a la voz humana, con características como la variabilidad y la expresión, nuevas aplicaciones pueden surgir y extenderse. Entre estas se pueden mencionar los asistentes virtuales inteligentes, la reconstrucción de voces de personajes históricos, la traducción voz a voz en tiempo real que conserve el tono y la expresión, la inclusión de cambios de acento en aplicaciones, y el doblaje automático de películas.

La síntesis paramétrica de voz basada en el uso de Modelos Ocultos de Markov (*Hidden Markov Models*, HMM) surgió en la década del año 2000 como una técnica destacada por la flexibilidad de sus resultados y sus bajos requerimientos. Sus características la hacen adecuada para incorporar nuevos algoritmos con los cuales se pueda experimentar ampliamente y buscar solventar las limitaciones que ha presentado, principalmente en cuando a naturalidad.

En la siguiente sección se describen los elementos principales de este tipo de síntesis.

## 1.2 Síntesis de voz basada en HMM

En la síntesis de voz basada en HMM los parámetros que modelan el habla (tales como el espectro, la frecuencia fundamental y la duración de cada sonido), se modelan y generan

estadísticamente. Los detalles de los procesos involucrados en la definición, entrenamiento y generación de parámetros de habla han sido descritos con formalidad en referencias recientes [4]. En esta sección se resumen algunos de los principales aspectos que forman parte de todo el proceso con la finalidad de introducir sus alcances y limitaciones.

### 1.2.1 Modelos Ocultos de Markov (HMM)

Un HMM se pueden definir como una tupla $\lambda = (S, \pi, a, b)$ donde:

- $S = 1, \ldots, m$ es un conjunto finito de estados.
- $\pi$ es un vector de probabilidades iniciales.
- $a$ es una matriz de transición de probabilidades.
- $b$ es una matriz de probabilidades de salida.

Se pueden caracterizar como procesos estocásticos dobles: un primer proceso estocástico describe la transición entre estados, y el segundo las salidas. El comportamiento del primer proceso en el tiempo $t$ depende solamente del estado predecesor,

$$p(S_t|S_1, S_2, \ldots, S_{t-1}) = p(S_t|S_{t-1}), \tag{1.1}$$

donde $p(S_t)$ indica la probabilidad del proceso de estar en el estado $S$ en el tiempo $t$.

En el segundo proceso estocástico, en cada instante de tiempo $t$ se genera una salida $O_t$, la cual tiene una distribución de probabilidad asociada, dependiente solamente del estado actual. Esto se puede describir como

$$p(O_t|O_1 \ldots O_{t-1}, S_1, \ldots, S_t) = p(O_t|S_t). \tag{1.2}$$

Los HMM utilizados en síntesis de voz son del tipo izquierda a derecha, donde las transiciones entre estados van hacia el estado inmediato siguiente a la derecha o hacia sí mismos. En la Figura 1.1, se muestra una representación de un HMM de tres estados con estas características. En esta, $p_{ij}$ representa la probabilidad de transición del estado $i$ al estado $j$, y $O_k$ representa la observación emitida en el estado $k$.

En la síntesis basada en HMM, las formas de onda de habla se pueden reconstruir a partir de secuencias de parámetros generadas por medio de vectores emitidos en cada estado [5]. Una implementación típica de esta modelo incluye vectores cuyas entradas consisten en:

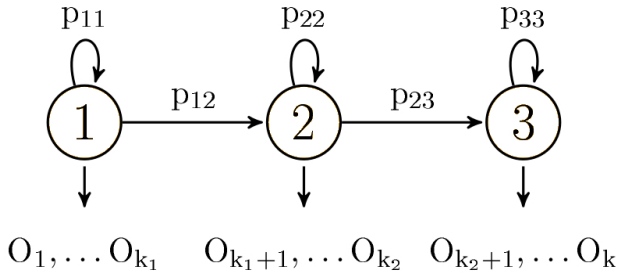



**Figura 1.1:** Ejemplo de un HMM tipo izquierda a derecha con tres estados

- Frecuencia fundamental ($f_0$)
- Coeficientes espectrales en la escala de mel (*Mel-frequency cepstral coefficients*, MFCC): Se obtienen a partir de la transformada discreta de Fourier en ventanas de la señal, cuyos resultados se mapean en la escala de Mel, a partir de filtros triángulares que representan la escala de percepción humana. Son ampliamente utilizados en aplicaciones de habla, partiendo del reconocimiento de voz.
- Aproximaciones de la primera y segunda derivada de los parámetros anteriores, llamadas delta y delta-delta.

La herramienta más utilizada para construir voces basadas en HMM es llamada HTS [6].

### 1.2.2 Parámetros del habla

El proceso inicia con la extracción de parámetros de una forma de onda de habla, los cuales consisten en un conjunto de MFCC y del coeficiente para $f_0$. Para el caso de habla, $f_0$ tiene la particularidad de tomar valores positivos o cero en una frase, dado que existen sonidos producidos con la vibración de las cuerdas vocales o sin ésta. Dado que la emisión de los estados de los HMM es determinado por distribuciones de probabilidad, las cuales se ajustan en el proceso de entrenamiento, se requiere de una distribución especial para modelar las particularidades de $f_0$. Para esto, en [7] se introdujo una *distribución de probabilidad multi-estado* (MSD), la cual combina valores numéricos (para aquellos sonidos con $f_0 > 0$) y símbolos discretos (para los sonidos con $f_0 = 0$).

Un ejemplo de contorno de $f_0$ de un fragmento de audio se muestra en la Figura 1.2, donde se señalan segmentos con valores positivos de $f_0$ (sonoros) y valores de $f_0 = 0$ (no sonoros).

Para incorporar elementos que contienen información de la variabilidad de los parámetros a lo largo del tiempo, se utilizan los coeficientes $\Delta$ y $\Delta\Delta$, los cuales representan aproximaciones de la primera y segunda derivadas a lo largo del tiempo. Estos coeficientes forman parte de los parámetros de habla que aprenden los HMM y por lo tanto que emiten sus estados durante el proceso de la síntesis. Estos parámetros se definen:

$$\Delta x_i = \frac{1}{2}(x_{i+1} - x_{i-1}) \tag{1.3}$$

$$\Delta\Delta x_i = x_{i-1} - 2x_i + x_{i+1} \tag{1.4}$$

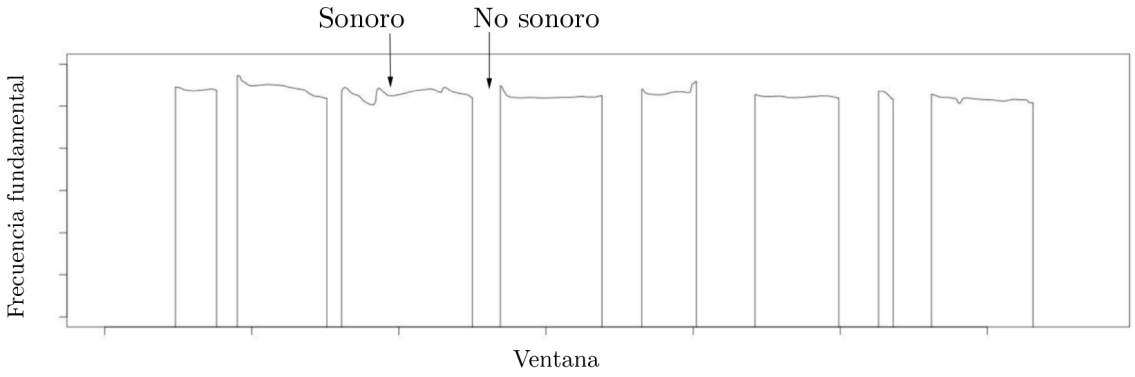



**Figura 1.2:** Contorno de $f_0$ para una frase de habla

### 1.2.3 Agrupamiento de acuerdo con el contexto

Es sabido que cada fonema pronunciado por una persona difiere de acuerdo con su contexto, es decir, por su posición en la palabra, en la frase o si está acentuada, entre otros factores. Estas diferencias se reflejan en términos de energía y tono, lo cual se conoce como información prosódica y variaciones por coarticulación.

Para representar adecuadamente estas variaciones, es necesario diferenciar los fonemas tomando en cuenta su contexto. El problema principal al estimar los parámetros de los fonemas de acuerdo con su contexto es la gran cantidad de éstos que existen en el habla, y por lo tanto la gran cantidad de grabaciones de un mismo hablante que se requerirían para entrenar los HMM de manera que se cuenta con suficientes ejemplos de cada uno.

Para resolver este problema, en [8] y [1] se formuló un criterio de agrupamiento de fonemas utilizando árboles de decisión, los cuales se generan de acuerdo con la cantidad de sonidos disponibles. Este agrupamiento se ve reflejado en primer lugar en la similitud de los fonemas presentes en la base de datos y la cantidad de los mismos. Los estados de los HMM se agrupan de acuerdo con lo definido por los árboles de decisión y de esta manera se estiman sus distribuciones de probabilidad. En el momento de la síntesis, las distribuciones de probabilidad de cada estado se seleccionan de acuerdo con el fonemas y su contexto, siguiendo el árbol de decisión, tal como se muestra en la Figura 1.3.

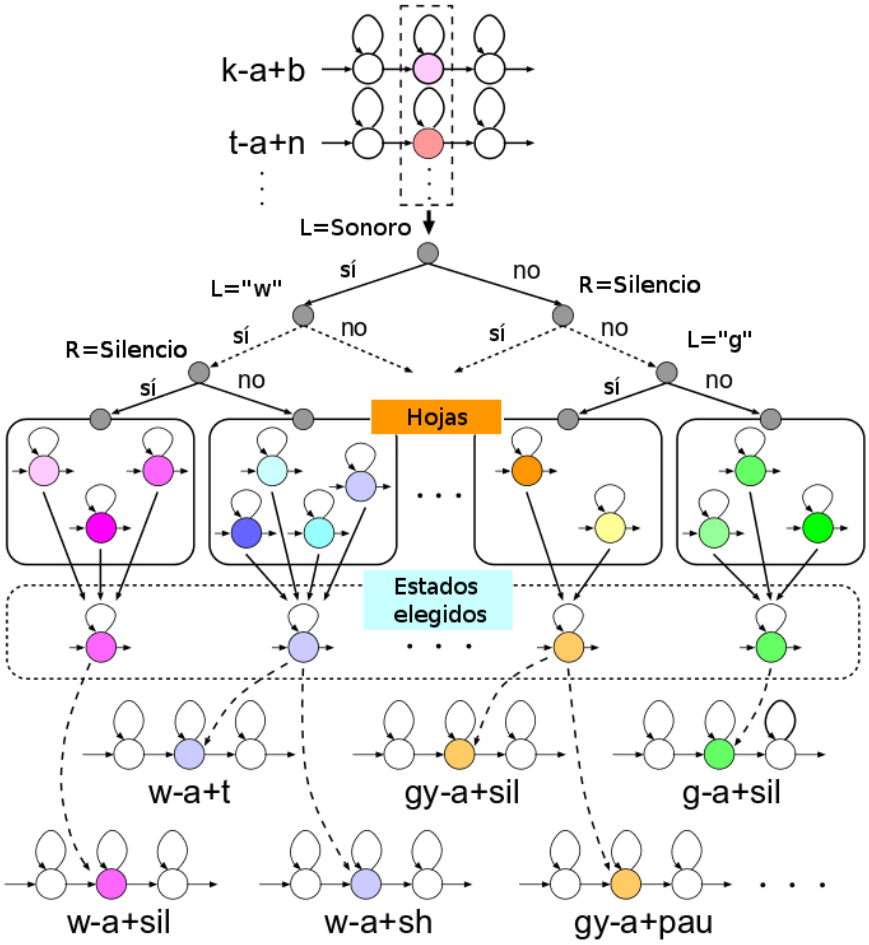



**Figura 1.3:** Agrupamiento de estados HMM para su entrenamiento y utilización en el proceso de síntesis. En las hojas se encuentran los HMM para sonidos específicos del habla (fonemas). Adaptada de [1].

Los árboles de decisión que se utilizan para agrupar los estados de los HMM son distintos para los valores de $f_0$ y los MFCC. Adicionalmente, se utilizan distribuciones de probabilidad para modelar cuántos vectores se emiten en cada estado de los HMM, y de esta manera establecer la duración de cada fonema. En [9] se propone que con distribuciones gaussianas se puede modelar

esta duración, y que éstas pueden determinarse en el entrenamiento utilizando información estadística de los fonemas en la base de datos. De esta manera, la tasa de habla de la voz sintetizada también se aproximará a la del hablante de la cual se generó.

### 1.2.4 Síntesis

El proceso de síntesis inicia con un texto, el cual se debe convertir en especificaciones que se pueden identificar con los HMM entrenados, los cuales contienen etiquetas del fonema y el contexto que representan, como se muestra en la Figura 1.4. Aquí "$-k+$" representa el fonema actual, "$\wedge\#-$" el previo (un silencio), "$+u0 =$" el siguiente y "$= a1$" el posterior a éste. @1_3 significa que es el primer fonema de una sílaba, en la cual restan tres fonemas más. Cada fonema corresponde con una etiqueta, por lo tanto con un HMM particular de $f_0$, MFCC y duración.

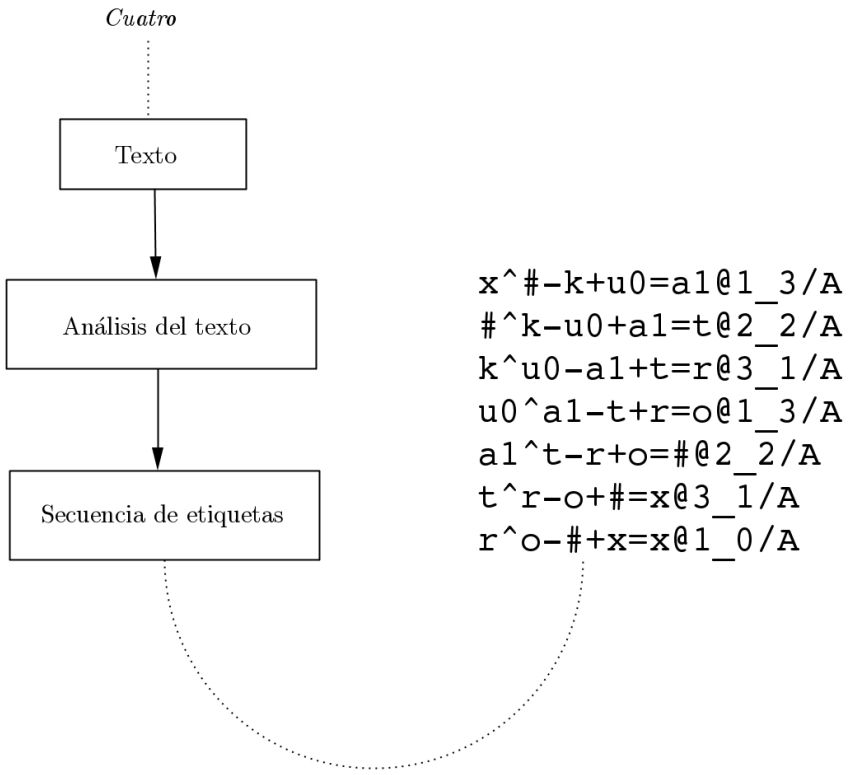



**Figura 1.4:** Generación de etiquetas para una frase de habla.

Al determinar todos los HMM que corresponden con la etiqueta se puede generar la secuencia de parámetros de los estados de los HMM. Finalmente, los coeficientes se procesan mediante

un filtro de reconstrucción de señal a partir del espectro en escala de Mel (*Mel Log Spectrum Approximation*, MLSA) [10][11][12], el cual genera la señal de audio correspondiente al habla a partir de los parámetros.

En el sistema HTS (*HMM-based Speech Synthesis System*), este filtro está implementado por medio del complemento *h*ts-engine, el cual genera la forma de onda con las salidas de los HMM.

## 1.3 Alcance y limitaciones de la síntesis de voz basada en HMM

A partir de los primeros desarrollos de la síntesis de voz basada en HMM, para los idiomas japonés e inglés, y la difusión del sistema de creación de voces HTS (en su primera versión del año 2002), la cantidad de implementaciones en múltiples idiomas ha sido creciente a lo largo del mundo [13] [14][15][16][17][18][19][20][21][22][23][24]. Esto ha incluido tanto voces en distintos idiomas y acentos, así como voces cantadas [25], conversión de voz a canto[26], cambio de acento y mezcla de voces.

A pesar de estas características, las desventajas de la técnica han sido reportadas en numerosas de estas implementaciones. Las principales son:

- Presencia de sonidos semejantes a zumbidos: Este efecto adverso es producido principalmente por la manera en que se modela el habla y el filtro que genera la forma de onda. En la síntesis basada en HMM, se considera que los sonidos con $f_0 > 0$ tienen una fuente puramente periódica, mientras que los sonidos con $f_0 = 0$ tienen como fuente ruido blanco. Este modelado se considera muy simplista y origen de este tipo de distorsión [27].
- Sonido generalmente apagado, en comparación con el habla natural: La fuente de este problema está en la naturaleza estadística del entrenamiento de los HMM. Como se mencionó en la sección anterior, existe un proceso de agrupamiento de estados en los HMM en el cual se promedian valores de los parámetros de los fonemas con los que se cuenta en la base de datos. El efectuar este promedio elimina elementos de variabilidad propios del habla natural, pero es requerido para poder entrenar adecuadamente los HMM.

También ha sido señalada la limitación de los árboles de decisión para modelar dependencias de contexto complejas, tales como énfasis específicos a nivel de palabras [28]. Es conocido que la calidad de los sistemas de habla artificial basados en HMM no alcanzan aún la de los mejores sistemas basados en tecnologías de concatenación de unidades dominantes en las aplicaciones actuales [29], lo cual ha llevado a un esfuerzo por mejorar sus resultados en dos sentidos:

1. Sustituir los HMM y los árboles de decisión del proceso actual por redes profundas, en las cuales la codificación del texto pueda utilizarse como entrada a la red, y los parámetros de habla sean la salida de esta red.
2. Utilizar post-filtros para mejorar los resultados de la síntesis, conservando todo el proceso de los HMM y árboles de decisión e incorporar en una última etapa redes profundas u otros algoritmos que acerquen los parámetros generados a los de una voz natural.

En la siguiente sección se describe el problema de investigación planteado para abordar la segunda de las posibilidades mencionadas.

## 1.4 Problema de investigación

La síntesis de voz se ha implementado en años recientes en múltiples aplicaciones. La técnica predominante en estas implementaciones ha sido la concatenación de unidades, en la cual se pueden crear nuevas frases a partir de segmentos pregrabados, trabajando directamente con las ondas sonoras.

La síntesis de voz basada en HMM u otros modelos paramétricos ha despertado gran interés por sus menores requisitos y su mayor flexibilidad que las técnicas de concatenación de unidades. Sin embargo, hasta ahora su calidad en términos de naturalidad no ha alcanzado a la de los mejores sistemas concatenativos.

Dadas las ventajas que presenta la síntesis basada en HMM, resulta de interés la incorporación de algoritmos que mejoren la calidad del audio resultante, de manera que se preserven sus ventajas, y la mejora en calidad abra nuevas áreas de oportunidad para la aplicación de voces artificiales. Algunos de estos algoritmos, basados en redes neuronales profundas, han surgido de forma incipiente, adaptando ideas utilizadas como etapas previas al reconocimiento o procesamiento de señales de habla (post-filtros), mostrando sus ventajas en estas áreas de investigación de tecnologías del habla, afines a la síntesis de voz.

Por lo tanto se destaca la importancia de desarrollar sistemas y algoritmos que permitan incorporar nuevas etapas a los procesos de creación de habla sintetizada con HMM para mejorar su calidad. Dado que las distorsiones y las componentes de ruido generadas en la síntesis basada en HMM son de naturaleza desconocida, existe la posibilidad de que un mapeo entre habla artificial y natural pueda generarse directamente de los datos, como un problema de regresión.

Este proceso puede llevarse también a otras áreas, tales como el mejoramiento de señales de voz donde el ruido es conocido, para proponer mejoras que sean aplicables en reconocimiento y mejora de señales.

## 1.5 Objetivo general

El objetivo general de la presente tesis se enuncia:

> Diseñar y evaluar estrategias de implementación de algoritmos de aprendizaje profundo que permitan mejorar los resultados de la síntesis de voz basada en Modelos Ocultos de Markov.

## 1.6 Objetivos particulares

1. Estudiar y analizar las propuestas para mejora del habla sintetizada presentadas en la literatura.
2. Identificar algoritmos de aprendizaje profundo aplicables a la mejora de señales de habla.
3. Diseñar propuestas basadas en aprendizaje profundo para mejorar señales de voz sintetizada.
4. Establecer áreas de oportunidad para aplicar las propuestas en nuevos contextos relacionados con síntesis de voz y mejora de señales de voz.

## 1.7 Procedimientos generales de la investigación

Para cumplir con los objetivos planteados en esta tesis se han seguido los siguientes procedimientos generales:

1. Identificación y comprensión de la literatura científica relacionada con la síntesis de voz basada en HMM y las propuestas de mejora a la calidad de voces resultantes.
2. Identificación y comprensión de la literatura científica relacionada con algoritmos de aprendizaje profundo adaptables a la mejora de señales de habla.
3. Estudio y comprensión del sistema de creación de voces HTS, para adaptarlo a un sistema de mejora de voces que permita la experimentación con algoritmos de aprendizaje profundo.

4. Desarrollo de algoritmos para incorporar redes neuronales profundas a la mejora en señales de voz con ruido.
5. Diseño y análisis experimental de los algoritmos propuestos, en comparación con métodos clásicos de mejora de señales de voz.
6. Desarrollo de algoritmos para incorporar redes neuronales profundas a la mejora en señales de voz artificiales que hayan sido alineadas con voces naturales mediante el sistema HTS adaptado.
7. Diseño y análisis experimental de los algoritmos propuestos para mejorar las voces artificiales basadas en HMM.
8. Análisis de resultados mediante pruebas estadísticas para determinar mejoras significativas.

## 1.8 Estructura de la tesis

El documento de tesis está organizado de la siguiente manera: En el Capítulo 2 se describe el modelo principal de redes neuronales que se propone incorporar a la síntesis de voz, llamado LSTM. Posteriormente, la tesis se divide en dos partes:

La primera abarca las propuestas realizadas para mejorar las voces HMM. La descripción general de esta parte se realiza en el Capítulo 3. En el Capítulo 4 se presenta la primer propuesta de mejora en la calidad de los voces producidas con HMM, mediante una colección de post-filtros basados en LSTM. En el Capítulo 5 se describe la primera extensión de la propuesta, con la combinación de dos tipos de filtros para atacar los problemas de la síntesis de voz en dos etapas. En el Capítulo 6 se encuentra la segunda extensión de la propuesta, con un sistema discriminativo que entrena y aplica los post-filtros de acuerdo con la naturaleza del segmento de la frase.

La segunda parte utiliza los procedimientos de la primera y los extiende a nuevas aplicaciones. La descripción general de esta segunda parte se realiza en el Capítulo 7. En el Capítulo 8 se muestran la aplicación de cambio de acento de una voz sintetizada, en el Capítulo 9 la mejora de señales de voz con ruido con redes LSTM, y en el Capítulo 10 una extensión de éste utilizando un sistema híbrido. Finalmente, en el Capítulo 11 se presentan las conclusiones de toda la tesis y las perspectivas de investigación futura.

# 2

# Modelos de memoria de corto y largo plazo (*Long Short-term Memory*, LSTM)

*En este capítulo se realiza una descripción del bloque principal adoptado como unidad en las redes neuronales a lo largo de la tesis, el cual permite almacenar valores y modelar más adecuadamente la dinámica propia del habla.*

## Índice

## 2.1 Introducción

En este capítulo se realiza una descripción del bloque principal utilizado como unidad en las redes neuronales a lo largo de la tesis. Este bloque recibe el nombre de LSTM (acrónimo de *Long Short-term Memory*), o modelo de memoria a corto y largo plazo.

Existen varias razones para su utilización en todos aquellos problemas donde exista información secuencial [30]. Una de ellas es la analogía con el procesamiento de información en el cerebro humano, en el cual la comprensión de los mensajes que llegan a través de los sentidos se apoya en información previa. Por ejemplo en la lectura, donde cada palabra toma sentido de acuerdo con las anteriores, conformando así frases e ideas. Es decir, existe una persistencia de información que permite comprenderla.

Las redes neuronales tradicionales (tales como el perceptrón) no pueden procesar la información de esta manera, que requiere tomar en cuenta estados inmediatos anteriores u otros ubicados aún más atrás en el tiempo. Con la creación de las redes neuronales artificiales recurrentes, que contienen conexiones de las unidades internas hacia sí mismas, surge la posibilidad de utilizar información de momentos anteriores en el instante presente.

En las siguientes secciones se describen las LSTM como una extensión de las redes recurrentes, que permiten almacenar la información a corto o largo plazo.

## 2.2 Redes Recurrentes

Las redes neuronales profundas (compuestas por unidades organizadas en múltiples capas) muestran problemas relacionados con el desvanecimiento en la propagación del error durante el proceso de entrenamiento. Este problema, conocido como “gradiente descendiente” se produce por las dificultades de propagar el valor de error que se calcula a partir del gradiente de una función de error con respecto a los pesos de la red [31].

Este gradiente se determina con el fin de minimizar el error mediante la búsqueda de un punto donde el gradiente se anule, es decir, alcance un máximo (o un mínimo). Los pesos de la red se actualizan a partir de ajustes calculados a partir del gradiente en la última capa, y en las demás se va propagando a partir del error de la capa inmediata anterior. En la Figura 2.1 se ilustra este procedimiento, donde cada iteración ajusta los parámetros $\theta_0, \theta_1$ para que se alcance el máximo de la función objetivo.

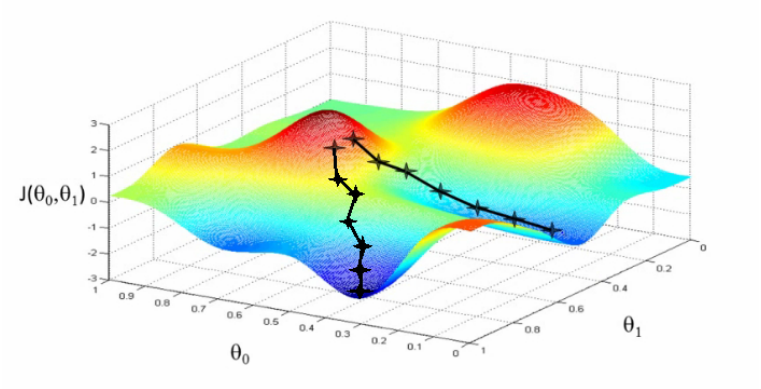



**Figura 2.1:** Ilustración del gradiente descendiente para los parámetros $\theta_0, \theta_1$. Tomado de [2]

El problema del gradiente descendiente al incluir muchas capas ocultas en las redes neuronales, además de las dificultades inherentes al no poder utilizar valores previos de las entradas para decidir sobre las salidas actuales, han hecho a grupos de investigadores pensar en incorporar capacidades que permitan manejar información secuencial. Este tipo de información es deseable en aplicaciones de habla, en las cuales hay una dependencia de la información previa en las características del sonido actual, tanto para el reconocimiento como para la síntesis. Algunas de las técnicas propuestas para trabajar con habla considerando esta naturaleza dinámica, han incluido las Redes Neuronales Recurrentes (*Recurrent Neural Networks*, RNN) [32]. La principal característica de esta variante es la retroalimentación de algunas o todas las neuronas en una capa, permitiendo de esta manera almacenar valores previos, como se ilustra en la Figura 2.2.

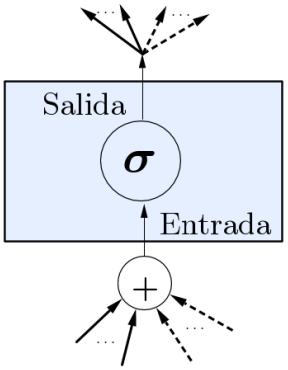



**Figura 2.2:** Unidad una red recurrente.

Un nuevo tipo de RNN, cuyo uso se ha extendido recientemente en diversas aplicaciones, ha sido propuesta en [30], y ha sido llamada *Long Short-term Memory* (LSTM), o Modelo de Memoria a Corto y Largo Plazo. Entre los alcances logrados en el área de reconocimiento de

habla [33, 34], se encuentra la menor tasa de error en reconocimiento de habla lograda hasta el momento con la base de datos TIMIT [35].

La característica principal de las LSTM es que, en lugar de una conexión de recurrencia simple hacia las neuronas de una capa, se tiene un bloque y una seria de compuertas para acceder, almacenar o propagar la información a través de él. De esta manera, los valores internos de la red se pueden almacenar por periodos cortos o largos de tiempo, dependiendo del accionar de las compuertas.

## 2.3 Bloque de memoria

En una RNN, la secuencia de vectores de salida $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ se calculan de las secuencia de entrada $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ y de las secuencias presentes en los estados internos $\mathbf{h} = (h_1, h_2, \ldots, h_T)$, iterando las siguientes ecuaciones para las entradas con índice 1 hasta $T$ [36]:

$$h_t = \sigma\left(\mathbf{W}_{xh}x_t + \mathbf{W}_{hh}h_{t-1} + b_h\right) \tag{2.1}$$

$$y_t = \mathbf{W}_{hy}h_t + b_y \tag{2.2}$$

donde $\mathbf{W}$ es la matriz de pesos. Por ejemplo, $\mathbf{W}_{xh}$ es la matriz de pesos entre las entradas y la capa oculta $h$, $b_h$ es el vector de sesgo en la capa $h$ (tal como en redes neuronales tipo perceptrón), y $\sigma$ es la función de activación en los nodos de las capas ocultas, usualmente la función sigmoide $f : \mathbb{R} \to \mathbb{R}$, $f(t) = \frac{1}{1+e^{-t}}$.

En las redes LSTM, en lugar de una función sigmoide con conexiones recurrentes, el bloque de memoria almacena los valores o los propaga a través de su sistema de compuertas. Este bloque puede considerarse que está constituido por la unidad de memoria, las compuertas y las conexiones entre ellos.

Existen varias implementaciones de bloques de memoria LSTM, con cantidades variables de compuertas, unidades de memoria por bloque y recurrencias. En la presente tesis se ha utilizado un tipo de LSTM con una sola unidad de memoria por bloque y tres compuertas, descritas a continuación:

- Entrada: Controla si los valores que provienen de la capa anterior y de las recurrencias podrán ingresar al bloque.

- Olvidar: Permite borrar el valor almacenado en memoria, por medio de una multiplicación con el valor a la salida de la activación de la compuerta.
- Salida: Controla si el valor almacenado en memoria se propaga hacia las siguientes capas y recurrencias del bloque de memoria.

Con estas compuertas se realiza el control del acceso, actualización y salida de la unidad de memoria, de manera que los valores puedan ser almacenados en el corto o largo plazo, y estar disponibles para influir en las salidas, aún muchos instantes después de que hayan ingresado a la red. Las compuertas reciben valores de recurrencia de todas las otras unidades en la misma capa, y de las salidas de la capa anterior. Una unidad de memoria se ilustra en la Figura 2.3.

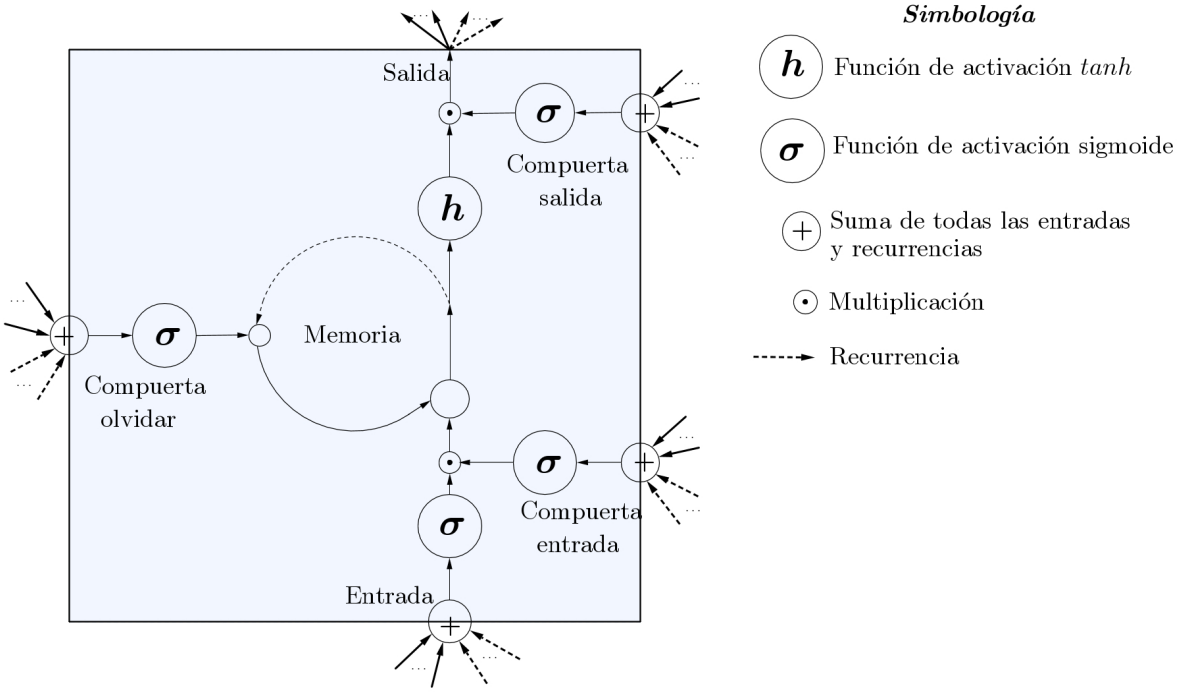



**Figura 2.3:** Unidad de memoria LSTM.

Las compuertas se implementan utilizando las siguientes ecuaciones, escritas en notación matricial:

$$i_t = \sigma\left(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i\right) \tag{2.3}$$

$$f_t = \sigma\left(\mathbf{W}_{xf}\vec{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f\right) \tag{2.4}$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c\right) \tag{2.5}$$

$$o_t = \sigma\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o\right) \tag{2.6}$$

$$h_t = o_t \tanh\left(c_t\right) \tag{2.7}$$

donde $\sigma$ es la función sigmoide, $\mathbf{x}$ es el vector de entrada, $f$ es la función de activación de la compuerta de olvidar, $o$ es la función de activación de la salida y $c$ es la función de activación de la memoria. $\mathbf{W}_{mn}$ representa las matrices de pesos de cada celda al vector de compuertas. $h$ es la salida de la unidad de memoria.

## 2.4 Entrenamiento

Una red LSTM consiste en una capa de entradas y otra de salidas, tal como otros modelos de redes neuronales, pero a diferencia de éstas, en las capas ocultas se ubican los bloques de memoria. La combinación de compuertas permite a una red LSTM decidir cuándo mantener o borrar la información en la unidad de memoria, acceder a su valor o prevenir a otras unidades ser afectadas por su valor actual [30]. La mayoría de aplicaciones prácticas requiere de una cantidad mayor de capas y de unidades por capa.

Tal como se realiza en las redes RNN, el entrenamiento consiste en un procedimiento hacia adelante y otro hacia atrás de la red, para ajustar los pesos $w_{ij}$ entre las entradas, compuertas, salidas y recurrencias. El paso hacia adelante inicia en $t = 1$, y recursivamente aplica las siguientes ecuaciones, mientras se incrementa $t$ desde 1 hasta $T$ (la cantidad de entradas) [37]:

- Ecuaciones de la compuerta de entrada: Los valores se obtienen a partir de los pesos y los valores provenientes de las entradas, las recurrencias y el sesgo, como se muestra en las ecuaciones

$$i_\iota^t = \sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1} + \sum_{c=1}^{C} w_{c\iota} s_c^{t-1} \tag{2.8}$$

$$b_\iota^t = \sigma\left(i_\iota^t\right) \tag{2.9}$$

- Ecuaciones de la compuerta olvidar: Como en el caso anterior, se determina su valor a partir de las entradas, recurrencias, sesgo y los pesos correspondientes.

$$a_{\phi}^{t} = \sum_{i=1}^{I} w_{i\phi} x_{i}^{t} + \sum_{h=1}^{H} w_{h\phi} b_{h}^{t-1} + \sum_{c=1}^{C} w_{c\phi} s_{c}^{t-1} \tag{2.10}$$

$$b_{\phi}^{t} = \sigma\left(a_{\phi}^{t}\right) \tag{2.11}$$

- Ecuaciones del valor de memoria: El nuevo valor de memoria se calcula con las entradas y el sesgo, y su actualización requiere el valor anterior.

$$a_{c}^{t} = \sum_{i=1}^{I} w_{ic} x_{i}^{t} + \sum_{h=1}^{H} w_{hc} b_{h}^{t-1} \tag{2.12}$$

$$s_{c}^{t} = b_{\phi}^{t} s_{c}^{t-1} + b_{\iota}^{t} \sigma\left(a_{c}^{t}\right) \tag{2.13}$$

- Ecuaciones de la compuerta de salida: El valor de activación se determina con recurrencias de las entradas, sesgo y el valor de memoria.

$$a_{w}^{t} = \sum_{i=1}^{I} w_{i\omega} x_{i}^{t} + \sum_{h=1}^{H} w_{h\omega} b_{h}^{t-1} + \sum_{c=1}^{C} w_{c\omega} s_{c}^{t} \tag{2.14}$$

$$b_{\omega}^{t} = \sigma\left(a_{\omega}^{t}\right) \tag{2.15}$$

- Ecuación del valor de salida:

$$b_{c}^{t} = b_{\omega}^{t} h(s_{c}^{t}) \tag{2.16}$$

donde $s$ es la salida del valor de memoria, $a_{j}^{t}$ es la entrada $j$ de la unidad en el tiempo $t$, $b_{j}^{t}$ es la activación de la unidad $j$ en el tiempo $t$, y los subíndices $c$, $\iota$, $\phi$ y $\omega$ se refieren al valor de memoria, la compuerta de entrada, la compuerta de olvidar y la compuerta de salida del bloque. $I$, $K$ y $H$ son la cantidad de entradas, salidas y celdas en la capa oculta, respectivamente.

El paso hacia atrás inicia en $t = T$ y de forma recursiva se calculan las siguientes ecuaciones de actualización de los pesos, mientras $t$ decrece hasta uno.

- Ecuación de las salidas: $\forall c \in C$, se define

$$\varepsilon_c^t = \sum_{k=1}^{K} w_{ck}\delta_k^t + \sum_{g=1}^{G} w_{cg}\delta_g^{t+1} \tag{2.17}$$

- Ecuación de las compuertas de salida:

$$\delta_w^t = \sigma'(a_w^t) \sum_{c=1}^{C} h(s_c^t \varepsilon_c^t) \tag{2.18}$$

- Ecuaciones de las memorias:

$$\varepsilon_s^t = b_\omega^t h'\left(s_c^t\right) \varepsilon_c^t + b_\phi^{t+1}\varepsilon_s^{t+1} + w_{c\imath}\delta_\imath^{t+1} + w_{c\phi}\delta_\phi^{t+1} + w_{c\omega}\delta_\omega^t \tag{2.19}$$

$$\delta_c^t = b_\imath^t \tanh'\left(a_c^t\right) \varepsilon_s^t \tag{2.20}$$

- Ecuación de las compuertas olvidar:

$$\delta_\phi^t = \sigma'\left(a_\phi^t\right) \sum_{c=1}^{C} s_c^{t-1}\varepsilon_s^t \tag{2.21}$$

- Ecuación de las entradas:

$$\delta_\imath^t = \sigma'\left(a_\imath^t\right) \sum_{c=1}^{C} g\left(a_c^t\right) \varepsilon_s^t \tag{2.22}$$

donde $\delta_c^t = \frac{\partial \Gamma}{\partial b_c^t}$, $\delta_s^t = \frac{\partial \Gamma}{\partial s_c^t}$. $\Gamma$ es la función de error, tal como el cuadrado de la distancia euclidiana.

El proceso de entrenamiento supervisado se puede describir de la siguiente manera:

1. En cada instante de tiempo, los valores son presentados en las entradas, mientras que los correspondientes a las clases o los valores deseados se presentan a la salida.
2. Las compuertas se activan o permanecen cerradas de acuerdo con las ecuaciones 2.8 a 2.16, aplicadas en ese orden.
3. Los valores de memoria se actualizan, borran o permanecen para el siguiente instante de tiempo, de acuerdo con las ecuaciones mencionadas y la combinación respectiva de compuertas.

4. Cuando se propagan los valores hasta la salida, se calcula un error con la diferencia entre la salida producida a partir de la entrada y los valores deseados.
5. Después de calcular el error, las ecuaciones 2.17 a 2.22 se aplican en ese orden para actualizar los pesos de la red.

Este es un proceso iterativo, donde intervienen criterios como el número de iteraciones o el error obtenido para decidir sobre la actualización de los pesos. Dada la cantidad de cálculos requeridos en cada paso se hace necesario utilizar sistemas de aceleración de hardware (como GPU) para la mayoría de aplicaciones prácticas.

## 2.5 Validación y prueba

En aplicaciones prácticas, los procesos de validación y prueba se realizan de forma semejante a otros modelos de redes neuronales. En primer lugar, se realiza una división del conjunto de datos disponible en tres: entrenamiento, validación y prueba. El proceso de entrenamiento sigue el procedimiento descrito en la sección anterior.

Para determinar si se almacenan como los mejores valores de los pesos los del paso actual, se calcula el error que se produce en el conjunto de validación. Si este valor de error no es mejor que el anterior, los pesos de la red no se actualizan. Finalmente, para reportar resultados se utiliza la red posterior al proceso de entrenamiento con el conjunto de prueba. Como criterios de paro del proceso de entrenamiento se pueden utilizar los mismos de otros modelos de red neuronal: cantidad de épocas o umbral de error. Una época es la propagación de valores hacia adelante de la red, y la actualización de pesos correspondiente de acuerdo con el error obtenido en la salida.

# Parte I

## Post-filtros basados en algoritmos de aprendizaje profundo para la mejora de la síntesis de voz basada en HMM

# 3

# Introducción a la primera parte

*En este capítulo se describe el problema de utilización de post-filtros en la síntesis de voz basada en HMM, lo cual introduce las tres propuestas realizadas en los capítulos siguientes. Se muestra la revisión del estado del arte en el tema y las arquitecturas de red utilizadas.*

## Índice

## 3.1 Introducción

En este capítulo se introducen los post-filtros basados en algoritmos de aprendizaje profundo en la mejora de la síntesis estadística paramétrica de voz generada en HMM. Esto se realiza considerando no solamente un post-filtro para un conjunto de los parámetros del habla, como se ha realizado previamente en la literatura del tema, sino un conjunto de éstos para mejorar la totalidad de los parámetros del habla, con una parametrización que considera $f_0$, coeficientes espectrales en la escala de Mel (*Mel Frequency Cepstral Coefficients*, MFCC) y energía.

Para considerar todos estos parámetros, se propone el uso de una colección de redes de distintos tipos, tanto autocodificadores (*autoencoders*) como memorias auto-asociativas. El objetivo principal de esta propuesta es determinar si este sistema basado en un conjunto de redes supera a aquellos basados solamente en una. Por otra parte, se desea determinar la conveniencia de utilizar redes basadas en unidades LSTM como post-filtros.

El capítulo está organizado de la siguiente manera: En la Sección 3.1.1 se realiza una descripción del sistema propuesto, en la Sección 3.1.2 se describe el trabajo relacionado. En la Sección 3.2 se presenta el sistema *H*TS-Parallel, con el cual se produce el alineamiento del habla natural y sintetizada. Finalmente, en la Sección 3.3 se describe el procedimiento general de los post-filtros y las arquitecturas de red utilizadas en el resto de la tesis.

### 3.1.1 Planteamiento del problema

El habla natural contiene características de variabilidad muy diversas, tanto en intensidad, duración y variaciones en los sonidos. Esto se refleja en cualquier parametrización realizada sobre la misma. Se pueden observar, por ejemplo, variaciones en frecuencia fundamental a lo largo de las frases en distintas emisiones de la misma palabra u oración aún siendo pronunciadas por el mismo hablante. Por otra parte, cuando se realiza síntesis de voz basada en HMM, los parámetros y características del habla son suavizadas y pierden variabilidad, debido al modelado estadístico y promediado realizado para hacer posible el entrenamiento de los HMM [38].

La posibilidad de mejorar los resultados de la voz sintetizada utilizando algoritmos de aprendizaje profundo, surge recientemente como una etapa posterior a la generación de parámetros, de manera que se pueda enriquecer la variabilidad de los mismos y reflejar de forma más cercana el habla natural. Esto sistemas son llamados post-filtros [39].

Para esto se pueden considerar los parámetros de habla sintetizada $R_Y$ como una versión ruidosa o degradada de la voz natural $R_X$. Si un conjunto de frases de habla sintetizada se encuentran alineadas con las de habla natural, cada ventana de habla natural y sintetizada se puede parametrizar, resultando en un vector

$$\mathbf{c} = [c_1, c_2, \ldots, c_M] \tag{3.1}$$

donde $M$ es el número de coeficientes extraídos. De esta manera, cada frase completa produce una matriz de tamaño $M \times T$, de la forma

$$\mathbf{R} = [\vec{c}_1^{\top}, \vec{c}_2^{\top}, \ldots, \vec{c}_T^{\top}]. \tag{3.2}$$

Siguiendo esta notación, sean $\mathbf{R_Y}$ y $\mathbf{R_X}$ matrices de habla sintética y natural, respectivamente, y $\vec{R_W}$ la matriz resultante de la concatenación de ambas.

En el proceso de implementar algoritmos de aprendizaje profundo a la manera de post-filtros, los parámetros de la voz sintetizada con HMM pueden mejorarse estimando una función $f$ directamente de los datos, con la cual se puedan mapear los parámetros sintéticos hacia los naturales, usando, por ejemplo, redes neuronales recurrentes. La función objetivo con la cual se puede realizar el procedimiento de estimación de parámetros del post-filtro se puede plantear [40]:

$$E(\mathbf{R_W}) = \|f(\mathbf{R_Y}; \mathbf{R_W}) - \mathbf{R_X}\|^2 \tag{3.3}$$

En la presente tesis, se propone el uso de redes LSTM para realizar el proceso de estimar la función $f$, o el conjunto de funciones $f_1, \ldots, f_k$ que mapean los parámetros de la voz basada en HMM hacia los del habla natural. En cada uno de los capítulos de la Parte I se describe la manera particular de entrenar los post-filtros y de utilizarlos para mejorar el habla artificial.

### 3.1.2 Trabajo relacionado

Después de las numerosas implementación de la síntesis de voz basada en HMM en diversos idiomas, realizadas posterior a la difusión del sistema de creación de voces basado en HMM (*HMM-based Speech Synthesis System*, HTS) en el año 2002 (y especialmente a partir del año 2009 donde se publican ejemplos adaptables a otros idiomas), han nacido algunas líneas de investigación dedicadas a mejorar los resultados. Por ejemplo, en [39] se propuso un post-filtro basado en el algoritmo de Modulación del espectro (MS) para mejorar los parámetros

relacionados con estas componentes de la señal. La idea de este algoritmo es acercar los parámetros espectrales a los naturales. Un enfoque similar fue presentado también en [41] [42] utilizando redes de creencia profunda, mostrando que los resultados no solamente son aplicables para mejorar voces basadas en HMM con el sistema HTS, sino con otras técnica paramétricas como las producidas a partir de mezclas de distribuciones gaussianas (*Gaussian Mixture Models*, GMM) y el software CLUSTERGEN [43]. En [44], los parámetros espectrales del habla sintética fueron mejorados utilizando Redes de Creencia Profunda (*Deep Belief Networks*, DBN), mientras que $f_0$ fue mejorado utilizando una trasformación lineal a partir de la media y la desviación estándar. La energía de la señal original fue usada en el habla final sin ningún cambio.

Los post-filtros en cascada para mejorar el espectro fueron propuestos en [42]. Estos post-filtros funcionan para mejorar los MFCC del habla sintetizada en dos etapas, combinando Máquinas restringidas de Boltzmann (*Restricted Boltzmann Machines*, RBM) y memorias bidireccionales asociativas (*Bidirectional Associative Memory*, BAM). La verificación de la mejora se realiza evaluando los picos y valles del espectro con respecto al habla original.

Más recientemente, el uso de redes neuronales recurrentes fue presentado en [45], en contraste con otras redes como las DBN presentadas previamente. De acuerdo con este estudio, la estructura y característica inherente de las redes recurrentes favorece la mejora del habla, ya que sus parámetros son por naturaleza dependientes del tiempo, lo cual fue propuesto también en [41].

A partir de estas referencias, se nota un abierto interés en mejorar la señal de habla producida por sistemas estadísticos paramétricos, como los basados en HMM. Las dos limitaciones principales que se encuentran para los procedimientos aplicados en la literatura surgen de dos aspectos:

- Dado que el habla sintetizada con modelos estadísticos no se encuentra alineada con el habla natural, no se pueden aplicar medidas de calidad propias de la mejora en las señales de habla. La mayoría de las referencias de post-filtros han recurrido a técnicas como Alineamiento temporal dinámico (*Dynamic Time Warping*, DTW) para realizar una correspondencia y mejora entre los segmentos del habla natural y artificial.

- Las referencias se han concentrado en la mejora de la calidad del espectro del habla sintetizado. Existen limitaciones en los modelos aplicados hasta el momento para mejorar el parámetro de $f_0$, por su naturaleza en parte binaria (segmentos con y sin $f_0$) y con valores reales en los segmentos sonoros, los cuales tienen un gran impacto en la percepción de calidad del resultado.

Con respecto al primer problema, la alineación del habla natural y sintetizada, en la siguiente sección se resume el sistema desarrollado para este fin en la presente tesis.

## 3.2 Alineamiento de habla natural y sintetizada

Dado un HMM $\lambda$ y una cantidad $T$ de ventanas que se desean generar a partir de éste, en la generación de parámetros se desea maximizar la secuencia de estados en el HMM para producirlas [46], es decir:

$$\log P(\mathbf{q}|\lambda, \mathbf{T}) = \sum_{k=1}^{K} \log p_k(d_k) \tag{3.4}$$

bajo la restricción

$$T = \sum_{k=1}^{K} d_k \tag{3.5}$$

donde $d_k$ es la duración del estado $k$, $p_k(d_k)$ es su probabilidad, y $K$ es el número de estados en el HMM $\lambda$. $p_k(d_k)$ se modela con una distribución gaussiana. Por esta razón, las duraciones que maximizan la ecuación 3.4 están dadas por

$$d_k = \xi(k) + \rho \cdot \sigma^2(k) \tag{3.6}$$

donde

$$\rho = \frac{\left(T - \sum_{k=1}^{K} \xi(k)\right)}{\sum_{k=1}^{K} \sigma^2(k)}. \tag{3.7}$$

$\xi(k)$ es la media y $\sigma^2(k)$ la varianza de la gaussiana que modela la duración del estado $k$. Dada la asociación de $\rho$ con $T$, la duración de los fonemas en el habla sintetizada a partir de HMM puede ser controlada por este parámetro. Por ejemplo, si $\rho = 0$, la duración del habla sintetizada será la media de la distribución correspondiente, pues resulta $T = \sum_{k=1}^{K} \xi(k)$. Para el caso $\rho < 0$ se tiene que el habla es más rápida, mientras que si $\rho > 0$ el habla es más lenta. De esta manera se puede variar la velocidad del habla en la síntesis basada en HMM.

Durante el proceso de entrenamiento del conjunto de HMM necesarios para generar una voz, se requiere de una segmentación de las frases que conforman la base de datos. Este proceso consiste en determinar las fronteras temporales en las que se encuentran los fonemas en los audios, para

realizar una identificación de los parámetros extraídos con la información lingüística de la frase.

En la Figura 3.1 se muestra un ejemplo del resultado del proceso de segmentación, en el cual se identifican con etiquetas temporales el inicio y el fin del fonema particular en una frase.

```
        0    2500000 #
  2500000    3500000 k
  3500000    4300000 o
  4300000    5100000 n
  5100000    5900000 e
  5900000    7000000 s
  7000000    7850000 t
  7850000    8930000 o1
  8930000    9410000 i0
  9410000   10410000 k
 10410000   11210001 o
```

**Figura 3.1:** Ejemplo de etiquetas temporales para una secuencia de fonemas. Las columnas de tiempo están dadas en nano segundos. La tercer columna indica los fonemas de la frase “con estoico”.

En el proceso de sintetizar una nueva frase utilizando los HMM ya entrenados, lo cual se esquematiza en la Figura 3.2, se utilizan los modelos de duración estimados a partir de los ejemplos de la base de datos, y se puede manipular el parámetro $\rho$ para hacer el habla más rápida o más lenta que la media de los valores.

En el sistema propuesto en la presente tesis se desarrolló una variante del sistema HTS, en el cual se genera una réplica de las frases utilizadas en el entrenamiento, pero en su versión sintetizadas. Este sistema se llama *HTS-Parallel*.

La forma de generar el habla en *HTS-Parallel* parte de la sustitución de los modelos de duración establecidos como distribuciones gaussianas, por las duraciones específicas obtenidas en el proceso de segmentación durante el entrenamiento. Con la duración específica de cada fonema, el software *hts engine* ajusta el valor de $\rho$ para generar la cantidad de vectores de parámetros necesarias para el tiempo especificado.

De esta manera *HTS-Parallel* opera eliminando los modelos de duración y sustituyéndolos por la información proveniente de la segmentación, como se ilustra en la Figura 3.3.

El objetivo de este software es elaborar una base de datos que consiste en frases de habla naturales (provenientes de la base de datos) y generadas artificialmente con el HTS, las cuales tengan una correspondencia ventana a ventana. Con esta correspondencia se puede plantear un mapeo directo entre parámetros del habla sintetizada y natural, sin recurrir a técnicas como DTW. El problema de utilizar post-filtros se simplifica en principio, gracias a la correspondencia directa de los datos.

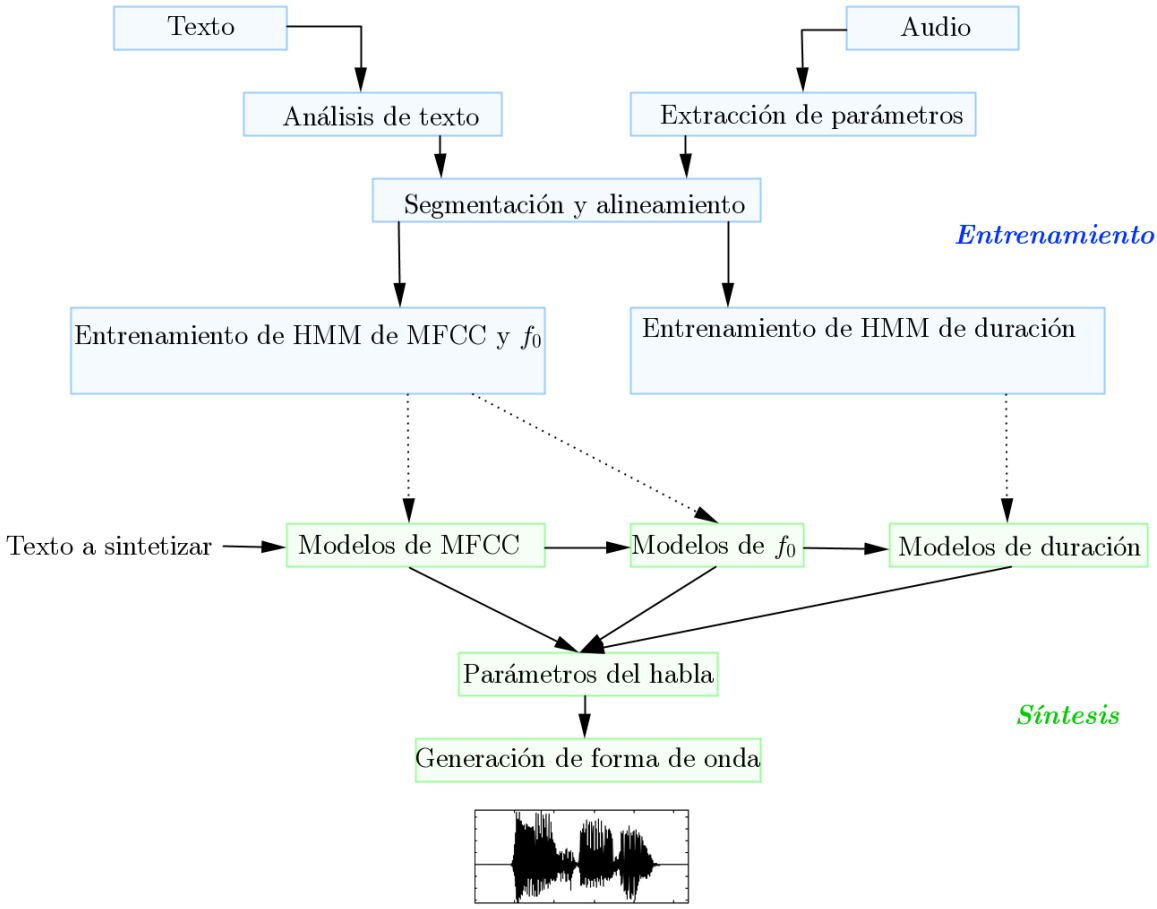



**Figura 3.2:** Esquema de generación de una nueva frase en síntesis basada en HMM.

En la Figura 3.4 se muestra un ejemplo del efecto de alineamiento que produce el software desarrollado en comparación con HTS. *H*TS-Parallel se encuentra disponible en la dirección `https://github.com/mcoto/HTS-ParallelTraining`.

## 3.3 Mejora de señales de voz con redes neuronales profundas

La idea de entrenar con el algoritmo de retropropagación una red neuronal tipo perceptrón con múltiples capas, con el fin de eliminar ruido o distorsiones de su entrada, data de la década de 1980. Investigadores como LeCun y Gallinari realizaron pruebas en redes con entradas binarias, en las cuales una fracción de éstas cambiaba su valor, para que la red aprendiera a eliminar estos

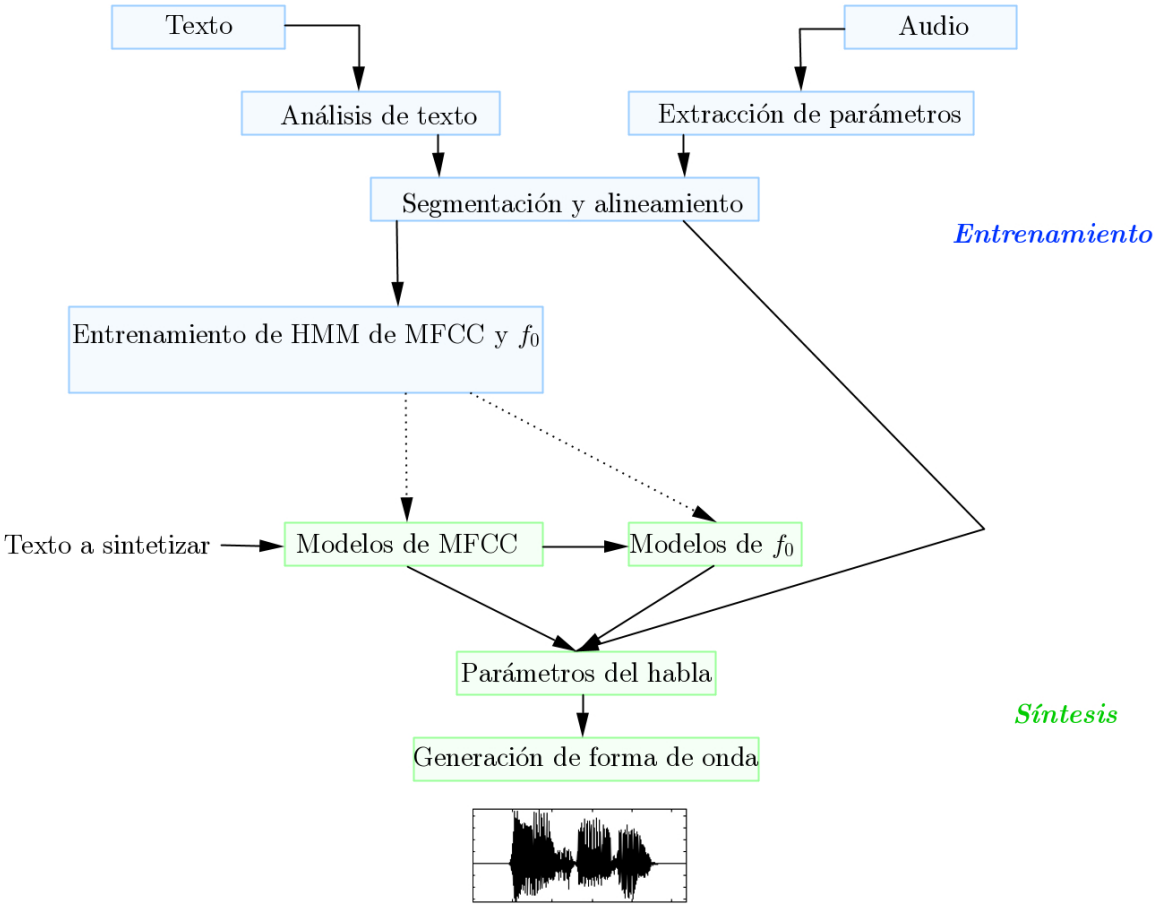
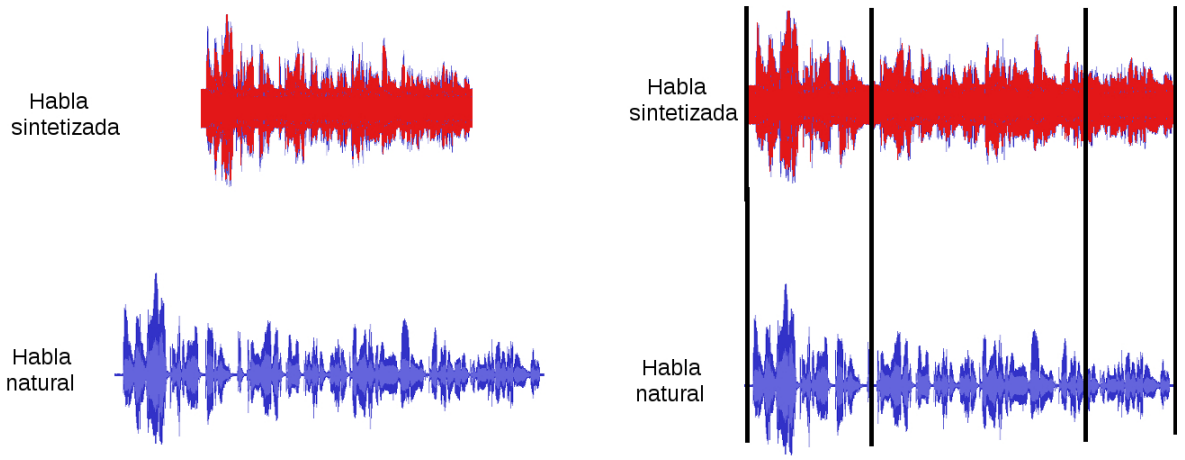



**Figura 3.3:** Esquema de generación de una nueva frase en HTS-Parallel.




(a) Oscilogramas de frases producidas con HTS

(b) Oscilogramas de frases producidas con *HTS-Parallel*

**Figura 3.4:** Comparación de oscilogramas de una frase generada con HTS (no alineada) y con *HTS-Parallel* (alineada).

cambios [47]. Este procedimiento guarda semejanzas con la reconstrucción de parámetros de habla como el $f_0$ que se degradan en presencia de ruido, aunque se trata en este caso de una señal más compleja.

Para el caso de parámetros no binarios, en una época semejante a la experiencia mencionada anteriormente, la predicción y mejora de éstos con redes neuronales de una sola capa tuvo un éxito moderado [48]. En esos momentos, sin embargo, ni la capacidad del hardware ni los algoritmos de aprendizaje eran adecuados para manejar redes con muchas capas ocultas o grandes cantidades de datos, por lo cual los beneficios no eran suficientemente buenos.

En la actualidad estas dificultades han sido superadas gracias a las nuevas tecnologías y modelos de redes neuronales, por lo que la tarea de mejorar señales de voz es accesible y se ha observado cómo es competitiva con otros algoritmos clásicos, en tareas como el reconocimiento automático del habla. El mapeo requerido entre el habla con ruido y sin ruido puede considerarse como un modelo de regresión, aprendido en pares correspondientes de datos limpios y ruidosos. En la etapa de mejoramiento, la red entrenada se utiliza para predecir los parámetros limpios a partir de los ruidosos que se presenten en su entrada [49].

La Figura 3.5 resume este proceso. Diversos algoritmos se han utilizado, algunas veces con variaciones de este esquema, como se describirá en la siguiente sección. Este mismo esquema puede ser aplicado al caso de habla sintetizada donde exista correspondencia ventana a ventana entre el habla sintetizada y la natural, tal como en el caso de eliminación de ruido.

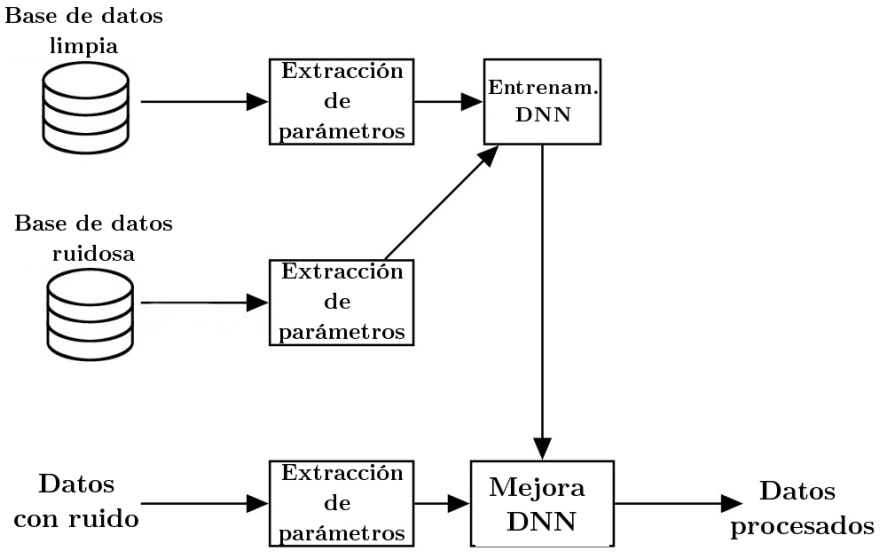



**Figura 3.5:** Aplicación típica de una red neuronal profunda para mejora de una señal de habla mediante reducción de ruido.

De igual forma que una red neuronal con entrenamiento supervisado, los parámetros de las redes

profundas se determinan de manera tal que se minimice el promedio de error de reconstrucción de la entrada. Es decir, que la función $f$ sea tal que $f(y)$ sea tan cercana como sea posible con la señal limpia $x$ [47]. Esta función, en la mayoría de aplicaciones prácticas, es desconocida. [50].

En las siguientes subsecciones se describen dos arquitecturas de redes que han sido utilizadas con éxito en problemas de mejoramiento de distintos tipos de señales, las cuales forman parte fundamental de las propuestas realizadas en la presente tesis.

### 3.3.1 *Autoencoders*

En cuanto a su arquitectura, un autocodificador (*autoencoder*) es simplemente una red neuronal que consiste en una capa de entrada, una de salida (ambos con igual número de unidades) y una o más capas ocultas. La diferencia principal con otras arquitecturas o aplicaciones de redes neuronales consiste en que la red es entrenada para reconstruir su entrada en la salida. Esto requiere que la cantidad de salidas coincida con la cantidad de entradas, y cuando alguna de las capas internas tiene una cantidad de unidades menor a la entrada, puede constituir una representación útil de los datos para tareas como compresión o reducción de dimensionalidad.

En el área de mejora de señales ruidosas, los *autoencoders* se han utilizado como algoritmos de eliminación de ruido, entrenados a partir de pares de señales limpias y ruidosas, como se muestra en la Figura 3.6. Dada la arquitectura de estas redes, si se tiene una capa oculta su funcionamiento se puede describir en dos etapas: una de codificación a partir de la entrada $\mathbf{y}$, la cual se puede describir como [51]

$$h(\mathbf{y}) = \sigma\left(\mathbf{W}_1\mathbf{y} + \mathbf{b}\right), \tag{3.8}$$

y una etapa de decodificación a partir de la representación interna, dada por

$$\hat{\mathbf{x}} = \mathbf{W}_2 h(\mathbf{y}) + \mathbf{c}, \tag{3.9}$$

donde $\mathbf{W}_1$ y $\mathbf{W}_2$ son las matrices con los pesos de la red neuronal entre la capa de entrada y la capa oculta, y entre la capa oculta y la de salida, respectivamente. $\mathbf{b}$ y $\mathbf{c}$ son vectores fijos de sesgo (*bias*) de las entradas y salidas. $\sigma$ es la función logística $\sigma(x) = (1 + exp{-x})^{-1}$. Los parámetros de la red se optimizan con la función objetivo

$$E(\theta) = \sum_i ||\mathbf{x_i} - \hat{\mathbf{x}}_\mathbf{i}||^2, \tag{3.10}$$

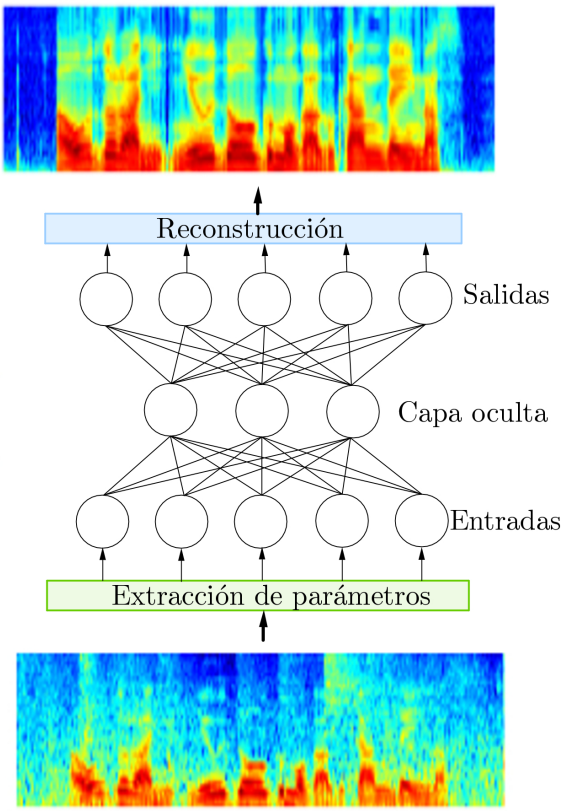
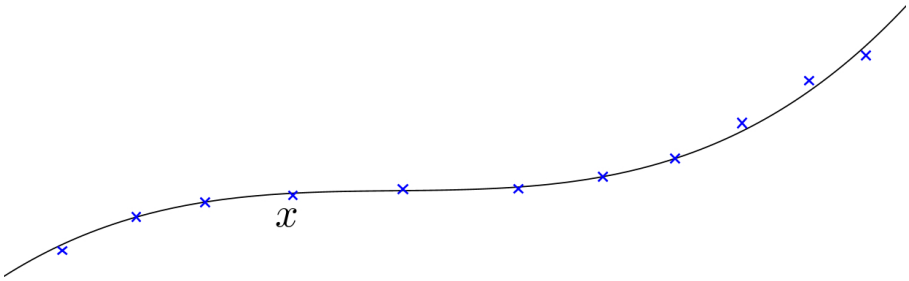



**Figura 3.6:** Ejemplo de *autoencoder* entrenado para eliminar ruido de una señal

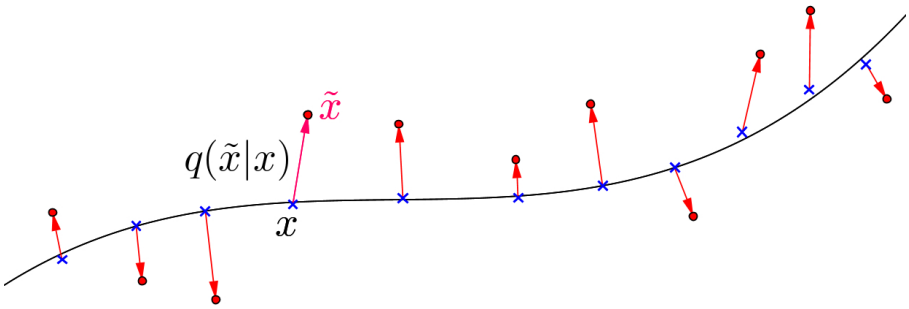
donde $\mathbf{x_i}$ es la versión limpia del habla correspondiente a la degradada $\mathbf{y_i}$, y $\theta = (\mathbf{W}, \mathbf{b}, \mathbf{c})$, son los parámetros de la red.

El proceso de eliminar ruido con un autoencoder se puede representar gráficamente como la aproximación de una variedad diferenciable [47], en la cual los puntos $x$ que representan parámetros limpios se encuentran cerca de la variedad, como se muestra en la Figura 3.7a. Cuando un proceso estocástico (como un ruido) afecta a la señal de habla, los puntos que se encontraban cerca de la variedad diferenciable son alejados de ésta para convertirse en parámetros $\tilde{x}$ ruidosos (Figura 3.7b). Finalmente, el *autoencoder* aprende la función $f(\tilde{x}|\theta)$ para mapear estos puntos de nuevo hacia la variedad diferencial(Figura 3.7c). La efectividad de este mapeo dependerá de la capacidad del *autoencoder*, de sus parámetros $\theta$ y de la cantidad de datos disponibles para estimarlos.

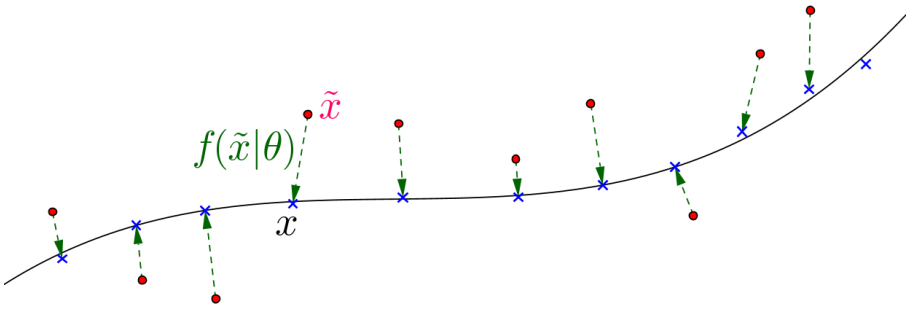
El éxito del proceso se puede representar como la capacidad del *a*utoencoder para acercar nuevamente los puntos hacia la variedad diferenciable.




**(a)** Representación gráfica de una variedad diferenciable cerca de la cual se encuentran los puntos $x$.




**(b)** Los puntos a los que se agrega ruido se alejan de la variedad mediante un proceso $q(\tilde{x}|x)$.




**(c)** El *autoencoder* aprende un mapeo $f$ de los puntos con ruido hacia la variedad diferenciable.

**Figura 3.7:** Representación gráfica del proceso de degradación por ruido y su posterior eliminación utilizando algoritmos de aprendizaje profundo.

### 3.3.2 Memorias auto-asociativas

Una memoria auto-asociativa es una red neuronal entrenada para aproximar la función identidad. Para esto, la arquitectura de la red requiere que la cantidad de entradas coincida con la cantidad de salidas, semejante al *autoencoder*. En el entrenamiento, los valores de las entradas son replicados en las salidas, de manera que los pesos de la red se ajustan a partir de estos ejemplos para producir la función deseada.

Su utilización como algoritmo para eliminar ruido se debe a que ha sido probada para restaurar errores en las entradas, cuando éstas han sido presentadas previamente en el entrenamiento. Su arquitectura puede ser semejante a la de un *autoencoder*, como el presentado previamente en la Figure 3.6.

En la presente tesis se plantea la utilización de una variante de memoria auto-asociativa para mejorar algunos de los parámetros del habla, tal como se describe en la Sección 4.2.

# 4

# Post-filtros basados en LSTM

*En este capítulo se desarrolla la primer propuesta de colecciones de post-filtros de múltiples arquitecturas basadas en LSTM para mejorar los resultados de la síntesis de voz basada en HMM.*

## Índice

## 4.1 Introducción

Las voces producidas con sistemas basados en parámetros, como en el sistema HTS, el cual utiliza HMM, tienen diferencias notables con las voces originales utilizadas para su generación, principalmente en términos de naturalidad. Existen comparaciones que las sitúan de forma cercana a una voz natural en inteligibilidad, cuando se cuenta con suficientes grabaciones de un hablante y éstas se realizan con características deseables, tales como la homogeneidad en la expresión y la tasa de habla.

En cuanto a la naturalidad, el proceso de entrenamiento de los HMM involucra el aprendizaje de parámetros mediante el cual los valores se promedian, de donde se pierden particularidades de la voz, además de los procesos de emisión de parámetros y reconstrucción que conllevan la degradación de la calidad de onda de habla. Previamente se ha propuesto la reducción de la diferencia entre habla sintetizada y natural a partir de algoritmos que aprendan la relación entre ambas directamente de los datos [42]. A diferencia de otras experiencias que utilizan la misma premisa, en nuestro caso partimos de frases de habla sintetizada que han sido alineadas con frases de habla natural, de manera que se puede establecer una correspondencia directa con los datos parámetro a parámetro extraído de ventanas del audio utilizando el sistema *H*TS-Parallel descrito en la Sección 3.2.

Para modelar la relación entre el habla sintetizada y natural, se debe tener en cuenta que en la producción de habla en el aparato fonador humano, existe una dependencia en la emisión de sonidos tanto en el corto plazo, por ejemplo en el fenómeno de coarticulación, como en el largo plazo con la prosodia. Experiencias previas realizadas con la finalidad de aprender esta dependencia con redes neuronales, incluyendo redes recurrentes, no han tenido el éxito deseado [37].

Las redes LSTM, descritas en el Capítulo 2, cuentan con unidades de memoria controladas con compuertas, las cuales permiten almacenar valores en el corto plazo o en el largo plazo. Esto las hace útiles en aplicaciones donde existan ambos tipos de dependencia entre los datos, como en la generación de letra manuscrita o en el reconocimiento de imágenes procesadas de forma secuencial [52].

En el proceso de mapeo entre parámetros de habla sintetizada y natural, esta característica constituye una ventaja con respecto a otras redes, por la mencionada dependencia del habla con los sonidos previos cercanos y lejanos. La propuesta de implementarlos en los post-filtros se muestra en la siguiente sección.

## 4.2 Sistema propuesto

El procedimiento inicia con la extracción de vectores de parámetros de habla natural y la correspondiente versión alineada de habla sintetizada, producida con *H*TS-Parallel. Para esto se definen ventanas de 10 ms, y se extraen los parámetros de $f_0$, 39 coeficientes MFCC y uno para energía, utilizando el sistema Ahocoder [53].

Para la mejora de estos conjuntos de parámetros, se contempla la utilización de dos tipos de redes profundas basadas en LSTM:

- Un *autoencoder* para los coeficientes MFCC. Este tipo de arquitectura de red ha sido probada previamente para la mejora de señales de voz degradadas con ecos, o con ruidos de distintos tipos y niveles como se mostró en la Sección 3.1.2.
- Memorias auto-asociativas para los coeficientes uni-dimensionales, como el $f_0$ y la energía. De acuerdo con los primeros experimentos realizados, la utilización de redes neuronales con una sola entrada y una salida no es suficiente para la aplicación de mejora de un único parámetro, como el $f_0$, aún en el caso de bloques LSTM. La ventaja de las memorias auto-asociativas con respecto a los *autoencoders* reside en la capacidad para mejorar este tipo de coeficientes cuando éstos se introducen en la red con parámetros coincidentes a la entrada y a la salida, aprendiendo en parte la función identidad como una memoria auto-asociativa tradicional, pero también la mejora del parámetro adicional correspondiente.

  De esta manera, se tiene una variante de memoria auto-asociativa, en la cual se entrena la función identidad para un número consecutivo de entradas $(y_2, \ldots, y_n)$, pero en una de ellas $(y_1)$ se realiza un mapeo hacia la versión natural de ésta $(x_1)$. En el proceso de prueba, la salida $\overline{x_1}$ se calcula a partir de las entradas $(y_1, \overline{x_2}, \overline{x_3}, \ldots, \overline{x_n})$.

El proceso de extracción de parámetros y su identificación como entradas y salidas a la red correspondiente se ilustra en las figuras 4.1 y 4.2. Dado que en ambos procesos existen coeficientes que permanecen sin mejorar, se pueden distinguir varios sistemas de post-filtros, de acuerdo con los parámetros contemplados en cada uno.

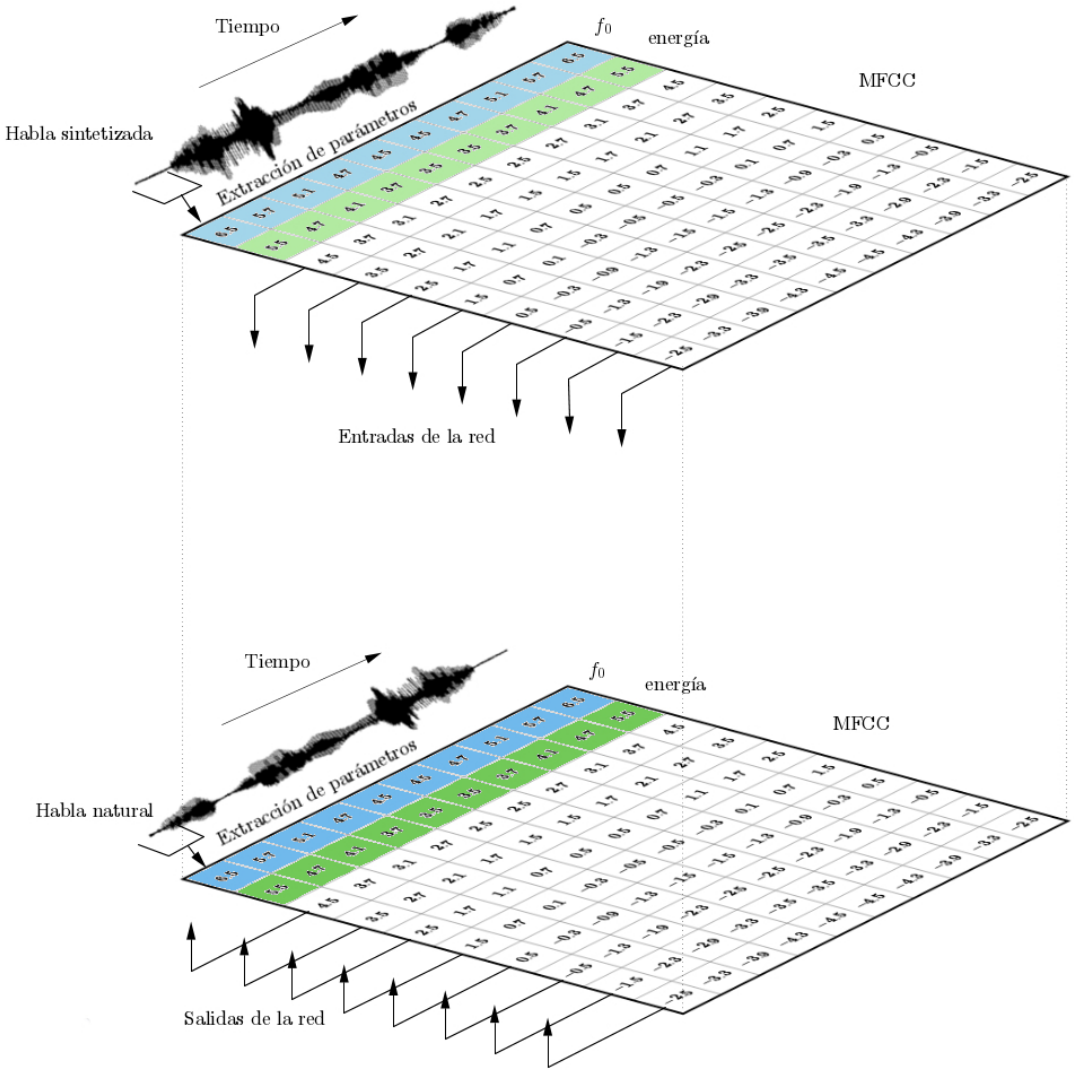



**Figura 4.1:** Ilustración del proceso de extracción de parámetros para el entrenamiento del *autoencoder* con MFCC del habla sintetizada y natural.

En el primer tipo propuesto (LSTM-1), los coeficientes MFCC se mejoran al extraer los parámetros del habla natural y sintetizada que se encuentran alineados. Durante el proceso de entrenamiento, al *autoencoder* correspondiente se presentan los parámetros en la entrada y la salida como se mostró en la Figura 4.1, y se describe con mayor detalle en el Algoritmo 1. En éste, el error se calcula con la suma de errores cuadráticos. En la etapa de mejora de nuevas frases, se utiliza el *autoencoder* entrenado para mejorar los MFCC, mientras que los coeficientes de $f_0$ y energía se conservan tal como se obtuvieron con la voz sintetizada HTS.

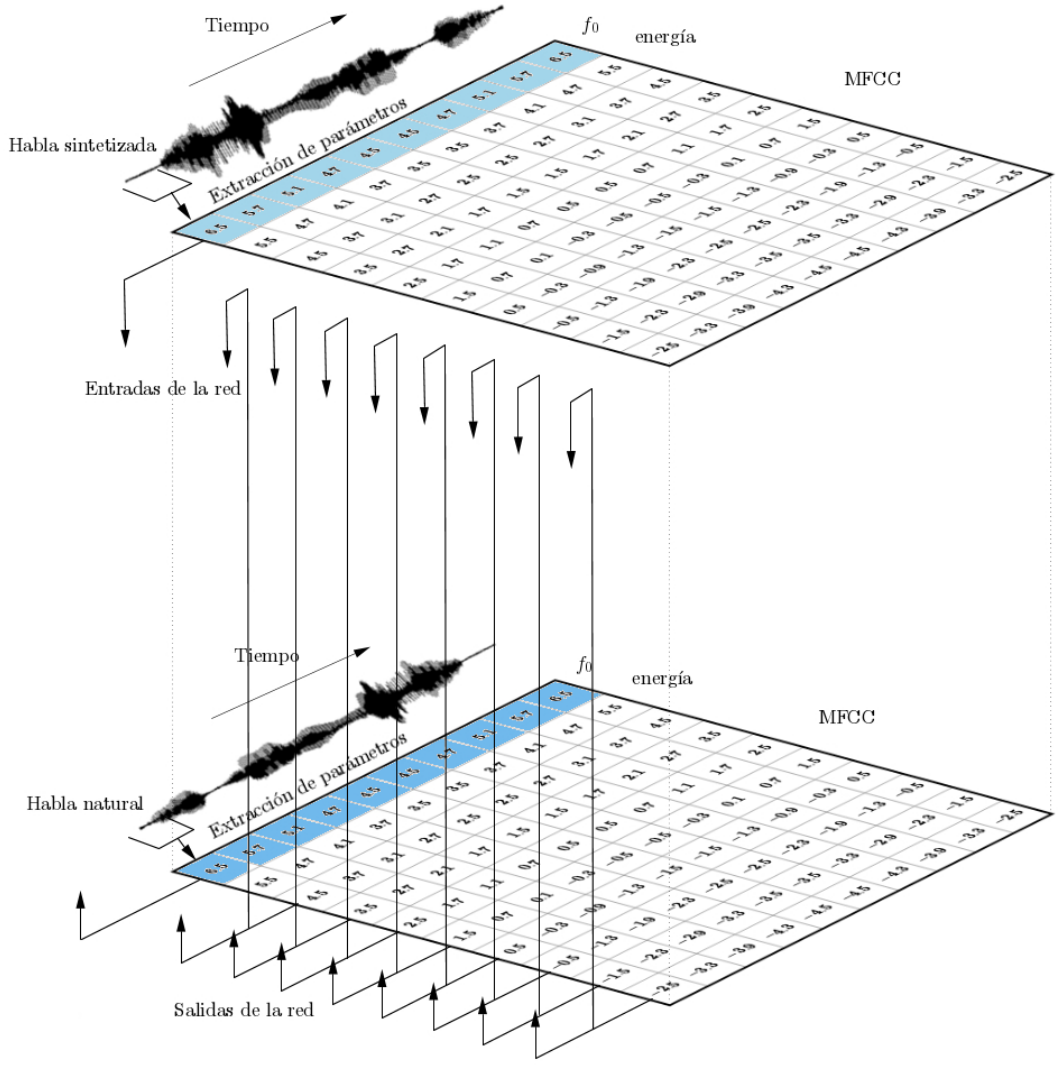



**Figura 4.2:** Ilustración del proceso de extracción de parámetros para el entrenamiento de la memoria auto-asociativa para la mejora del parámetro $f_0$ en el habla sintetizada.

**Algoritmo 1** Entrenamiento de red LSTM para mejorar coeficientes MFCC del habla sintetizada con HTS

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: Frases de habla HTS alineadas
**Entrada:** $L$: Red LSTM inicializada
**Entrada:** $N$: Número de iteraciones
**Entrada:** $K$: Número de ventanas
**Salida:** $L_{ae}$: Red LSTM entrenada para eliminar ruido en 39 MFCC

**mientras** iteración < $N$ **hacer**
  **mientras** ventana < $K$ **hacer**
    extraer características: $M_1$: 39 MFCC de $n$ y $M_2$: 39 MFCC de $r$
    // propagar 39 MFCC de $r$ de la entrada hacia la salida de $L$ (ecuaciones 2.8 a 2.16) //
    calcular $f_L(M_2)$
    calcular error: $E(f_L(M_2), M_1)$
    **si** error < error mínimo **entonces**
      // ajustar pesos de $L$ (ecuaciones 2.17 a 2.22) //
      $L_{ae} \leftarrow L$
    **fin si**
  **fin mientras**
**fin mientras**
**devolver** $L_{ae}$

En el segundo tipo de post-filtro, LSTM-2, los MFCC se extraen y utilizan en el entrenamiento de forma semejante a como se realiza en el LSTM-1. Adicionalmente, una memoria auto-asociativa basada en LSTM se entrena para aprender la relación entre los coeficientes de energía del habla sintetizada con los correspondientes al habla natural, siguiendo el procedimiento previamente mostrado en la Figura 4.2 y se describe en el Algoritmo 2.

En la etapa de mejora de nuevas frases, los 39 MFCC se presentan al *autoencoder* entrenado para este fin, mientras que el coeficiente de energía y los 39 MFCC mejorados se presentan a la memoria auto-asociativa, de la cual se utiliza solamente la salida correspondiente al coeficiente de energía. En este segundo tipo de post-filtro se conserva el coeficiente de $f_0$ de la voz sintetizada con HTS.

**Algoritmo 2** Entrenamiento de red LSTM para mejorar el coeficiente de energía del habla sintetizada con HTS

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: Frases de habla HTS alineadas
**Entrada:** $L$: Red LSTM inicializada
**Entrada:** $N$: Número de iteraciones
**Entrada:** $K$: Número de ventanas
**Salida:** $L_{maa_2}$: Red entrenada para eliminar ruido en parámetro de energía

**mientras** iteración $< N$ **hacer**
  **mientras** ventana $< K$ **hacer**
    extraer características: $M_1$: 39 MFCC y energía de $n$ y $M_2$: 39 MFCC de $n$ y energía de $r$
    // propagar 39 MFCC y energía de $r$ de la entrada hacia la salida de $L$ (ecuaciones 2.8 a 2.16) //
    calcular $f_L(M_2)$
    calcular error: $E(f_L(M_2), M_1)$
    **si** error $<$ error mínimo **entonces**
      *a*justar pesos de $L$ (ecuaciones 2.17 a 2.22)
      $L_{maa_2} \leftarrow L$
    **fin si**
  **fin mientras**
**fin mientras**
**devolver** $L_{maa_2}$

El tercer tipo de post-filtro, LSTM-3, contiene una memoria auto-asociativa para proceder de forma semejante a LSTM-2 con el coeficiente de energía, pero en esta ocasión para procesar $f_0$. Las demás características de entrenamiento y mejora coinciden con LSTM-2. Este proceso se ilustra en el Algoritmo 3.

**Algoritmo 3** Entrenamiento de red LSTM para mejorar el coeficiente $f_0$ del habla sintetizada con HTS

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: Frases de habla HTS alineadas
**Entrada:** $L$: Red LSTM inicializada
**Entrada:** $N$: Número de iteraciones
**Entrada:** $K$: Número de ventanas
**Salida:** $L_{maa_1}$: Red entrenada para eliminar ruido en parámetro $f_0$

**mientras** iteración $< N$ **hacer**
  **mientras** ventana $< K$ **hacer**
    extraer características: $M_1$: 39 MFCC y $f_0$ de $n$ y $M_2$: 39 MFCC de $n$ y $f_0$ de $r$
    // propagar 39 MFCC y $f_0$ de $r$ de la entrada hacia la salida de $L$ (ecuaciones 2.8 a 2.16) //
    calcular $f_L(M_2)$
    calcular error: $E(f_L(M_2), M_1)$
    **si** error $<$ error mínimo **entonces**
      *a*justar pesos de $L$ (ecuaciones 2.17 a 2.22)
      $L_{maa_1} \leftarrow L$
    **fin si**
  **fin mientras**
**fin mientras**
**devolver** $L_{maa_1}$

Finalmente, para efectos de comparación, se utiliza un sistema llamado LSTM-S, en el cual una sola red, con la arquitectura de un *autoencoder*, se entrena para aprender el mapeo entre todos los coeficientes del habla sintetizada y el habla natural. Esta red se considera para comparar los beneficios del sistema basado en una colección de redes en lugar de utilizar una sola.

De esta manera, cada post-filtro LSTM intenta aprender a resolver el problema de regresión requerido para transformar los parámetros del habla sintetizada en habla natural. La idea es que los sistemas propuestos mejoran la calidad de cualquier nueva frase producida con HTS para una voz particular, utilizando las redes como un medio de refinar los parámetros y acercarlos cada vez más a los de una voz natural.

En el Algoritmo 4 se describe la utilización de las tres redes entrenadas para mejorar las frases de HTS considerando el sistema más complejo (LSTM-3).

**Algoritmo 4** Mejora de frases HTS de prueba con post-filtros LSTM

**Entrada:** Frases de habla natural, frases de habla HTS, red inicializada, $N$ (número de iteraciones).
**Salida:** Onda procesada mejorada con los post-filtros
**mientras** exista frase **hacer**
Leer actual
extraer características: $f_0$, energía, 39 MFCC
predecir 39 MFCC mejoradas con entradas 39 MFCC ruidosas
predecir $f_0$ mejorada con entrada ($f_0$ ruidosa, 39 MFCC mejoradas)
predecir energía mejorada con entrada (energía ruidosa, 39 MFCC mejoradas)
**fin mientras**
Reconstruir habla con ($f_0$ mejorada, energía mejorada, 39 MFCC mejorados
**devolver** Onda de habla reconstruida

En las Figuras 4.3 a 4.6 se ilustra el proceso de mejora de parámetros con las redes entrenadas descritas anteriormente.

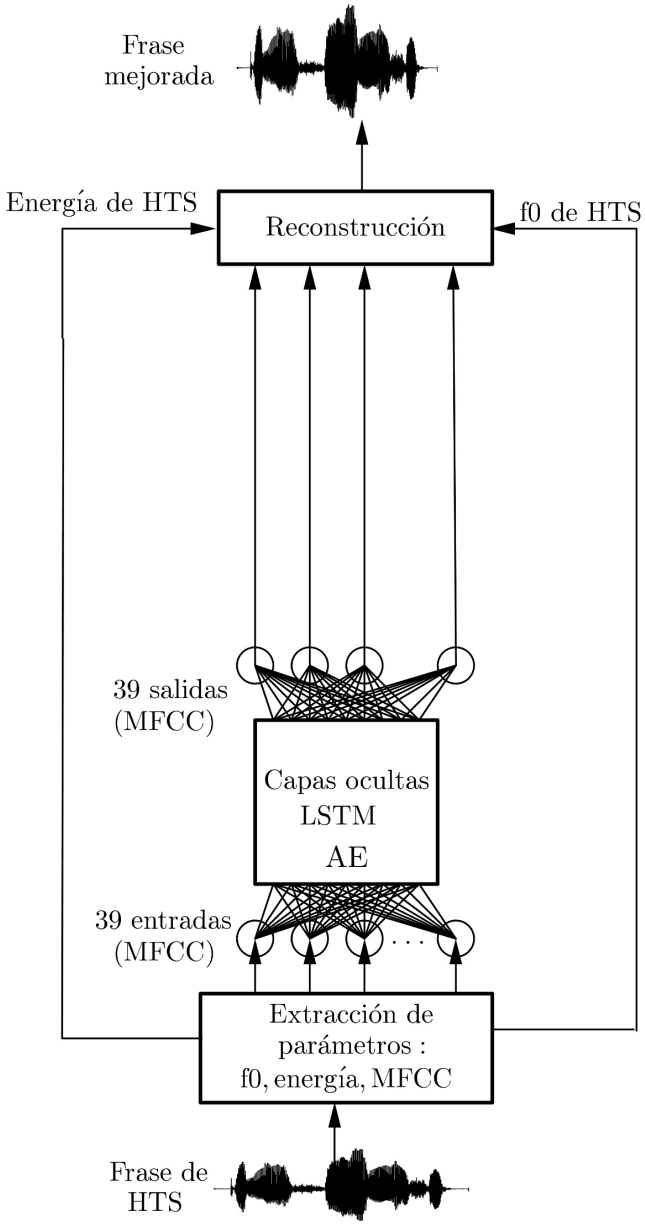



**Figura 4.3:** Mejora de parámetros con el sistema LSTM-1. Solamente los MFCC son procesados con el *autoencoder* (AE) entrenado a partir de los datos.

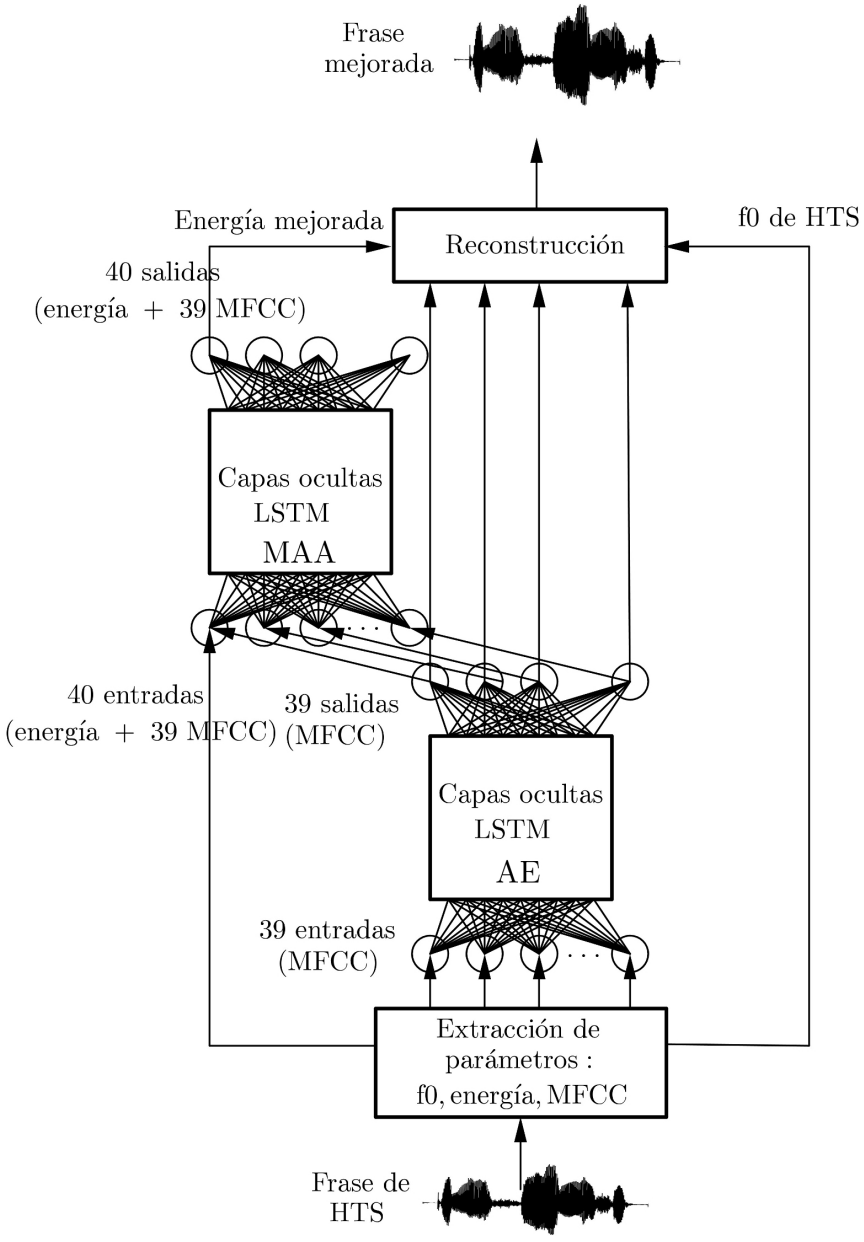



**Figura 4.4:** Mejora de parámetros con el sistema LSTM-2. Los MFCC son procesados con el *autoencoder* (AE) entrenado a partir de los datos y el coeficiente de energía con una memoria auto-asociativa (MAA) a partir de los MFCC mejorados y el coeficiente de energía.

En la siguiente sección se describe el procedimiento experimental diseñado para validar la propuesta en distintas voces generadas con HMM y el sistema HTS.

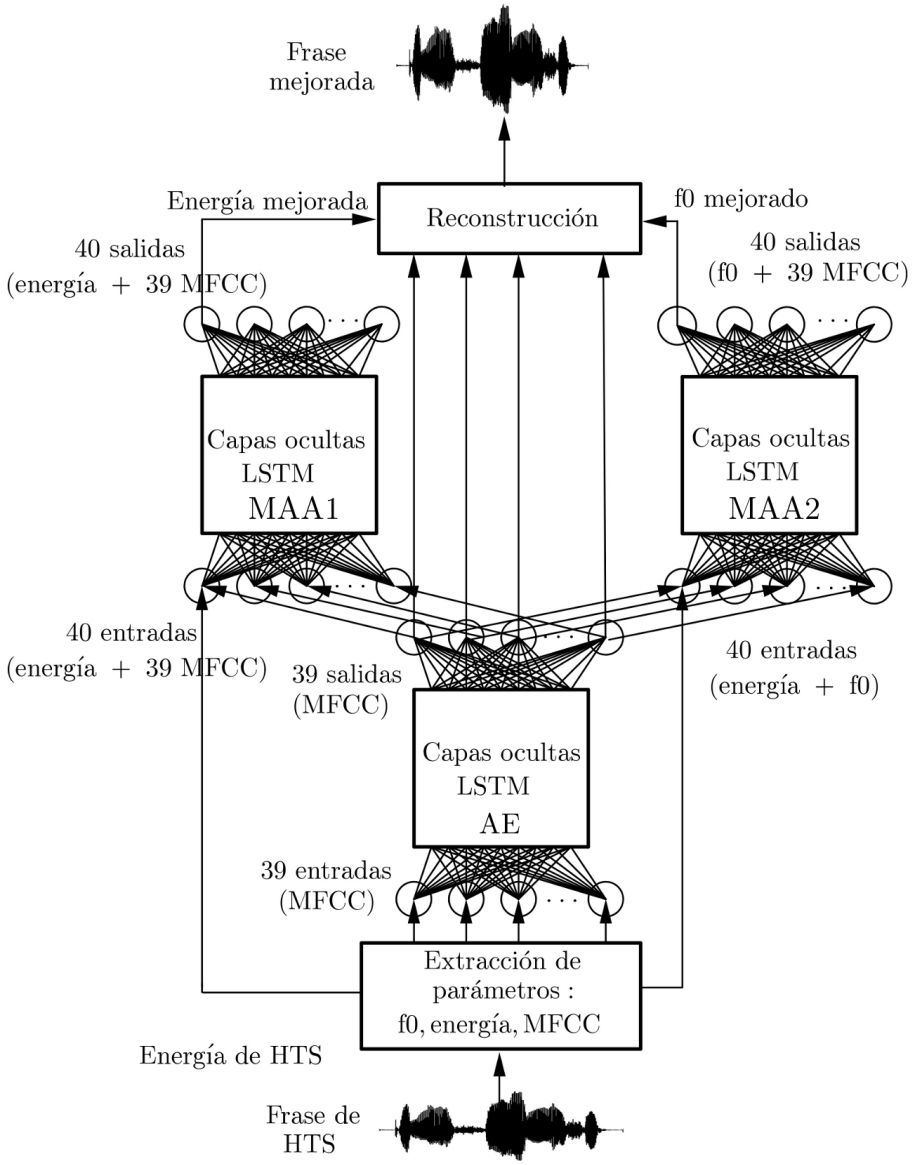



**Figura 4.5:** Mejora de parámetros con el sistema LSTM-3. Los MFCC son procesados con el *autoencoder* (AE) y los coeficientes de energía y $f_0$ con dos memorias auto-asociativas independientes (MAA1 y MAA2), a partir de los MFCC mejorados y los coeficientes de HTS.

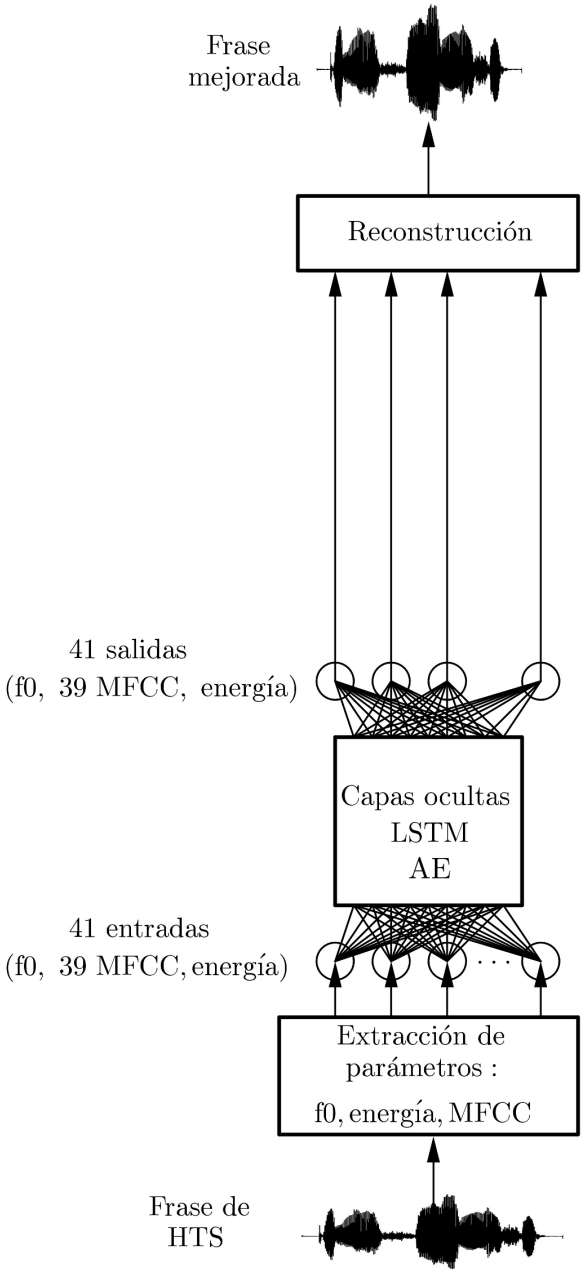



**Figura 4.6:** Mejora de parámetros con el sistema LSTM-S. Todos los coeficientes de la parametrización se procesan con un único *a*utoencoder LSTM.

## 4.3 Experimentación

### 4.3.1 Descripción de los datos

Las bases de datos *CMU Artic* fueron producidas en el Instituto de Tecnologías del lenguaje de la Universidad Carnegie Mellon en los Estados Unidos de América, en idioma inglés con acento de dicho país. Inicialmente se desarrollaron para la producción de voces con el método de selección de unidades, y consisten en 1132 frases seleccionadas de textos del Proyecto Gutenberg, los cuales se encuentran libres de derechos de autor. Estas bases de datos se encuentran disponibles de forma gratuita en Internet (`http://www.festvox.org/cmu_arctic/`).

Incluyen tanto grabaciones de hablantes masculinos como femeninos. Un reporte detallado de la estructura y contenido de las bases de datos se encuentra disponible en el Reporte del Instituto de Tecnologías del Lenguaje CMU-LTI-03-177 18 [54]. Las voces elegidas para la experimentación se identifican con tres letras, siguiendo la notación definida originalmente, y son: BDL (masculina), CLB (femenina), RMS (masculina), JML (masculina) y SLT (femenina).

### 4.3.2 Extracción de características

Para establecer los datos que corresponden a entrenamiento, validación y prueba de los sistemas propuestos, se generó una réplica de cada base de datos con el sistema *H*TS-Parallel descrito en la Sección 3.2, de forma tal que de las 1132 frases se cuenta con una versión natural y una sintetizada con el sistema HTS. Para procesar estos archivos y establecer las entradas y salidas de las redes LSTM, los archivos se remuestrean a 16kHz para extraer los parámetros (requisito del sistema Ahocoder, utilizado con este fin en el presente trabajo).

En este sistema, se extrae la frecuencia fundamental $f_0^k$ (de valor 0 en caso de los sonidos no vocales), 39 MFCC y un coeficiente correspondiente a la energía de cada ventana de 10 ms. De esta manera, cada una de las ventanas se representa con un vector de dimensión 41: $V_k = \left[f_0^k, e^k, MFCC_k^1, \ldots, MFCC_k^{39}\right]$, y existe uno de estos vectores de voz natural y uno correspondiente a voz artificial.

Después del proceso de mejora con las redes entrenadas, a la salida de cada uno de los tipos de post-filtros es posible generar nuevamente una forma de onda de audio con los coeficientes, utilizando el sistema Ahodecoder, el cual está integrado en el anterior.

### 4.3.3 Experimentos

Con cada una de las voces parametrizadas siguiendo el proceso de extracción de parámetros descrito en la sección anterior, el conjunto de vectores resultantes se dividió en entrenamiento, validación y prueba. La cantidad de datos de cada voz se muestra en la Tabla 4.1. Como se puede observar, existen diferencias entre la cantidad de datos disponibles en cada una. Esto se debe a que, a pesar de que cada voz cuenta con las mismas frases, cada voz fue grabada con una tasa de habla distinta.

**Tabla 4.1:** Cantidad de datos (vectores) disponibles para cada voz en la base de datos

| Voz | Género | Total | Entrenamiento | Validación | Prueba |
|---|---|---|---|---|---|
| BDL | Masculino | 676554 | 473588 | 135311 | 67655 |
| SLT | Femenino | 677970 | 474579 | 135594 | 67797 |
| CLB | Femenino | 769161 | 538413 | 153832 | 76916 |
| RMS | Masculino | 793067 | 555147 | 158613 | 79307 |
| JMK | Masculino | 635503 | 541856 | 62135 | 31512 |

Para definir la cantidad de unidades en las capas ocultas de las redes LSTM, se utilizó un procedimiento de prueba y error, considerando desde una hasta tres capas ocultas, y desde cincuenta unidades hasta trescientas en cada una, en saltos de cincuenta en cada caso. En esta prueba preliminar, se utilizó una cantidad reducida de los datos, y se consideró no solamente el error medio obtenido a la salida, sino el tiempo requerido para el entrenamiento. La selección final fue una arquitectura de tres capas ocultas con ciento cincuenta unidades LSTM en la primer capa, cien en la segunda y ciento cincuenta en la tercera. La cantidad de entradas y salidas está en correspondencia con el tipo de LSTM descrito previamente. En total se utilizaron diez autocodificadores o *autoencoders* (uno para los coeficientes MFCC de cada voz y uno para todos los parámetros en LSTM-S) y diez memorias auto-asociativas (dos por cada una de las voces, para $f_0$ y energía).

El proceso de entrenamiento fue acelerado con hardware GPU NVIDIA, con el cual se requiere aproximadamente siete horas para entrenar cada red. Este tiempo es considerable en comparación con el entrenamiento de una red perceptrón multi-capa con la misma arquitectura, el cual se reduce en un factor aproximado de treinta veces. Sin embargo, los resultados no se acercan en

calidad a los de las redes LSTM. Es importante mencionar que una vez que cada red ha sido entrenada, el tiempo requerido para mejorar las nuevas frases no difiere al de otros tipos de redes, por lo que la mejora de una frase de habla sintetizada se puede realizar cercano al tiempo real.

## 4.4 Evaluación

Para evaluar los resultados obtenidos con los sistemas de mejora de post-filtros en las voces HTS, se seleccionaron tres medidas objetivas. Estas medidas han sido utilizados con frecuencia en publicaciones relacionadas con la mejora de señales de voz, y es posible utilizarlas en esta tesis para habla sintetizada gracias a la alineación realizada con el habla natural, con el sistema descrito en la Sección 3.2.

Las medidas son:

- Relación señal a ruido segmental en el dominio de la frecuencia (*Frequency Domain Segmental SNR*), $\text{SegSNR}_\text{f}$: Es una medida basada en ventanas para estimar la calidad del habla, calculada al promediar la SNR de cada ventana de acuerdo con la siguiente ecuación:

$$\text{SegSNR}_\text{f} = \frac{10}{N} \sum_{i=1}^{N} \log \left[ \frac{\sum_{j=0}^{L-1} S^2(i,j)}{\sum_{j=0}^{L-1} \left(S(i,j) - X(i,j)\right)^2} \right] \tag{4.1}$$

  donde $X(i, j)$ es el coeficiente de la Transformada de Fourier de la ventana $i$ en la frecuencia $j$, y $S(i, j)$ es el coeficiente correspondiente del habla procesada. $N$ es el número de ventanas y $L$ el número de frecuencias. Para las evaluaciones realizadas a los experimentos de esta tesis, se utilizaron ventanas de 256 muestras, con un traslape de 128 muestras. Los valores están restringidos al intervalo $[-20, 35]$ dB.

- PESQ: Esta medida utiliza un modelo psicoacústico para predecir la calidad del habla percibida subjetivamente. Está definida en la recomendación ITU-T P.862.ITU [55]. Los resultados se dan en el intervalo $[0.5, 4.5]$, donde 4.5 corresponde a una coincidencia perfecta con la señal original.

  PESQ se calcula de acuerdo con la ecuación:

$$\text{PESQ} = a_0 + a_1 D_{ind} + a_2 A_{ind} \tag{4.2}$$

donde $D_{ind}$ es la perturbación promedio, y $A_{ind}$ es la perturbación asimétrica [56]. Los $a_k$ se eligen para optimizar la medida PESQ de manera que se refleje la distorsión del habla, el ruido, y la calidad general [57].

- *Weighted-slope spectral distance (WSS):* Esta medida calcula las diferencias entre las pendientes de bandas del espectro, al medir la diferencia entre magnitudes adyacentes. Se calcula utilizando la ecuación [57]

$$WSS = \frac{1}{N}\sum_{i=0}^{N}\frac{\sum_{j=1}^{K} W(i,j)\,(S_s(i,j) - S_x(i,j))^2}{\sum_{j=1}^{K} W(i,j)} \tag{4.3}$$

donde $S_s(j,i)$ y $S_x(j,i)$ son las pendientes del espectro en la $j$-ésima banda de frecuencia en la ventana $i$, $K$ es el número de bandas de frecuencia y $W(j,i)$ son pesos asignados a cada diferencia, calculados de acuerdo con lo establecido en [58].

Además de las tres medidas anteriores, en este capítulo se utiliza un ASR para comparar la inteligibilidad del resultado con un único reconocedor de palabras de propósito general, llamado Speechmatics [59]. Con éste, dado el texto que se ha pronunciado en la frase original se puede calcular una tasa de error de palabras (WER) de cada conjunto de frases procesadas de la misma manera. Esta medida se define como

$$WER = \frac{S + D + I}{N} \tag{4.4}$$

donde $S$ es el número de sustituciones (palabras reemplazadas por otras), $D$ el número de palabras no reconocidas, $I$ la cantidad de palabras introducidas que no formaban parte del texto original, y $N$ el número total de palabras en las frases.

## 4.5 Resultados y discusión

El análisis de resultados se realiza en dos niveles: En el primero se hace una comparación de los algoritmos entre sí, para determinar cuál es el mejor, y si existen diferencias significativas entre ellos. En un segundo nivel, se comparan todos los tipos de post-filtros propuestos con las voces HTS, para determinar cuáles de los sistemas mejoran significativamente la voz sintetizada. Este segundo análisis se lleva a cabo ya que es posible que algunos de los mejores resultados de las medidas aplicadas no representen mejoras importantes a las características del habla producida con HMM.

La significancia estadística de las diferencias se realiza utilizando los valores de las medidas de evaluación en las cincuenta frases del conjunto de prueba en un test de Análisis de Varianza (ANOVA). Luego de aplicar el ANOVA entre el conjunto de valores del mejor resultado y los demás algoritmos, se realiza una Prueba HSD de Tukey para determinar las diferencias entre los resultados y la voz de HTS. Todas estas pruebas se realizaron con un nivel de significancia de 0.95, el cual es un valor usual en la literatura para pruebas estadísticas comparativas entre grupos de datos.

Como se ha señalado, los fonemas contenidos en estas bases de datos se degradan en el proceso de producir voces con HTS, debido al efecto de sobre-suavizamiento de los parámetros, como ha sido reportado previamente en las referencias [60]. Dado que cada fonema tiene diferentes probabilidades de ocurrencia, es de esperar que el impacto en la calidad de los fonemas sea diferente en cada uno, acorde con la cantidad de éstos emitidos en las grabaciones.

Los tipos de post-filtros LSTM pueden afectar los fonemas de distintas maneras: LSTM-1 solamente afecta la parte espectral, y se puede esperar que beneficie a todos los fonemas de forma similar. LSTM-2, el cual también considera el parámetro de energía, puede beneficiar a ciertos fonemas al proveer de un valor más cercano al natural, el cual pudo ser afectado en el proceso de HTS. Un ejemplo claro pueden ser las vocales, las cuales tienen valores mayores de energía en el habla natural. Finalmente, LSTM-3, el cual considera la mejora en la frecuencia fundamental afecta solamente a los fonemas sonoros, es decir, aquellos para los cuales este valor cumple $f_0 > 0$. Las evaluaciones, sin embargo, son aplicadas a las frases completas, pero se espera que el beneficio de los post-filtros, aunque se dé en segmentos de éstas, sean detectados en las medidas.

Los resultados de la evaluación con la medida WSS se muestran en la Tabla 4.2. Se observa cómo el post-filtro LSTM-2 obtiene el mejor resultado en tres de las cinco voces. En las otras dos, los mejores resultados se obtuvieron con el post-filtro LSTM-1, aunque la diferencia del LSTM-2 con éste no es significativa.

Para la voz HTS, los valores de WSS son más altos que los de estos post-filtros, lo cual indica el beneficio de aplicar los post-filtros en todos los casos. Para el caso de LSTM-3, en tres de las voces las diferencias no son significativas con el mejor resultado, pero presenta un caso muy desfavorable para la voz SLT. En ninguno de los casos el sistema LSTM-S presentó valores que mejoraran los del habla artificial de HTS.

Los resultados de las medidas PESQ se muestran en la Tabla 4.3. Se destacan los post-filtros LSTM-1 y LSTM-2, los cuales tienen los mejores valores para las cinco voces o resultados que no difieren significativamente con el mejor. LSTM-3 tiene dos de los mejores resultados, pero, como en la medida anterior, su valor es considerablemente inferior a los demás para la voz SLT. En este caso, LSTM-S sí presentó buenos resultados para las voces CLB y JMK.

**Tabla 4.2:** Comparación de la media de resultados de la medida WSS para el conjunto de prueba. Los valores menores indican mejor resultado. Las medidas se realizaron con referencia a la voz natural.

| Voz | HTS | LSTM-S | LSTM-1 | LSTM-2 | LSTM-3 |
|---|---|---|---|---|---|
| CLB | 47.79 | 55.35 | **34.92** | **33.98*** | **36.35** |
| JMK | 43.35 | 51.39 | **31.59*** | **32.24** | 38.37 |
| RMS | 44.99 | 54.32 | **32.85** | **32.65*** | **32.68** |
| SLT | 55.94 | 68.09 | **43.90*** | **44.22** | 60.44 |
| BDL | 48.31 | 57.38 | **37.5** | **37.06*** | **38.90** |

*Nota:* * indica el mejor resultado.
*Nota:* Se resaltan en negrita los resultados que no difieren significativamente del mejor, de acuerdo con la prueba ANOVA.

**Tabla 4.3:** Comparación de la media de resultados de la medida PESQ para el conjunto de prueba. Los valores mayores indican mejor resultado. Las medidas se realizaron con referencia a la voz natural.

| Voz | HTS | LSTM-S | LSTM-1 | LSTM-2 | LSTM-3 |
|---|---|---|---|---|---|
| CLB | **1.26** | **1.25*** | **1.21** | **1.23** | 1.00 |
| JMK | **1.38** | **1.45*** | **1.44** | **1.38** | **1.19** |
| RMS | **1.58** | 1.24 | **1.55** | **1.57** | **1.64*** |
| SLT | **1.04** | **0.95** | **0.97** | **0.99*** | 0.52 |
| BDL | **1.45** | 1.19 | **1.34** | **1.35** | **1.38*** |

*Nota:* * indica el mejor resultado.
*Nota:* Se resaltan en negrita los resultados que no difieren significativamente del mejor, de acuerdo con la prueba ANOVA.

La Tabla 4.4 muestra la comparación de las medias de la medida $\text{SegSNR}_\text{f}$. Para ésta, los tres tipos de post-filtros que consideran la mejora individual de los parámetros (LSTM-1, LSTM-2 y LSTM-3) mejoran los resultados de la voz HTS y del post-filtro único LSTM-S. Los mejores resultados se obtuvieron con LSTM-1 y LSTM-2.

Comparando los resultados de las Tablas 4.2 a 4.4, se puede observar que los post-filtros LSTM-1 y LSTM-2 generalmente presentan los mejores resultados para las cinco voces. En dos casos (voces RMS y BDL) el post-filtro LSTM-3 obtuvo los mejores valores de PESQ, y para estas mismas voces los resultados de las medidas $\text{SegSNR}_\text{f}$ y WSS no son significativamente diferentes del mejor.

La razón por la cual se obtuvieron resultados desfavorables para la voz SLT se puede observar en la Figura 4.7, en la cual los diagramas de violín muestran que la distribución de valores de $f_0$ de la voz original difiere considerablemente de los de la voz sintetizada con HTS. Para las otras

**Tabla 4.4:** Comparación de la media de resultados de la medida $\text{SegSNR}_\text{f}$ para el conjunto de prueba. Los valores mayores indican mejor resultado. Las medidas se realizaron con referencia a la voz natural.

| Voz | HTS | LSTM-S | LSTM-1 | LSTM-2 | LSTM-3 |
|---|---|---|---|---|---|
| CLB | 1.29 | -6.53 | **2.69*** | 1.96 | **2.25** |
| JMK | -0.94 | -12.99 | **2.21*** | **1.67** | 0.46 |
| RMS | 0.05 | -10.12 | **2.43** | **2.59*** | **2.52** |
| SLT | -1.72 | -8.95 | **1.09** | **1.32*** | 0.02 |
| BDL | -1.36 | -12.39 | **1.51*** | **1.33** | **1.14** |

*Nota:* * indica el mejor resultado.
*Nota:* Se resaltan en negrita los resultados que no difieren significativamente del mejor, de acuerdo con la prueba ANOVA.

cuatro voces, los diagramas de violín muestran una correspondencia cercana entre los valores de $f_0$ del habla natural y sintetizada, lo cual indica que el sistema HTS fue capaz de entrenar los HMM correspondientes para que produjeran estos valores de forma similar a la voz original.

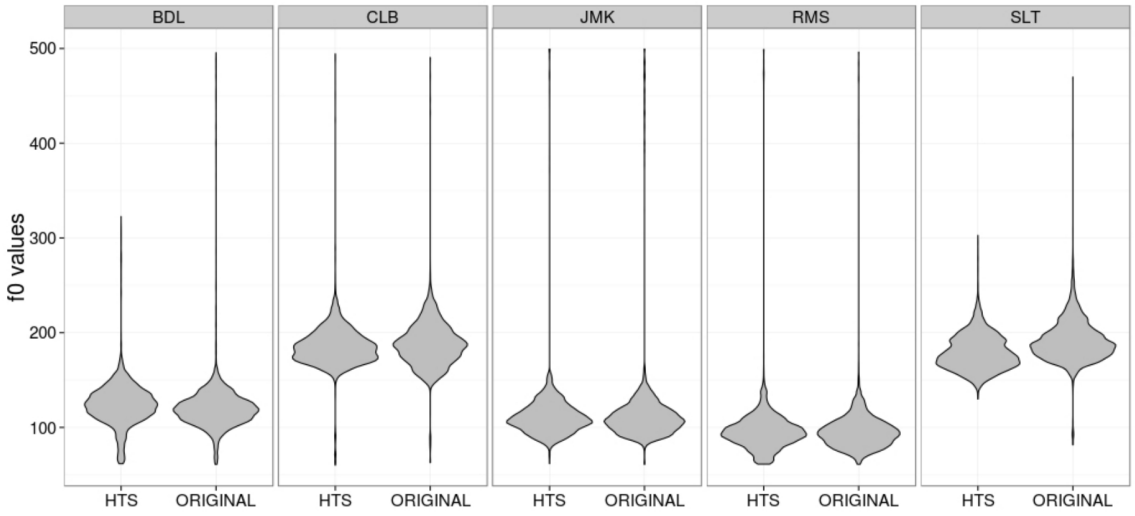



**Figura 4.7:** Diagramas de violín para los valores de $f_0$ de las voces naturales y sintetizadas con el sistema HTS.

Esto significa que para los casos en los cuales la distribución de valores de $f_0$ no difiere notablemente entre la voz de HTS y la natural, el post-filtro LSTM-3 puede mejorar estos valores. Por su parte, en la voz SLT, la diferencia entre el parámetro $f_0$ de la voz artificial y la natural hace que el mapeo de la memoria auto-asociativa adicional que contempla LSTM-3 sea inferior a los demás casos.

Por lo tanto, una discrepancia entre los valores de $f_0$ entre una voz y aquella creada a partir de

ésta con el sistema HTS (la cual puede ser detectada inmediatamente después del proceso de generación con el sistema) contiene distorsiones que limitan la capacidad del post-filtro LSTM-3 para mejorar el resultado. Esta limitación no se presenta en aquellos casos para los cuales las distribuciones de valores de $f_0$ no son significativas entre las voces naturales y las artificiales del sistema HTS.

Para el último análisis estadístico de este capítulo, se reporta la comparación entre las medidas de evaluación de las frases de prueba de los post-filtros y los del habla sintetizada con el sistema HTS, es decir, previo a la aplicación de los sistemas propuestos. Esta comparación indica los posibles beneficios que se derivan de aplicar los post-filtros LSTM. Utilizando la Prueba HSD de Tukey, la comparación con la voz HTS se realiza algoritmo a algoritmo, y los resultados se muestran en la Tabla 4.5. En esta tabla, los símbolos ✓ indican una mejora significativa producida por el post-filtro para la medida correspondiente. Lo señalado como "ns" significa que la mejora no es estadísticamente significativa, mientras que los espacios en blanco indican que el resultado del post-filtro empeoró en la medida con respecto al de la voz HTS.

**Tabla 4.5:** Mejora significativa de los resultados para el conjunto de prueba con relación a la voz HTS, de acuerdo con la prueba HSD de Tukey

| Medida | Voz | LSTM-S | LSTM-1 | LSTM-2 | LSTM-3 |
|---|---|---|---|---|---|
| WSS | CLB | | ✓ | ✓ | ✓ |
| | JMK | | ✓ | ✓ | ✓ |
| | RMS | | ✓ | ✓ | ✓ |
| | SLT | | ✓ | ✓ | |
| | BDL | | ✓ | ✓ | ✓ |
| PESQ | CLB | ns | ns | ns | |
| | JMK | ns | ns | ns | ns |
| | RMS | | ns | ns | ns |
| | SLT | ns | ns | ns | |
| | BDL | | ns | ns | ns |
| $\text{SegSNR}_{\text{f}}$ | CLB | | ✓ | ✓ | ✓ |
| | JMK | | ✓ | ✓ | ✓ |
| | RMS | | ✓ | ✓ | ✓ |
| | SLT | | ✓ | ✓ | ✓ |
| | BDL | | ✓ | ✓ | ✓ |

*Nota:* Los ✓ indican mejora significativa entre las medidas del post-filtro correspondiente con respecto a los de la voz HTS. "ns" denota que el resultado no difiere de forma estadísticamente significativa con la voz HTS. Los espacios en blanco indican resultados desfavorables para los post-filtros.

De acuerdo con esta tabla, los post-filtros LSTM-1 y LSTM-2 mejoran o su resultado es comparable a las voces de HTS en todos los casos. Sin embargo, en las tablas anteriores se

observa que LSTM-2 tiene más resultados entre los mejores que LSTM-1. Además, LSTM-2 presentó al menos un mejor resultado en las tres medidas, superando de esta manera a LSTM-1. A pesar de que el post-filtro LSTM-3 da los mejores resultados en algunas de las evaluaciones, las dificultades en mejorar el parámetro $f_0$ cuando los valores de la voz artificial de HTS difieren a los de la voz natural, lo hacen una opción menos viable en estas propuestas.

Un ejemplo de un parámetro generado por la voz HTS y la mejora obtenida con el post-filtro LSTM-2 se muestra en la Figura 4.8. Se puede observar cómo el post-filtro ajusta la trayectoria del parámetro de mejora manera en relación con el producido solamente con el sistema HTS.

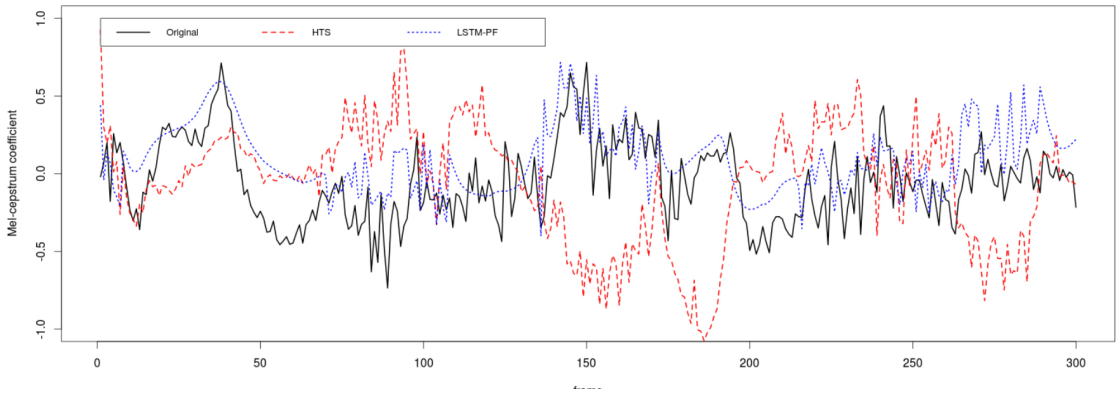



**Figura 4.8:** Ilustración de la mejora en el quinto coeficiente MFCC por el post-filtro LSTM-2, para la voz BDL.

En la Figura 4.9 se muestra una comparación de tres espectrogramas de la frase “Will we ever forget it?” tanto de la voz natural, sintetizada con HTS y luego de la aplicación del post-filtro LSTM-2. Se destaca en esta figura las bandas que se producen en las altas frecuencias en las voces HTS, las cuales no están presentes en la voz natural. Luego de la aplicación del post-filtro se observa que éstas se desvanecen, haciendo esta representación más cercana a la de la voz original.

## 4.6 Resumen de contribuciones

En este capítulo se presentó una nueva propuesta para mejorar los resultados de la síntesis estadística paramétrica de voz generada con HMM en el sistema HTS, aplicando post-filtros LSTM. Esta propuesta ha considerado desde el enfoque de un solo post-filtro (previamente analizado en la literatura con otras unidades distintas a LSTM) hasta una colección de éstos en diferentes arquitecturas, las cuales han mostrado los beneficios en mejorar la calidad.

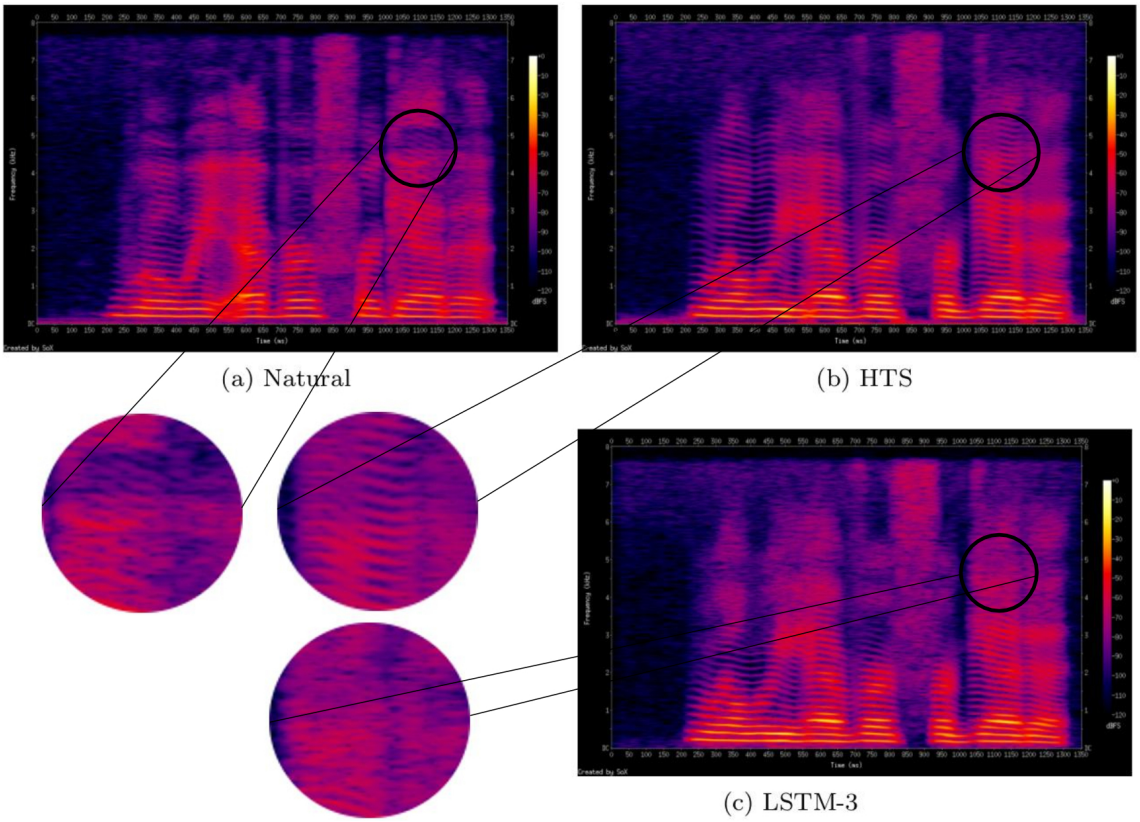



**Figura 4.9:** Comparación de tres espectrogramas de la frase “Will we ever forget it?”, de la voz RMS.

Se describió la manera de entrenar y aplicar los post-filtros en aquellos casos donde se combinan varias arquitecturas, y se utilizan los resultados de unos para mejorar otros parámetros en los siguientes. Dado que en este caso se cuenta con frases de habla natural y sintetizada que se encuentran alineadas, fue posible aplicar medidas que se utilizan tradicionalmente en el área de mejora de señales de voz: WSS, PESQ y $\text{SegSNR}_\text{f}$. Para realizar una comparación amplia, se utilizaron cinco voces, tanto masculinas como femeninas, en las cuales se utilizaron las tres medidas.

Los resultados muestran cómo dos de los sistemas de post-filtros propuestos, LSTM-1 y LSTM-2 mejoraron significativamente las medidas $\text{SegSNR}_\text{f}$ y WSS en comparación con la voz producida por el sistema HTS. Los valores de PESQ no difieren significativamente en estos casos con los de la voz HTS.

El tercer tipo de post-filtro propuesto, LSTM-3, mostró resultados favorables en un conjunto reducido de casos, y un análisis más detallado realizado a este caso considera que esto es causado por la dificultad de mapear los valores de $f_0$ entre las voces artificiales y naturales.

En la gran mayoría de los casos, la idea de aplicar una colección de post-filtros para parámetros

específicos del habla mejoró considerablemente los resultados de aplicar solamente uno de ellos. Observaciones realizadas a espectrogramas y las trayectorias de los parámetros a lo largo de una frase muestran la conveniencia de aplicar de esta manera las redes LSTM.

# 5

# POST-FILTROS HÍBRIDOS

*En este capítulo se describe la segunda propuesta de utilización de post-filtros múltiples para mejorar los resultados de la síntesis de voz basada en HMM, en combinación con filtros Wiener.*

## Índice

## 5.1 Introducción

En el presente capítulo se introduce la idea de utilizar un sistema híbrido para mejorar el habla sintetizada generada métodos estadísticos paramétricos. En el Capítulo 4 se presentaron los post-filtros LSTM, aplicados como una colección de *autoencoders* y memorias auto-asociativas para distintos parámetros del habla sintetizada, y de esta manera acercar la calidad del resultado al de una voz natural.

Los post-filtros aplicados de esta manera abordan el problema de regresión, al realizar una aproximación del mapeo necesario para transformar los parámetros del habla sintetizada en los de habla natural. En el presente capítulo se introducirá una etapa previa a la aplicación de los post-filtros, la cual consiste en filtros Wiener. La motivación principal proviene del hecho que la voz producida con HTS tiene un componente de ruido de bajo nivel, el cual está presente en el proceso de entrenamiento de las redes LSTM.

Esto significa que las redes no solamente deben aprender a transformar los parámetros como tales, sino que deben eliminar este ruido. La posibilidad de eliminar el ruido de forma previa al entrenamiento de las redes constituye una oportunidad de mejorar su eficacia en la tarea de mejora de señales de habla.

La contribución principal del presente capítulo es extender las técnicas presentadas en el Capítulo 4 hacia un nuevo enfoque híbrido en dos etapas, lo cual abre nuevas posibilidades a este campo, con combinaciones de algoritmos de distinta naturaleza que puedan atacar problemas específicos de la señal de habla y aportar al incremento de su naturalidad.

## 5.2 Filtros Wiener

Los filtros Wiener fueron introducidos en la década de 1940 por Norbert Wiener, planteándolos como una solución al problema de estimación de señales estacionarias (constantes en sus parámetros estadísticos a lo largo del tiempo) [61]. El objetivo del filtro Wiener, a partir de su implementación en el área de mejora de señales, es la eliminación del ruido presente en una señal.

Típicamente, los filtros se diseñan para un rango de frecuencia específico, deseando que a su salida se obtenga una señal que sea tan cercana como sea posible a la señal limpia. Las implementaciones más comunes de filtro Wiener se pueden caracterizar por [62]:

- Tanto la señal como el ruido (aditivo) deben ser estacionarios con características espectrales conocidas.
- El filtro se debe poder implementar físicamente (es decir, debe ser causal).
- Se utiliza como criterio de desempeño la suma de errores cuadráticos.

Si bien estos filtros han sido implementados en aplicaciones como reducción de ruido en imágenes o en general en transferencias de datos, en el caso de habla, resulta conveniente realizar estimaciones en los intervalos de silencio, donde el ruido presente se encuentra aislado [63]. Si bien la formulación del filtro requiere conocer la densidad espectral de potencia, en la práctica se debe realizar siempre una estimación de ésta.

Existen varios métodos para este parámetro, de cuya precisión depende la capacidad del filtro para realizar su tarea. Por ejemplo, se utilizan métodos iterativos, estimadores de SNR u otros provenientes de otras técnicas, como los de sustracción espectral.

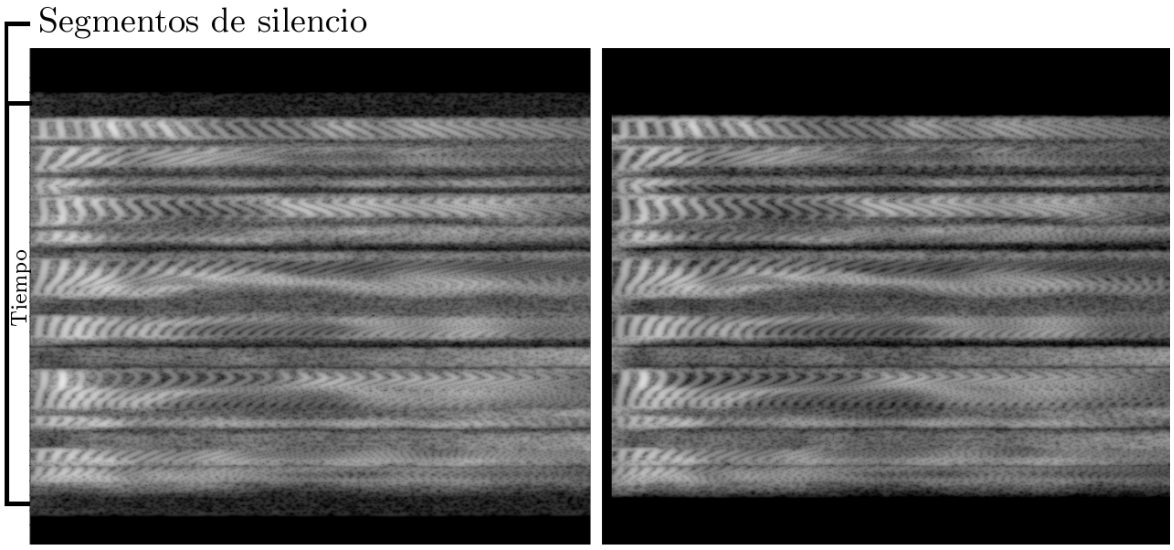
En el caso de señales de habla, estos estimadores y el filtro Wiener han probado su utilidad en numerosas referencias, contemplando distintos tipos de ruido y condiciones. En el presente capítulo se utiliza la implementación descrita en [64]. Detalles adicionales se muestran en la Sección 9.4.2.

## 5.3 Sistema propuesto

En el análisis de las señales de habla sintetizada a partir de HMM, por ejemplo utilizando el sistema HTS, se observa que los parámetros de las voces artificiales tienen una mayor variación y particularidades en comparación a las voces artificiales producidas. Esto se debe principalmente al llamado sobre-suavizamiento propio del proceso de entrenamiento y generación de parámetros con los HMM [38].

De acuerdo con el planteamiento del problema de mejorar las voces artificiales con algoritmos de aprendizaje profundo presentado en la Sección 3.1.1, los parámetros del habla sintetizada $R_Y$ se consideran una versión ruidosa o distorsionada de los de la correspondiente habla natural $R_X$, y se puede plantear el problema de mejorar los primeros utilizando la ecuación 3.3. En el presente capítulo proponemos la utilización de un sistema híbrido en dos etapas, la primera de las cuales es un filtro Wiener para reducción de ruido.

La razón de utilizar el filtro Wiener radica en su capacidad probada para reducción de ruido, así como en las implementaciones que se encuentran disponibles para distintos lenguajes de




(a) HTS (a) Wiener

**Figura 5.1:** Espectrogramas de una frase generada con HTS (izquierda) y posteriormente filtrada con Wiener (derecha).

programación. En particular, para señales de voz ha sido estudiada su eficacia en reducción de ruido en múltiples referencias, como en [65][66][67][68], por mencionar algunas recientes.

Se está considerando al habla producida con HMM como una distorsión del habla original con un componente de ruido. A la salida del filtro Wiener se obtiene una nueva versión del habla sintetizada, denotada $\overline{\mathbf{R}}_\mathbf{Y}$. Se espera que esta nueva versión represente habla más cercana a la natural, o al menos que la reducción del componente de ruido permita un mejor mapeo de los parámetros en la segunda etapa, que considera post-filtros LSTM.

De esta manera, luego de la aplicación del filtro Wiener, las redes LSTM se entrenan tomando como entradas los parámetros del habla filtrada, y como salidas el habla natural. El problema principal se puede reescribir:

$$E(\vec{R_W}) = \|f(\overline{\mathbf{R}}_\mathbf{Y}; \mathbf{R_W}) - \mathbf{R_X}\|^2, \tag{5.1}$$

donde $\mathbf{R_W}$ es la matriz conformada por los parámetros del habla sintetizada $R_Y$ y natural $R_X$.

La presencia del componente de ruido de bajo nivel se puede comprobar con la observación de los segmentos de silencio al principio y final de cada frase de la base de datos. Por ejemplo, en la Figura 5.1, donde se muestra el espectrograma de una frase de habla producida con HMM y posteriormente procesada con un filtro Wiener, en la cual los segmentos del inicio y fin de la frase se ven notablemente libres de ruido en el segundo caso.

En el presente capítulo, todas las frases de habla producidas con el sistema HTS han sido alineadas temporalmente con las de habla natural utilizando el sistema *H*TS-Parallel, de manera que existe una correspondencia precisa entre los parámetros extraídos de ambas. En la primera etapa de la propuesta, todas las frases de habla artificial son filtradas utilizando Wiener, de manera que se cuenta con tres versiones de cada una:

1. La frase original, grabada con voz natural.
2. La frase sintetizada, producida a partir del sistema HTS.
3. La frase sintetizada anterior, filtrada con Wiener.

Para cada una de estas tres versiones de las frases, se definen ventanas de 10 ms en cada una, para extraer vectores que consisten en un coeficiente para $f_0$, un coeficiente para energía y 39 coeficientes MFCC, utilizando el sistema Ahocoder [53]. De forma semejante a lo presentado en el Capítulo 4, se utilizarán tres tipos de post-filtros, combinando *autoencoders* y memorias auto-asociativas de unidades LSTM:

- LSTM-1: Las entradas de la red LSTM, con arquitectura de *autoencoder* consistirán en los parámetros MFCC de la versión sintetizada o procesada con filtro Wiener, mientras que las salidas serán los correspondientes MFCC del habla natural. El coeficiente de energía y el de $f_0$ se tomarán sin cambios, ya sea del habla generada con HTS o bien de la filtrada con Wiener.
- LSTM-2: Considera un *autoencoder* entrenado con las condiciones descritas en LSTM-1, pero adicionalmente incluye una memoria auto-asociativa para mejorar ya sea el parámetro proveniente de la voz HTS o de su versión filtrada con Wiener.
- LSTM-3: Incluye una memoria auto-asociativa adicional a la indicada en LSTM-2, la cual se entrena y opera de forma semejante para el parámetro $f_0$.

Los esquemas de aplicación de los post-filtros coinciden con los indicados en el Capítulo 4, así como los procedimientos descritos en los algoritmos 1 a 4. La única variación radica en que las entradas a las redes son las versiones de la señal filtradas con Wiener en lugar de la señal de voz HTS.

Se excluirá en esta ocasión el post-filtro LSTM-S, dado que sus resultados son notablemente inferiores a los demás, de acuerdo con los resultados del Capítulo 4.

La principal intención de considerar estos sistemas es determinar si el problema de regresión que se plantea para transformar los parámetros del habla sintetizada hacia la natural, obtiene una mejor solución con la aplicación previa del filtro Wiener. En la Figura 5.2 se esquematiza el sistema propuesto para el caso en que se aplican las tres redes LSTM, entrenadas de forma individual.

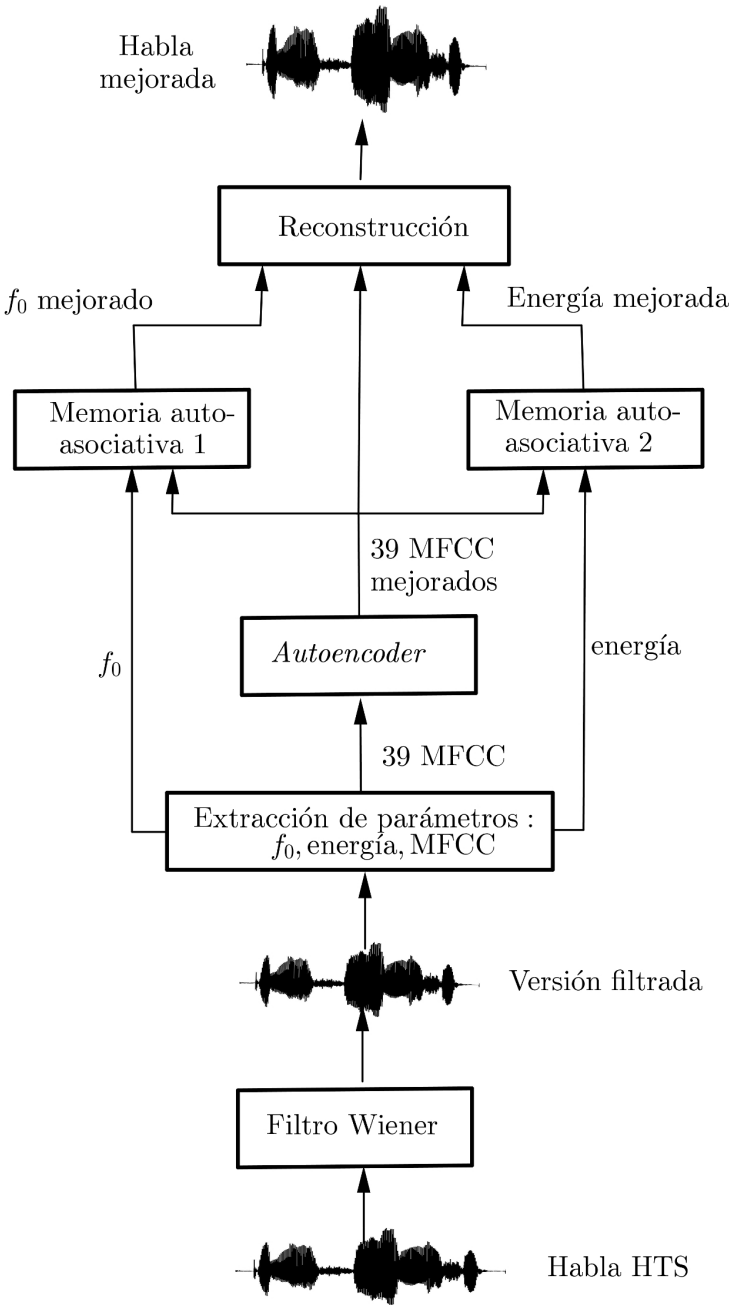



**Figura 5.2:** Representación del sistema propuesto, para el caso de tres redes LSTM que combinan las arquitecturas propuestas para mejorar todos los parámetros del habla

Las bases de datos y la extracción de características utilizadas en este capítulo son semejantes a las descritas en las secciones 4.3 y 4.3.2. Por su parte, la cantidad de datos utilizados para entrenamiento, validación y prueba coinciden con los mostrados en la Tabla 4.1.

En esta ocasión se experimentó con redes LSTM cuya arquitectura consiste en tres capas, con ciento cincuenta unidades en cada una. Un total de cuarenta redes LSTM se entrenaron, diez *autoencoders* y veinte memorias auto-asociativas. Esta arquitectura fue seleccionada después de un proceso de prueba y error, en el cual se consideró la factibilidad de entrenar la cantidad de redes necesarias para todas las voces. Dado el tiempo que requiere el entrenamiento de una red LSTM, es limitada la cantidad de experimentación posible actualmente para establecer las arquitecturas más adecuadas.

En la Tabla 5.1 se muestra la nomenclatura de los algoritmos utilizados. Para efectos de comparación se entrenaron también los sistemas LSTM-1, LSTM-2 y LSTM-3 semejantes a los del Capítulo 4.

**Tabla 5.1:** Nomenclatura de los algoritmos de los sistemas híbridos para mejora de señales de voz HMM

| Algoritmo | Nomenclatura |
|---|---|
| Wiener | Filtro Wiener Adaptativo |
| LSTM-1 | Sistema de un solo *autoencoder* LSTM para mejorar los MFCC del habla generada con HTS |
| LSTM-2 | Colección de un *autoencoder* LSTM para mejorar los MFCC y una memoria auto-asociativa para mejorar el parámetro de energía del habla generada con HTS |
| LSTM-3 | Colección de un *autoencoder* LSTM para mejorar los MFCC y dos memorias auto-asociativas par mejorar de forma individual el parámetro de energía y de $f_0$ del habla generada con HTS |
| HW-LSTM-1 | Primer sistema híbrido, que contempla el procesamiento del habla HTS con el filtro Wiener y posteriormente la aplicación de LSTM-1 |
| HW-LSTM-2 | Sistema híbrido en el cual se aplica el filtro Wiener al habla HTS, y posteriormente LSTM-2 |
| HW-LSTM-3 | Sistema híbrido en el cual se aplica el filtro Wiener y posteriormente LSTM-3 |

Todas las redes consideradas requieren un entrenamiento individual y específico para cada una de las voces, así como un entrenamiento distinto si se trata de LSTM solamente o del sistema híbrido, en el cual se realiza el entrenamiento a partir de la salida del filtro Wiener. Los resultados y su análisis se muestran en la siguiente sección.

## 5.4 Resultados y discusión

El análisis de resultados se presenta en dos partes. En la primera se comparan las medidas aplicadas a los algoritmos propuestos, tanto en la colección de LSTM como en los sistemas híbridos y la salida del filtro Wiener. Esta comparación se realiza para determinar cuál de ellos obtiene los mejores resultados, y mediante un ANOVA establecer si los demás algoritmos no presentan diferencias estadísticamente significativas con respecto a éste.

En una segunda parte, se realiza una comparación de los algoritmos con la voz HTS, para determinar si tienen éxito en mejorar significativamente la calidad de ésta en cada una de las medidas utilizadas. Determinar cuál es el mejor de los algoritmos se realiza con la media de la evaluación en cincuenta frases del conjunto de prueba, en cada medida.

La mejora con significancia estadística se realiza con una Prueba HSD de Tukey [69]. Todas las pruebas estadísticas se realizaron con un nivel de significancia de 0.95, el cual es un estándar para el análisis estadístico comparativo.

### 5.4.1 Medidas objetivas

En el presente capítulo se aplicarán las medidas objetivas que se utilizan a lo largo de esta tesis: WSS, PESQ y $\text{SegSNR}_{\text{f}}$. Como en el Capítulo 4, se cuenta con las frases de habla sintetizada y natural alineadas, por lo que las medidas reflejarán la mejora obtenida. Adicionalmente, en este capítulo se incluirá la distancia media absoluta (*Mean Absolute Distance*, MAD) entre los coeficientes MFCC producidos por el sistema HTS, los post-filtros y la voz natural.

Los resultados de la medida WSS se muestran en la Tabla 5.2. Se puede observar cómo en tres de las cinco voces, los sistemas híbridos obtuvieron el mejor resultado, y en las otros dos, los resultados de estos sistemas no difieren significativamente del mejor. Se puede observar en esta misma tabla que el filtro Wiener por sí mismo no logró mejorar ninguna de las voces en esta medida en particular. Sin embargo, dado los resultados obtenidos con los sistemas híbridos, el mapeo desde las frases filtradas con Wiener tiende a tener mejores resultados que aquellos donde los LSTM fueron aplicados directamente.

Los resultados de la medida PESQ se muestran en la Tabla 5.3. Para esta medida objetiva, los sistemas híbridos obtuvieron mejores resultados en dos de las voces (JMK, RMS), mejor resultado que los LSTM simples en un caso (la voz BDL) y resultados sin diferencia significativa con el mejor para la voz SLT. Se destaca que el filtro Wiener aplicado de forma única a las voces

**Tabla 5.2:** Resultados de la medida WSS para los sistemas propuestos. Los datos están dispuestos por columnas. Los valores menores representan mejores resultados. * indica el mejor. En negrita las medidas que no difieren significativamente del mejor.

| | Voz | | | | |
|---|---|---|---|---|---|
| Sistema | BDL | CLB | JMK | RMS | SLT |
| HTS (Ninguno) | 43.32 | 37.20 | 34.96 | 37.03 | 48.17 |
| Wiener | 49.42 | 49.10 | 43.39 | 61.06 | 58.71 |
| LSTM-1 | **40.04** | **34.94** | **31.69** | **32.39** | 42.78* |
| LSTM-2 | **39.90** | **34.94** | 31.44* | **32.54** | **43.21** |
| LSTM-3 | **41.57** | 36.92 | 35.65 | 38.62 | 69.54 |
| HW-LSTM-1 | 38.87* | **33.81** | **31.70** | 31.98* | **46.19** |
| HW-LSTM-2 | **39.10** | 33.17* | **32.34** | **32.53** | **45.21** |
| HW-LSTM-3 | 41.65 | 38.72 | 51.68 | 40.84 | **49.50** |

obtuvo dos mejores resultados en esta medida, lo cual indica que en este caso sí es representativa la mejora en las señales producidas por este filtro, gracias a la eliminación del ruido. El único caso de voz en la cual no se obtuvo beneficio significativo con los sistemas híbridos fue CLB.

**Tabla 5.3:** Resultados del parámetro PESQ. Los datos están dispuestos por columnas. Los valores más altos representan mejores resultados. * indica el mejor. En negrita las medidas que no difieren significativamente del mejor.

| | Voz | | | | |
|---|---|---|---|---|---|
| Sistema | BDL | CLB | JMK | RMS | SLT |
| HTS (Ninguno) | 1.19 | 1.32 | 1.32 | 1.65 | 1.04 |
| Wiener | 1.43* | 1.33* | **1.31** | 1.08 | **1.04** |
| LSTM-1 | **1.32** | **1.18** | **1.41** | **1.59** | **1.03** |
| LSTM-2 | **1.32** | **1.18** | **1.45** | **1.54** | 1.05* |
| LSTM-3 | 1.19 | 1.09 | **1.25** | **1.45** | 0.62 |
| HW-LSTM-1 | **1.26** | 1.04 | 1.46* | 1.61* | **0.98** |
| HW-LSTM-2 | **1.34** | 1.05 | **1.37** | **1.60** | **1.02** |
| HW-LSTM-3 | 1.16 | 0.91 | 0.84 | **1.59** | **1.01** |

Los resultados de la medida $\text{SegSNR}_{\text{f}}$ para las redes LSTM, el filtro Wiener y la propuesta de sistemas híbridos se muestran en la Tabla 5.4. Éstos son semejantes a los obtenidos con las medidas previas: Los sistemas híbridos presentan mejores resultados en tres de las cinco voces, y sin diferencias significativas con el mejor en las otras dos. Tal como se presentó para el caso de WSS, el filtro Wiener aplicado de forma única no parece beneficiar las voces producidas con HTS, pero permiten que las redes LSTM obtengan mejores resultados en los sistemas híbridos.

**Tabla 5.4:** Resultados de la medida $\text{SegSNR}_\text{f}$. Los datos están dispuestos por columnas. Los valores más altos representan mejores resultados. * es el mejor. En negrita las medidas que no difieren significativamente del mejor.

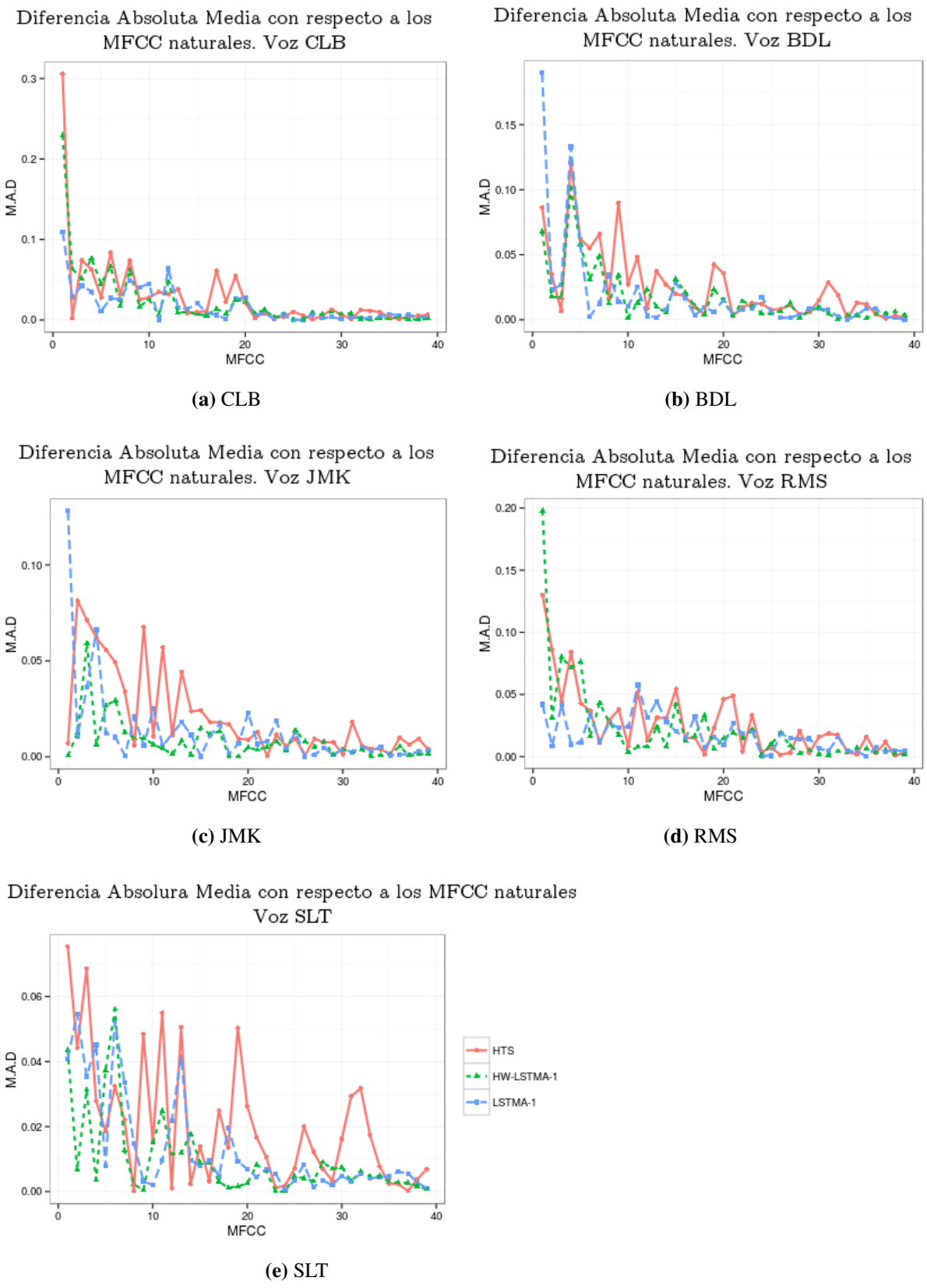
| | Voz | | | | |
|---|---|---|---|---|---|
| Sistema | BDL | CLB | JMK | RMS | SLT |
| HTS (Ninguno) | 0.46 | 2.46 | 1.50 | 1.57 | 0.70 |
| Wiener | -1.53 | 1.17 | -1.25 | -0.09 | -1.54 |
| LSTMA-1 | **1.12** | 2.73 | **1.91** | **2.41** | **1.21** |
| LSTMA-2 | **1.08** | 2.21 | **1.11** | 2.52* | 1.72* |
| LSTMA-3 | **1.05** | 2.38 | 0.80 | 1.42 | 0.30 |
| HW-LSTMA-1 | 1.23* | **3.36** | **1.87** | 1.20 | **1.60** |
| HW-LSTMA-2 | 0.60 | 3.79* | 1.99* | **1.62** | **1.20** |
| HW-LSTMA-3 | 0.17 | 2.78 | 0.33 | 1.19 | **1.16** |

Finalmente, la diferencia media absoluta (MAD) entre los MFCC de los sistemas híbridos y los post-filtros basados en LSTM se presentan en la Figura 5.3, comparados también con aquellos de las voces HTS sin ningún post-filtro. Para esta comparación se eligió LSTM-1 y HW-LSTM-1, ya que ambos modifican únicamente los coeficientes MFCC.

De acuerdo con los datos representados en estas gráficas, la propuesta híbrida HW-LSTM-1 presentada en este capítulo permite una mejor aproximación a los MFCC de la voz natural en 28 de los 39 coeficientes para la voz STL (un 72 % de los casos), en comparación con la mejora de 26 de 39 (67 % de los casos) coeficientes que logra el sistema LSTM-1. Resultados similares se obtienen en las otras voces, con la excepción de CLB, en la cual tanto el sistema LSTM-1 como el híbrido HW-LSTM-1 mejoran 25 de 39 coeficientes MFCC (64 % de los casos). La mejora más significativa obtenida con el híbrido HW-LSTM-1 se encuentra en la voz JMK, donde 31 de 39 (79 % de éstos) coeficientes obtienen mejora con respecto a la voz HTS, en comparación con 27 coeficientes mejorados por LSTM-1 (69 % de los casos).

A partir de estos resultados, se observa que el sistema híbrido de post-filtros Wiener-LSTM beneficia a algunas voces de forma más significativa que a otras. Una explicación de estos hechos es que el filtro Wiener aplicado como primera etapa puede afectar negativamente los MFCC si la voz contiene algunas características particulares. Como se muestra en la figura 5.4, las voces femeninas (SLT, CLB) tienen valores de $f_0$ notablemente mayores que las masculinas. En particular, la voz RMS presenta valores menores de este parámetro que el resto de las voces, y el filtro Wiener puede afectar de forma considerablemente distinta a las voces de acuerdo con el rango de frecuencias donde se ubique su rango de $f_0$, el cual puede entrar dentro de la región de frecuencias del ruido contenido en las voces sintetizadas.

La Figura 5.5 presenta el impacto del filtro Wiener en los MFCC de las voces con los rangos más graves y más agudos del parámetro $f_0$. Es notorio cómo el filtro Wiener impacta positivamente




**(a)** CLB

**(b)** BDL

**(c)** JMK

**(d)** RMS

**(e)** SLT

**Figura 5.3:** Comparación de las diferencias absolutas medias entre los MFCC de los distintos algoritmos y la voz natural.

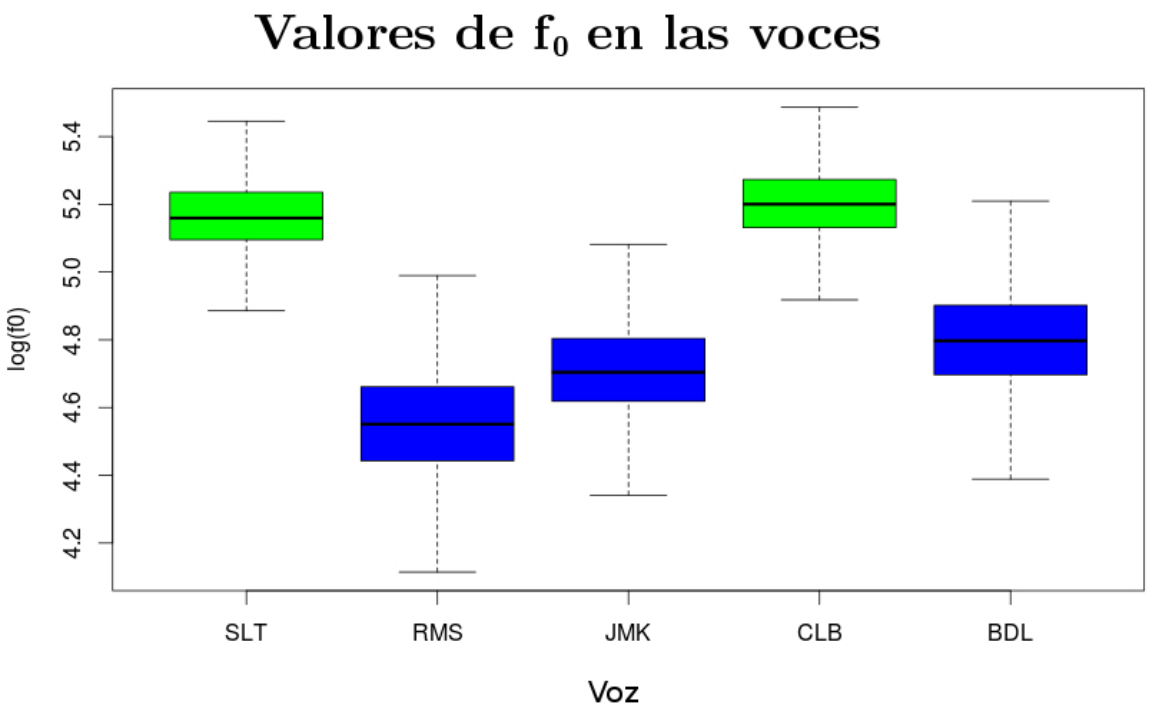



**Figura 5.4:** Diagramas de caja de $log(f_0)$ para las cinco voces consideradas. SLT y CLB son voces femeninas, mientras que RMS, JMK y BDL son masculinas.

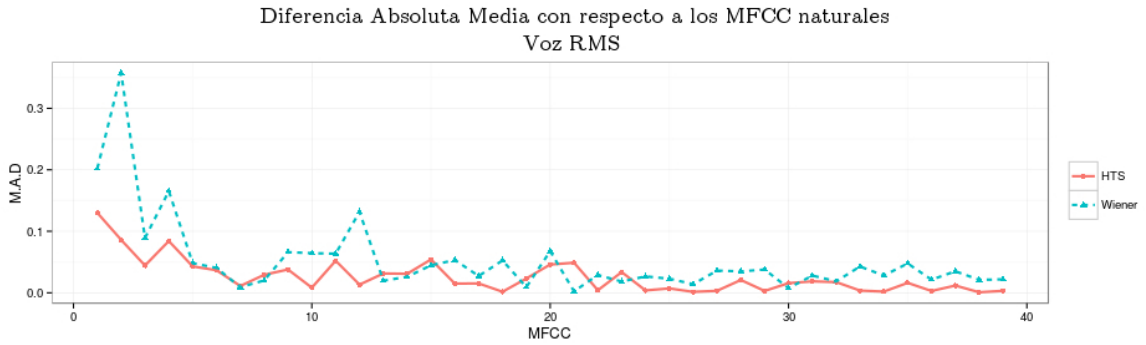
los MFCC de la voz CLB, mientras que impacta negativamente los MFCC de la voz RMS.

Dado que el filtro Wiener es el primer paso del sistema híbrido, las características de la voz y la observación de la manera en que el filtro Wiener afecta los MFCC y los demás parámetros puede ayudar a decidir en qué casos los sistemas híbridos de post-filtros pueden beneficiar las voces.

### 5.4.2 Mejora estadísticamente significativa del habla sintetizada

En esta sección se presenta el análisis estadístico realizado con el propósito de determinar si los resultados mostrados previamente mejoran significativamente los de las voces sintetizadas con HTS. La razón de realizarlo es el hecho de que si bien un sistema de post-filtros puede presentar un resultado significativamente mejor que los demás, aún en ese caso el resultado puede no ser significativamente mejor que el de la voz sintetizada.

Para este análisis estadístico se aplicó la Prueba HSD de Tukey, la cual realiza comparaciones




**(a)** RMS


**(b)** CLB

**Figura 5.5:** Comparación de la diferencia absoluta media de los MFCC en las voces con mayores y menores valores de log $f_0$.

entre pares de conjuntos de datos, en este caso entre las medidas de las evaluaciones objetivas de la voz HTS y las de los sistemas de post-filtros. En las tablas 5.7 a 5.7 se reporta cuáles de los sistemas realizan una mejora significativa.

En la Tabla 5.7 se observa que los sistemas híbridos mejoran todas las voces HTS para la medida WSS, y de forma significativa en tres de los casos. Entre todos los sistemas, el que mejor resultados presenta para esta medida es el híbrido HW-LSTM-2.

Para el caos de PESQ, en la Tabla 5.6 se muestra que ninguno de los algoritmos mejora significativamente esta medida con respecto a las voces de HTS. Sin embargo, los sistemas LSTM-1 y LSTM-2 mejoran el valor medio de esta medida en dos casos, al igual que lo hacen los híbridos HW-LSTM-1 y HW-LSTM-2. Por su parte, el filtro Wiener realiza mejoras que no son estadísticamente significativas en tres voces: BDL, CLB, SLT.

Finalmente, para la medida $\text{SegSNR}_\text{f}$, se observa que el híbrido HW-LSTMA-1 mejora significativamente dos de las voces y de forma no significativa otras dos, dejando únicamente sin mejora a RMS, es decir, la voz con el rango más grave de $f_0$. Por su parte, el híbrido HW-LSTM-2

mejora significativamente dos voces, y el resto no. Para esta medida, junto con LSTM-1 es el sistema que muestra su efectividad, de forma significativa o no, para todas las voces.

**Tabla 5.5:** Resultados de la medida WSS. ✓ indica que la mejora es significativa. "ns" representa que el valor de la medida no difiere significativamente de la voz HTS. Los espacios en blanco indican que no se obtuvo beneficio con el procedimiento.

| | Voz | | | | |
|---|---|---|---|---|---|
| Sistema | BDL | CLB | JMK | RMS | SLT |
| Wiener | | | | | |
| LSTM-1 | ns | ns | ✓ | ns | ns |
| LSTM-2 | ns | ns | ✓ | ✓ | ns |
| LSTM-3 | ns | ns | | ✓ | |
| HW-LSTM-1 | ✓ | ns | ✓ | ✓ | ns |
| HW-LSTM-2 | ✓ | ✓ | ns | ✓ | ns |
| HW-LSTM-3 | ns | | | | ns |

**Tabla 5.6:** Resultados de la medida PESQ. ✓ indica que la mejora es significativa. "ns" representa que el valor de la medida no difiere significativamente de la voz HTS. Los espacios en blanco indican que no se obtuvo beneficio con el procedimiento.

| | Voz | | | | |
|---|---|---|---|---|---|
| Sistema | BDL | CLB | JMK | RMS | SLT |
| Wiener | ns | ns | | | ns |
| LSTM-1 | ns | | ns | | ns |
| LSTM-2 | ns | | ns | | ns |
| LSTM-3 | | | | | |
| HW-LSTM-1 | ns | | ns | | |
| HW-LSTM-2 | ns | | ns | | |
| HW-LSTM-3 | | | | | |

De acuerdo con estos resultados se observa que los sistemas híbridos obtienen mejores valores mayor cantidad de veces que el filtro Wiener solo y que los sistemas LSTM-1, LSTM-2 y LSTM-3. En particular el híbrido HW-LSTM-2 es el que muestra mayor cantidad de mejoras en las medidas objetivas.

Por su parte, la comparación de la distancia absoluta media muestra que los sistemas híbridos también presentan mejores resultados que los sistemas no híbridos en cuanto a mejorar los coeficientes MFCC.

**Tabla 5.7:** Resultados de la medida $\text{SegSNR}_{\text{f}}$. ✓ indica que la mejora es significativa. "ns" representa que el valor de la medida no difiere significativamente de la voz HTS. Los espacios en blanco indican que no se obtuvo beneficio con el procedimiento.

| | Voz | | | | |
|---|---|---|---|---|---|
| Sistema | BDL | CLB | JMK | RMS | SLT |
| Wiener | | | | | |
| LSTM-1 | ✓ | ns | ns | ✓ | ns |
| LSTM-2 | ns | | | ✓ | ✓ |
| LSTM-3 | ns | | | | |
| HW-LSTM-1 | ✓ | ✓ | ns | | ns |
| HW-LSTM-2 | ns | ✓ | ns | ns | ns |
| HW-LSTM-3 | | ns | | | ns |

## 5.5 Resumen de contribuciones

En este capítulo se ha presentado una propuesta de un sistema híbrido para mejorar el resultado de las voces generadas con el sistema HTS. Este se ha basado en filtros Wiener y colecciones de redes LSTM, con arquitectura de *autoencoders* y memorias auto-asociativas. Se realizó una extensa comparación utilizando cinco voces, tanto masculinas como femeninas, en idioma inglés. Para la evaluación se utilizó, además de las tres medidas consideradas en capítulos anteriores, la media de las diferencias absolutas entre los valores de los distintos sistemas y la voz natural.

Tanto la medición de estas diferencias absolutas entre coeficientes como los resultados de las pruebas estadísticas realizadas para determinar significancia estadística, muestran mejoras representativas de los sistemas híbridos en varias de las medidas. Destacan la entre los coeficientes MFCC y las medidas WSS y $\text{SegSNR}_{\text{f}}$.

La principal consideración para aplicar el sistema híbrido proviene de la observación de ruido de baja intensidad en las voces generadas con el sistema HTS. El filtro Wiener mejora la señal al reducir este componente ruidoso, lo cual se ha reflejado principalmente en las medidas de PESQ, mientras que las colecciones de redes LSTM tienen mejores posibilidades de realizar el mapeo de estos parámetros filtrados hacia los de la voz natural. Con respecto a la propuesta del Capítulo 4, la propuesta actual elimina el requerimiento de reducción de ruido en los LSTM, es decir, se reduce la complejidad en la regresión requerida en los post-filtros.

Los resultados muestran como los sistemas híbridos benefician varias de las voces en todas las medidas utilizadas. Las mejoras no están presentes en algunas de las voces con características particulares. Por ejemplo, en las voces con los valores menores de $f_0$, en los cuales el filtro

Wiener, además de eliminar el componente de ruido, distorsiona los MFCC en mayor medida que las otras voces.

Este enfoque híbrido propuesto puede abrir nuevas posibilidades para mejorar voces artificiales, combinando múltiples etapas o seleccionando los mejores parámetros de cada una.

# 6

# POST-FILTROS DISCRIMINATIVOS

*En este capítulo se plantea la tercera propuesta de utilización de post-filtros múltiples para mejorar los resultados de la síntesis de voz basada en HMM, aplicándolos de forma discriminativa a segmentos sonoros y no sonoros del habla sintetizada.*

## Índice

## 6.1 Introducción

En este capítulo se propone una extensión a la aplicación de los post-filtros basados en LSTM presentados en el Capítulo 4, hacia un enfoque discriminativo. Previamente se realizó un proceso de aprendizaje con una colección de distintas arquitecturas LSTM, para aprender el problema de regresión entre habla natural y sintetizada.

Este proceso de aprendizaje se llevó a cabo sobre las frases completas, pretendiendo estimar con una única red la función que permita mejorar subconjuntos de parámetros del habla ($f_0$, energía o MFCC). En el presente capítulo, la propuesta es diferenciar aquellos segmentos de las frases que corresponden a la generación de fonemas sonoros (con $f_0 > 0$) y no sonoros (con $f_0 = 0$).

El realizar esta diferenciación constituye un refinamiento del procedimiento, ya que se cuenta con una separación de sonidos de acuerdo con su naturaleza, y de forma independiente se pueden atacar los problemas que afectan a cada uno. La intención es comparar este procedimiento discriminativo con la aplicación de los post-filtros en toda la frase realizados en el Capítulo 4.

## 6.2 Sistema propuesto

Para solventar los problemas de calidad propios de la generación de voces artificiales a partir de HMM, los post-filtros presentados en los capítulos 4 y 5 constituyen un aporte para la obtención de parámetros $\overline{R_Y}$ que se asemejen más a los correspondientes $R_X$ de la voz natural, a partir de un aprendizaje directo de los datos del habla sintetizada $R_Y$. Estos post-filtros se implementaron combinando distintos filtros y arquitecturas de redes LSTM para aplicar una función de regresión a todos los fonemas de una frase.

En la presente propuesta, se consideran todos los fonemas presentes en la base de datos, y se realiza un agrupamiento de éstos en dos conjuntos mutuamente excluyentes, los cuales corresponden a los fonemas sonoros y no sonoros. A partir de esta separación, se definen y entrenan post-filtros para dos procesos de regresión separados, con funciones $f_{21}$ y $f_{22}$, las cuales deben mapear las características del grupo correspondiente de fonemas artificiales hacia los naturales.

El proceso se ilustra en la Figura 6.1, en la cual, en un primer nivel, los post-filtros aproximan una única función de regresión $f_{11}$ para mapear los parámetros (ya sea energía, $f_0$, o MFCC) de todos los fonemas sintetizados hacia los naturales. En el enfoque discriminativo, dos funciones

independientes, $f_{21}$ y $f_{22}$ se utilizan para mapear los parámetros del conjunto 1 del habla sintetizada hacia el conjunto 1 del habla natural, y de forma semejante con el conjunto 2.

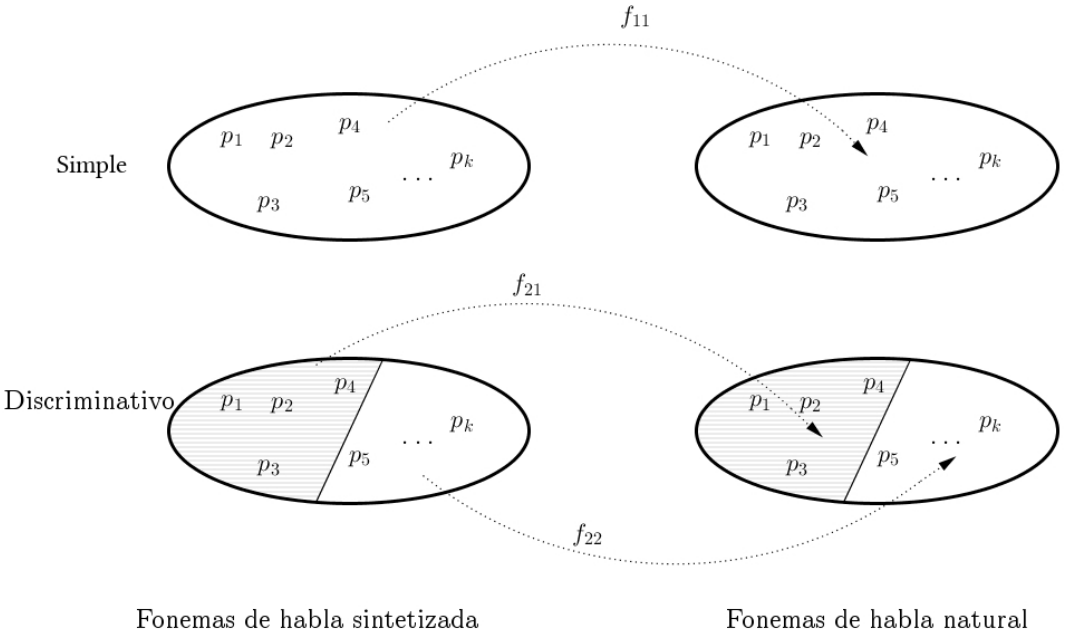
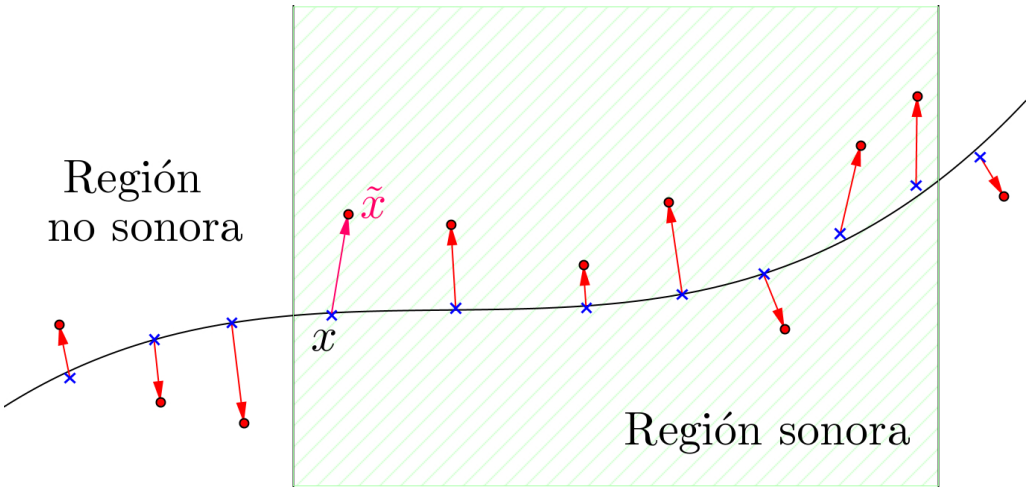



**Figura 6.1:** Ilustración del proceso de agrupamiento y aplicación discriminativa de funciones de regresión entre parámetros de habla sintetizada y natural.

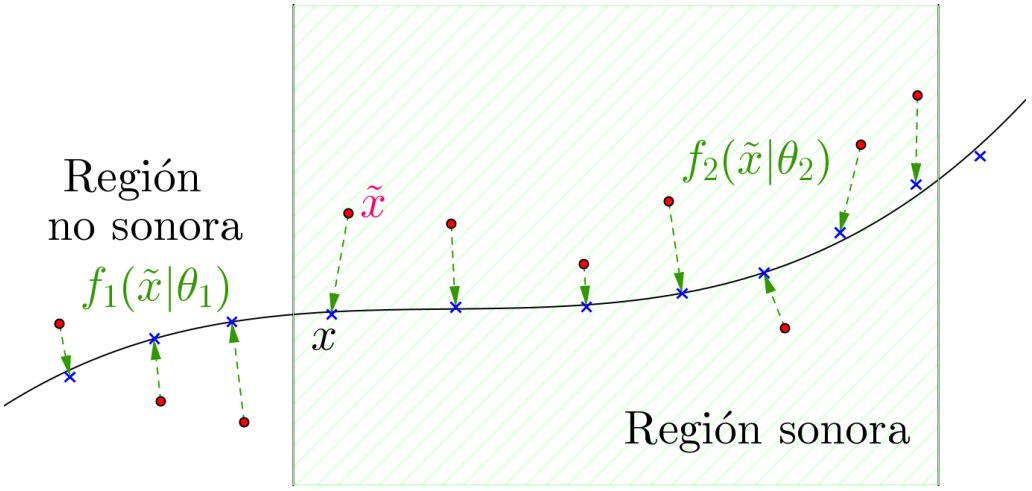
El proceso de agrupamiento y regresión con el cual se entrena cada conjunto y sus correspondientes post-filtros, se aplica finalmente a un conjunto de frases de prueba, para determinar la mejora lograda con el proceso discriminativo en comparación con el proceso regular, en términos de medidas objetivas y subjetivas.

Para el proceso de regresión, tal como se ha realizado a lo largo de la tesis, se utilizarán *autoencoders* y memorias auto-asociativas con unidades LSTM, de acuerdo con los parámetros que se deseen mejorar. Cada conjunto de estas redes se aplicará de acuerdo con la naturaleza de los parámetros en cada frase. En la Figura 6.2 se observa cómo se puede establecer una diferencia entre las perturbaciones de los parámetros del habla dependiendo de los sonidos que se están produciendo, distinguiendo entre sonoros (como las vocales, la <m>, <n>, <l>) y no sonoros (como la <p>, <t>, <s>). Luego de entrenar las redes específicas para cada una de estas regiones, el mapeo se realiza aplicando la red correspondiente a cada una de las regiones de la frase, como se muestra en la Figura 6.2.

En este capítulo se utilizarán nuevamente frases de habla natural y sintetizada que se encuentran alineadas ventana a ventana, de manera que se puede establecer una correspondencia entre cada vector de parámetros extraído en cada una, ya sea que corresponda a una región sonora o no




**(a)** Representación gráfica de una variedad diferenciable cerca de la cual se encuentran los puntos $x$. Se hace diferencia de las regiones sonoras y no sonoras.




**(b)** El *autoencoder* aprende un mapeo de los puntos de habla artificial hacia la variedad diferenciable. Este mapeo es discriminativo pues se estima de acuerdo con la región del habla.

**Figura 6.2:** Representación gráfica del proceso de post-filtro discriminativo. $\tilde{x}$ es la versión distorsionada del parámetro $x$ y $\theta$ lo correspondiente a la red LSTM para región sonora o no sonora.

sonora. Dada cada una de estas ventanas, se extrae un vector que contiene un coeficiente para $f_0$, un coeficiente para energía y 39 coeficientes MFCC, utilizando el sistema Ahocoder.

Después de la parametrización, se separan los vectores tanto del habla natural como sintetizada en sonoros (con valores de $f_0 > 0$) y no sonoros (con valores de $f = 0$), a partir de la estimación de este parámetro realizada con el sistema Ahocoder. Para cada uno de estos conjuntos, se entrenan las colecciones de redes LSTM para mejorar los parámetros de forma independiente.

En cada una de las regiones definidas aplicamos tres tipos de post-filtros:

- LSTM-1 Discriminativo: Las entradas de la red LSTM, con arquitectura de *autoencoder* consistirán en los parámetros MFCC de la versión sintetizada, distinguiendo entre sonora y no sonora, mientras que las salidas serán los correspondientes MFCC del habla natural. El coeficiente de energía y el de $f_0$ se tomarán sin cambios de los generados con HTS. El procedimiento de entrenamiento se muestra en el Algoritmo 5.
- LSTM-2 Discriminativo: Considera un *autoencoder* entrenado con las condiciones descritas en LSTM-1, pero adicionalmente incluye dos memorias auto-asociativa para mejorar el parámetro de energía proveniente de la voz HTS, una para la parte sonora y otra para la parte no sonora. El proceso de entrenamiento se muestra en el Algoritmo 6.
- LSTM-3: Incluye una memoria auto-asociativa adicional a las indicadas en LSTM-2, la cual se entrena y opera de forma semejante para el parámetro $f_0$. Se destaca que no es necesario realizar ningún procedimiento adicional para el coeficiente $f_0$ en las regiones no sonoras, pues permanece con su valor $f_0 = 0$. El proceso de entrenamiento se muestra en el Algoritmo 7.

**Algoritmo 5** Entrenamiento de post-filtros discriminativos LSTM para mejorar el coeficiente energía en las frases de HTS

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: frases de habla HTS (alineadas)
**Entrada:** $L$: red inicializada
**Entrada:** $N$: número de iteraciones
**Entrada:** $K$: número de ventanas
**Salida:** $L_{ae_1}$: Red entrenada para mejorar parámetro MFCC en segmentos sonoros
**Salida:** $L_{ae_2}$: Red entrenada para mejorar parámetro MFCC en segmentos no sonoros

```
mientras iteración < N hacer
  mientras ventana < K hacer
    extraer características: M1: 39 MFCC de n y M2: 39 MFCC de r
    // propagar 39 MFCC de segmentos sonoros r de la entrada hacia la salida de L (ecuaciones 2.8 a 2.16) //
    si f0 > 0 entonces
      calcular fL(M2)
      calcular error: E(fL(M2), M1)
      si error < error mínimo entonces
        ajustar pesos de L (ecuaciones 2.17 a 2.22)
        Lae1 ← L
      fin si
    fin si
    si f0 = 0 entonces
      calcular fL(M2)
      calcular error: E(fL(M2), M1)
      si error < error mínimo entonces
        ajustar pesos de L (ecuaciones 2.17 a 2.22)
        Lae2 ← L
      fin si
    fin si
  fin mientras
fin mientras
devolver Lae1, Lae2
```

**Algoritmo 6** Entrenamiento de post-filtros discriminativos LSTM para mejorar el coeficiente energía en las frases de HTS

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: frases de habla HTS (alineadas)
**Entrada:** $L$: red inicializada
**Entrada:** $N$: número de iteraciones
**Entrada:** $K$: número de ventanas
**Salida:** $L^1_{maa_2}$: Red entrenada para mejorar parámetro de energía en segmentos sonoros
**Salida:** $L^2_{maa_2}$: Red entrenada para mejorar parámetro de energía en segmentos no sonoros

```
mientras iteración < N hacer
  mientras ventana < K hacer
    extraer características: M1: 39 MFCC y energía de n y M2: 39 MFCC de n y energía de r

    // propagar 39 MFCC y energía de segmentos sonoros r de la entrada hacia la salida de
    L (ecuaciones 2.8 a 2.16) //
    si f0 > 0 entonces
      calcular fL(M2)
      calcular error: E(fL(M2), M1)
      si error < error mínimo entonces
        ajustar pesos de L (ecuaciones 2.17 a 2.22)
        L^1_maa2 ← L
      fin si
    fin si
    si f0 = 0 entonces
      calcular fL(M2)
      calcular error: E(fL(M2), M1)
      si error < error mínimo entonces
        ajustar pesos de L (ecuaciones 2.17 a 2.22)
        L^2_maa2 ← L
      fin si
    fin si
  fin mientras
fin mientras
devolver L^1_maa2, L^2_maa2
```

**Algoritmo 7** Entrenamiento de post-filtros discriminativos LSTM para mejorar el coeficiente $f_0$ en las frases de HTS

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: frases de habla HTS (alineadas)
**Entrada:** $L$: red inicializada
**Entrada:** $N$: número de iteraciones
**Entrada:** $K$: número de ventanas
**Salida:** $L^1_{maa_1}$: Red entrenada para mejorar parámetro $f_0$

```
mientras iteración < N hacer
  mientras ventana < K hacer
    extraer características: M1: 39 MFCC y f0 de n y M2: 39 MFCC de n y f0 de r
    // propagar 39 MFCC y f0 de segmentos sonoros r de la entrada hacia la salida de L
    (ecuaciones 2.8 a 2.16) //
    si f0 > 0 entonces
      calcular fL(M2)
      calcular error: E(fL(M2), M1)
      si error < error mínimo entonces
        ajustar pesos de L (ecuaciones 2.17 a 2.22)
        L^1_maa1 ← L
      fin si
    fin si
    si f0 = 0 entonces
      f0 = 0
    fin si
  fin mientras
fin mientras
devolver L^1_maa1
```

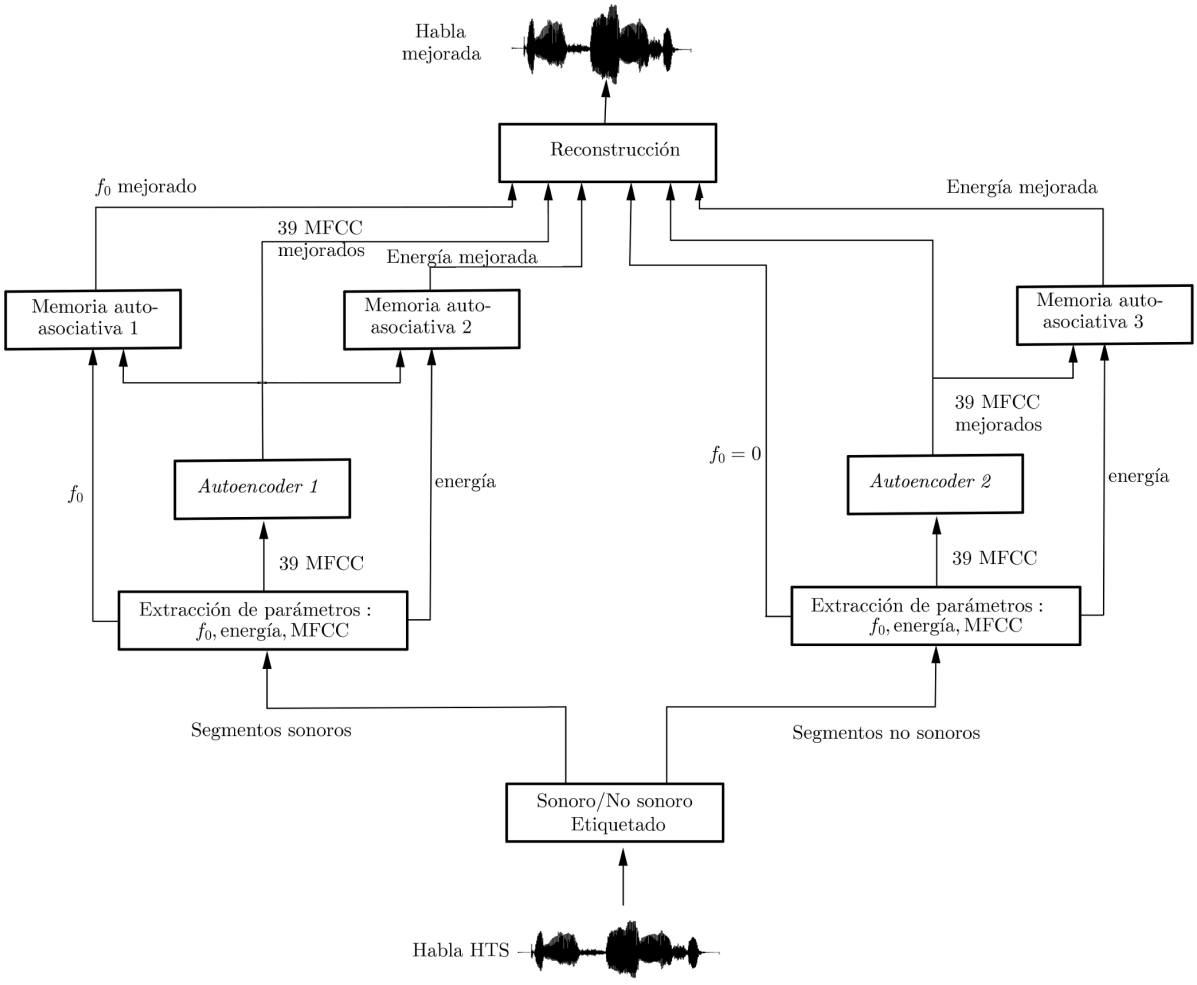



**Figura 6.3:** Sistema propuesto para aplicar post-filtros discriminativos a voces HTS

Cada conjunto de post-filtros aproxima la función de regresión de la región particular de la frase. La intención de discriminar los segmentos es permitir una mejora en la señal con respecto al enfoque presentado en los capítulos precedentes, ya que la red modelará más específicamente la regresión requerida en sonidos que tienen naturaleza distinta. La Figura 6.3 esquematiza el sistema propuesto para el caso en que se mejoran los tres tipos de parámetros.

Una parte del proceso necesaria para poder reconstruir adecuadamente la forma de onda de habla mejorada, es el etiquetado de cada ventana con un número de secuencia, el cual se realiza de forma previa a la aplicación de los post-filtros. Con este etiquetado es posible reordenar los parámetros al finalizar el proceso, ya que se reciben de forma independiente aquellos que corresponden a regiones sonoras y no sonoras. En el Algoritmo 8 se describe este procedimiento.

Las bases de datos y la extracción de características utilizadas en este capítulo son semejantes a las descritas en las secciones 4.3 y 4.3.2. Por su parte, la cantidad de datos utilizados para entrenamiento, validación y prueba coinciden con los mostrados en la Tabla 4.1. En cuanto a la arquitectura de los *autoencoders* y los memorias auto-asociativas, las características se conservaron según lo presentado en la Sección 5.3.

**Algoritmo 8** Mejora de frases HTS con los post-filtros de $f_0$, energía y MFCC

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: Frases de habla HTS (alineadas)
**Entrada:** $L_{ae_1}$, $L_{ae_2}$: *a*utoencoders LSTM entrenados para 39 MFCC sonoros y no sonoros
**Entrada:** $L^1_{maa_1}$: memoria auto-asociativas entrenada para $f_0$ de segmentos sonoros.
**Entrada:** $L^1_{maa_2}$, $L^2_{maa_2}$: memorias auto-asociativas entrenada para energía de segmentos sonoros y no sonoros.
**Salida:** Onda procesada con post-filtros

```
mientras exista frase hacer
    extraer características: f0, e: energía, g: 39 MFCC de r
    numerar vectores
    // predecir parámetros sonoros //
    si f0 > 0 entonces
        P^s_g = f_{L_ae1}(g)
        P^s_f0 = f^1_{L_maa1}(f0, P^s_g)
        P^s_e = f^1_{L_maa2}(e)
    fin si
    // predecir parámetros no sonoros //
    si f0 = 0 entonces
        P^ns_g = f_{L_ae2}(g)
        P^ns_f0 = 0
        P^ns_e = f^2_{L_maa2}(e)
    fin si
fin mientras
// Unir y ordenar vectores de acuerdo con numeración //
Reconstruir habla con P_f0, P_g, P_e // (f0 mejorada, energía mejorada, 39 MFCC mejorados) //
devolver Onda de habla reconstruida
```

## 6.3 Resultados y discusión

Las medidas utilizadas para evaluar los resultados coinciden con las utilizadas en la Sección 5.3.1, y se añade una evaluación subjetiva de percepción de naturalidad.

Para la presentación de los resultados y su análisis, se utilizará la siguiente nomenclatura:

- HTS: La voz sintetizada producida con el sistema HTS, basada en HMM
- LSTM-1: Sistema de un solo *autoencoder* LSTM para mejorar los MFCC del habla generada con HTS
- LSTM-2: Colección de un *autoencoder* LSTM para mejorar los MFCC y una memoria auto-asociativa para mejorar el parámetro de energía del habla generada con HTS
- LSTM-3: Colección de un *autoencoder* LSTM para mejorar los MFCC y dos memorias auto-asociativas par mejorar de forma individual el parámetro de energía y de $f_0$ del habla generada con HTS
- LSTM-1 Discriminativo: Sistema de dos *autoencoders* LSTM para mejorar de forma independiente los coeficientes MFCC de los segmentos sonoros y no sonoros del habla generada con HTS
- LSTM-2 Discriminativo: Colección de dos *autoencoders* LSTM para mejorar los MFCC y dos memorias auto-asociativas para mejorar el parámetro de energía del habla generada con HTS, de forma independiente en ambos casos para los segmentos sonoros y no sonoros
- LSTM-3 Discriminativo: Semejante al anterior, con una memoria auto-asociativa adicional para la mejora de los segmentos sonoros del habla generada con HTS

### 6.3.1 Medidas objetivas

Los resultados para la medida WSS se muestran en la Tabla 6.1. Se puede observar cómo en cuatro de las cinco voces, los mejores resultados se obtuvieron con los post-filtros discriminativos, mientras que en el caso restante (la voz BDL), el resultado de los tres post-filtros discriminativos no difieren significativamente del mejor.

Se destaca también en estos resultados, que para la medida WSS, el post-filtro Discriminativo LSTM-3 para la voz SLT tiene el mejor resultado, y ninguno de los otros casos tiene resultados semejantes. Esto indica que en esta ocasión la mejora del parámetro $f_0$, el cual se realiza exclusivamente en segmentos que cumplen $f_0 > 0$, es un aporte significativo a la mejora de este parámetro.

Resultados semejantes a los anteriores se obtuvieron para la medida PESQ, como se muestra en la Tabla 6.2, donde los post-filtros discriminativos obtuvieron los mejores resultados o resultados

**Tabla 6.1:** Resultados de la medida WSS para los sistemas discriminativos propuestos. Los valores menores representan mejores resultados. * indica el mejor. En negrita las medidas que no difieren significativamente del mejor.

| | | Simple | | | Discriminativo | | |
|---|---|---|---|---|---|---|---|
| Voz | HTS | LSTM-1 | LSTM-2 | LSTM-3 | LSTM-1 | LSTM-2 | LSTM-3 |
| SLT | 46.30 | 42.78 | 43.21 | 69.54 | 42.18 | 41.97 | 33.84* |
| RMS | 38.30 | **32.39** | **32.54** | 38.62 | 30.76* | **31.39** | **31.45** |
| JMK | 35.26 | **31.69** | **31.45** | 35.65 | 30.50* | **31.18** | **32.10** |
| CLB | 37.20 | **34.96** | **34.94** | 36.92 | **32.55** | 32.23* | 36.61 |
| BDL | 41.71 | **37.20** | 37.09* | 41.59 | **37.82** | **37.60** | **38.72** |

que no difieren del mejor en todos los casos. Para la voz SLT, el post-filtro discriminativo LSTM-3 obtuvo el mejor resultado, por encima de todos los demás sistemas.

**Tabla 6.2:** Resultados de la medida PESQ para los sistemas propuestos. Los valores mayores representan mejores resultados. * indica el mejor. En negrita las medidas que no difieren significativamente del mejor.

| | | Simple | | | Discriminativo | | |
|---|---|---|---|---|---|---|---|
| Voz | HTS | LSTM-1 | LSTM-2 | LSTM-3 | LSTM-1 | LSTM-2 | LSTM-3 |
| SLT | 1.0 | 1.0 | 1.0 | 0.6 | 1.0 | 1.0 | 1.3* |
| RMS | 1.5 | **1.6*** | **1.5** | **1.4** | 1.6* | **1.4** | **1.4** |
| JMK | 1.3 | 1.4* | 1.4* | **1.2** | **1.3** | **1.2** | **1.2** |
| CLB | 1.3 | 1.2* | 1.2* | **1.1** | 1.2* | **1.1** | **1.0** |
| BDL | 1.4 | 1.4* | 1.4* | 1.1 | 1.4* | 1.4* | **1.3** |

Los resultados para la medida $\text{SegSNR}_\text{f}$ se muestran en la Tabla 6.3, donde nuevamente para tres de las cinco voces los resultados obtenidos con los post-filtros discriminativos son los mejores, mientras que para los dos casos restantes (RMS y BDL) los resultados no difieren del mejor.

**Tabla 6.3:** Resultados de la medida $\text{SegSNR}_\text{f}$ para los sistemas propuestos. Los valores menores representan mejores resultados. * indica el mejor. En negrita las medidas que no difieren significativamente del mejor.

| | | Simple | | | Discriminativo | | |
|---|---|---|---|---|---|---|---|
| Voz | HTS | LSTM-1 | LSTM-2 | LSTM-3 | LSTM-1 | LSTM-2 | LSTM-3 |
| SLT | 0.5 | 1.2 | 1.7 | 0.3 | 1.5 | 1.8 | 2.8* |
| RMS | 1.4 | **2.4** | 2.5* | 1.4 | **2.2** | **2.0** | 1.8 |
| JMK | 1.7 | **1.9** | 1.1 | 0.8 | **2.0** | 2.1* | **2.0** |
| CLB | 2.4 | 2.7 | 2.2 | 2.4 | 3.4* | **3.1** | **2.8** |
| BDL | 0.5 | **1.4** | 1.5* | 0.7 | **1.3** | **1.3** | **1.2** |

En los anteriores resultados es claro que los post-filtros discriminativos tienen los mejores resultados para la gran mayoría de los casos y las medidas objetivas, mientras que para el resto, los resultados no difieren significativamente del mejor, lo cual muestra los beneficios de aplicar el sistema discriminativo, en comparación con el enfoque no discriminativo. En particular, se destaca cómo, a diferencia de los post-filtros no discriminativos, la mejora en el parámetro $f_0$ con el caso LSTM-3 se encuentra entre los mejores resultados en este capítulo. Esto se puede explicar por el hecho de que el mapeo entre este parámetro de la voz HTS y la natural se realiza entre valores positivos, en lugar de segmentos sonoros y no sonoros como en lo presentado en los capítulos anteriores.

En la Figura 6.4 se ilustran los resultados de la Media de la diferencia absoluta para los MFCC de los post-filtros discriminativos y los simples. Ambos se comparan con los MFCC de la voz HTS.

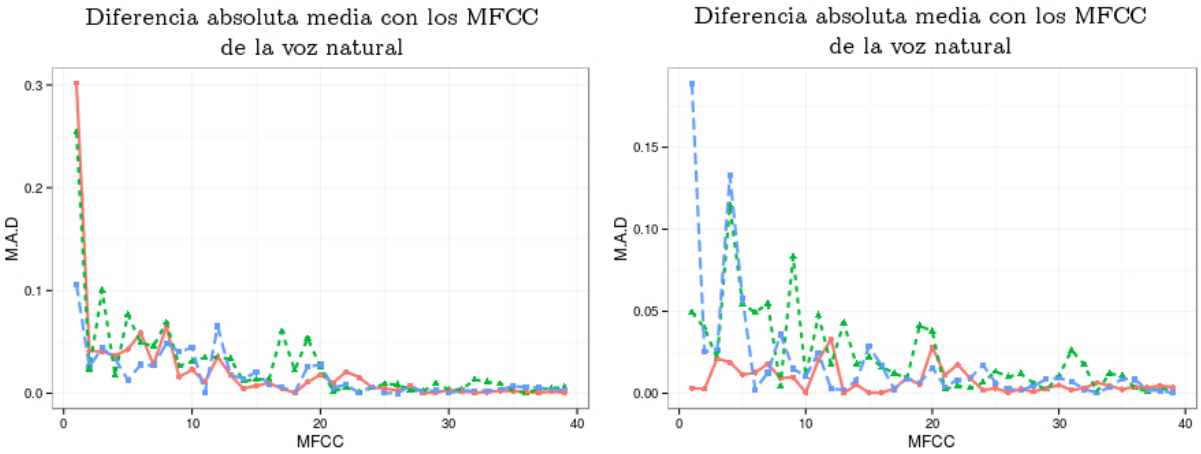
Esta comparación se realiza para el sistema LSTM-1 y el sistema LSTM-1 discriminativo, ya que ambos son entrenados para mejorar exclusivamente los coeficientes MFCC. Para el caso de la voz SLT, el post-filtro discriminativo realiza una mejora en 20 de los 39 coeficientes con respecto al no discriminativo es decir 51.3 % de los casos. Mejoras similares o mayores se obtuvieron con las otras voces. La más notoria se presenta en la voz JMK, donde 29 de los 39 coeficientes MFCC han sido mejor aproximados por el post-filtro discriminativo que con el simple, es decir, un 75.4 %. Este caso, en donde las mejoras se comparan con los post-filtros simples en lugar de la voz HTS, y se puede comprobar cómo se presentan mejoras en todos los casos, es una muestra de los beneficios del enfoque propuesto en el presente capítulo.

### 6.3.2 Mejora estadísticamente significativa del habla sintetizada

En esta sección se presenta un análisis estadístico enfocado a determinar en cuáles de los casos presentados la mejora es estadísticamente significativa con respecto a las medidas de la voz HTS. La razón de realizar este análisis es el hecho de que algunos de los sistemas propuestos pueden obtener un mejor resultado entre los post-filtros, pero aún así no mejorar los valores de la voz sintetizada.

Para el análisis estadístico se aplicó la Prueba HSD de Tukey, la cual realiza comparaciones entre pares de grupos de datos, tomando como referencia los del habla HTS. En las tablas 6.4 a 6.6 se reportan los resultados de esta prueba.

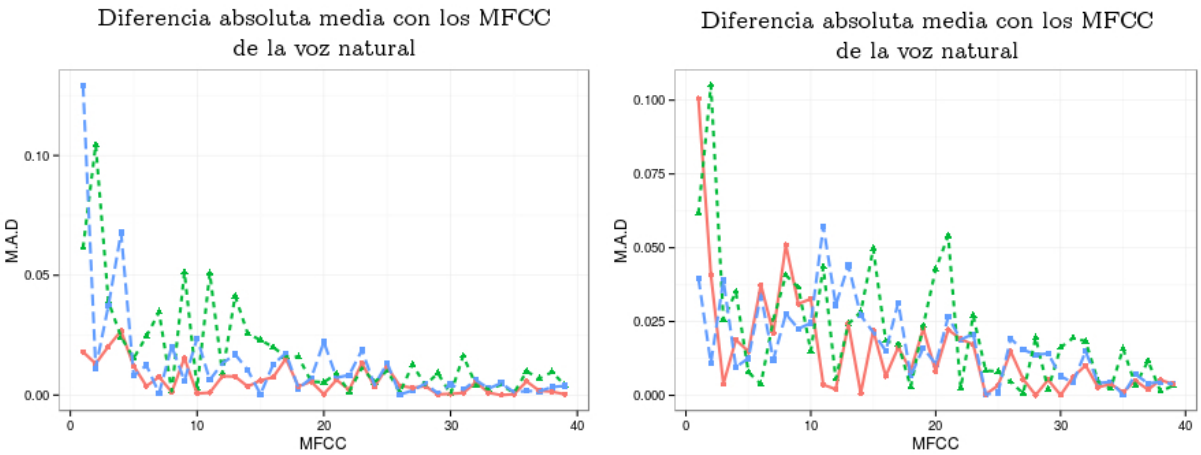
En la Tabla 6.4, la prueba estadística muestra la capacidad de los post-filtros discriminativos para mejorar la medida WSS en la voz HTS en más casos que los sistemas no-discriminativos. El caso más destacado con respecto a los post-filtros de los capítulos 5 y 6 se encuentra en el




**(a)** CLB



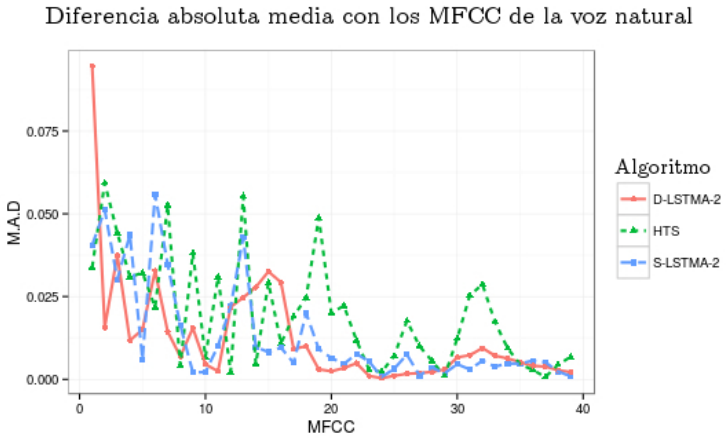

**(b)** BDL




**(c)** JMK




**(d)** RMS




**(e)** SLT

**Figura 6.4:** Comparación de la medida de diferencias absolutas para los coeficientes MFCC entre los distintos algoritmos y la voz natural.

post-filtro discriminativo LSTM-3, lo cual significa que este sistema tiene una capacidad mucho mayor para la mejora en el parámetro $f_0$.

**Tabla 6.4:** Resultados de la medida WSS. ✓ indica una mejora estadísticamente significativa con respecto a la voz HTS. "ns" indica un valor que no difiere significativamente del obtenido por la voz HTS, mientras que los espacios en blanco representan que no hubo beneficio con la aplicación de los post-filtros.

| | Simple | | | Discriminativo | | |
|---|---|---|---|---|---|---|
| Voz | LSTM-1 | LSTM-2 | LSTM-3 | LSTM-1 | LSTM-2 | LSTM-3 |
| SLT | ns | ns | | ns | ns | ✓ |
| RMS | ✓ | ✓ | | ✓ | ✓ | ✓ |
| JMK | ✓ | ✓ | | ✓ | ✓ | ✓ |
| CLB | ns | ns | ns | ✓ | ✓ | ns |
| BDL | ✓ | ✓ | ns | ✓ | ✓ | ns |

Los resultados de la medida PESQ se muestran en la Tabla 6.5. El post-filtro discriminativo LSTM-1 obtuvo una mejora no significativa, de forma semejante a los no discriminativos LSTM-1 y LSTM-2. Dos resultados se destacan de esta tabla: ninguno de los post-filtros analizados mejoraron la medida de PESQ en la voz CBL, y el post-filtro discriminativo LSTM-3 fue el único que obtuvo una mejora significativa en una de las voces (SLT), lo cual refuerza el hecho de que se obtuvo un beneficio considerable con la mejora lograda en el parámetro $f_0$.

**Tabla 6.5:** Resultados de la medida PESQ. ✓ indica una mejora estadísticamente significativa con respecto a la voz HTS. "ns" indica un valor que no difiere significativamente del obtenido por la voz HTS, mientras que los espacios en blanco representan que no hubo beneficio con la aplicación de los post-filtros.

| | Simple | | | Discriminativo | | |
|---|---|---|---|---|---|---|
| Voz | LSTM-1 | LSTM-2 | LSTM-3 | LSTM-1 | LSTM-2 | LSTM-3 |
| SLT | ns | ns | | ns | ns | ✓ |
| RMS | ns | ns | | ns | | |
| JMK | ns | ns | | ns | | |
| CLB | | | | | | |
| BDL | ns | ns | | ns | ns | |

La mejora estadísticamente significativa del parámetro $\text{SegSNR}_\text{f}$ presenta resultados similares que la medida WSS, como se muestra en la Tabla 6.6. Los post-filtros discriminativos mejoraron de forma significativa o sin diferencia considerable en más casos que los post-filtros no discriminativos.

**Tabla 6.6:** Resultados de la medida $\text{SegSNR}_\text{f}$. ✓ indica una mejora estadísticamente significativa con respecto a la voz HTS. "ns" indica un valor que no difiere significativamente del obtenido por la voz HTS, mientras que los espacios en blanco representan que no hubo beneficio con la aplicación de los post-filtros.

| | Simple | | | Discriminativo | | |
|---|---|---|---|---|---|---|
| Voz | LSTM-1 | LSTM-2 | LSTM-3 | LSTM-1 | LSTM-2 | LSTM-3 |
| SLT | ns | ✓ | | ✓ | ✓ | ✓ |
| RMS | ✓ | ✓ | ns | ns | ✓ | ns |
| JMK | ns | | | ns | ns | ns |
| CLB | ns | | ns | ✓ | ✓ | ns |
| BDL | ✓ | ✓ | ns | ns | ns | ns |

### 6.3.3 Evaluación subjetiva

Los resultados de la propuesta realizada en este capítulo fueron sometidos a una evaluación subjetiva, con la finalidad de medir el impacto en la percepción de la voz en escuchas humanos. Para esto, se seleccionaron veinte frases, seleccionadas al azar del conjunto de prueba en las cinco voces analizadas. En la evaluación subjetiva participaron sesenta personas, para las cuales se diseñó una prueba en línea donde se presentaron las frases de habla generada con HTS, procesada con el post-filtro simple y con el post-filtro discriminativo.

Todos los participantes son nativos de los Estados Unidos de América, tanto hombres como mujeres, con edades entre los veinte y los cincuenta años de edad. A cada participante se le pidió seleccionar la voz que representara la mejor calidad en términos de naturalidad.

Los resultados de preferencia se muestran en la Figura 6.5. Se observa cómo en todos los casos las voces procesadas con los post-filtros discriminativos fueron seleccionadas como más naturales que las producidas con el sistema HTS y las procesadas con los post-filtros simples. Las diferencias más notorias se encuentran en las voces RMS y BDL.

## 6.4 Resumen de contribuciones

En este capítulo se introdujo la idea de aplicar de forma discriminativa los post-filtros para la mejora del habla generada con el sistema HTS. De esta manera, se definió la división más general en los segmentos del habla, lo cual corresponde a regiones sonoras y no sonoras. Se

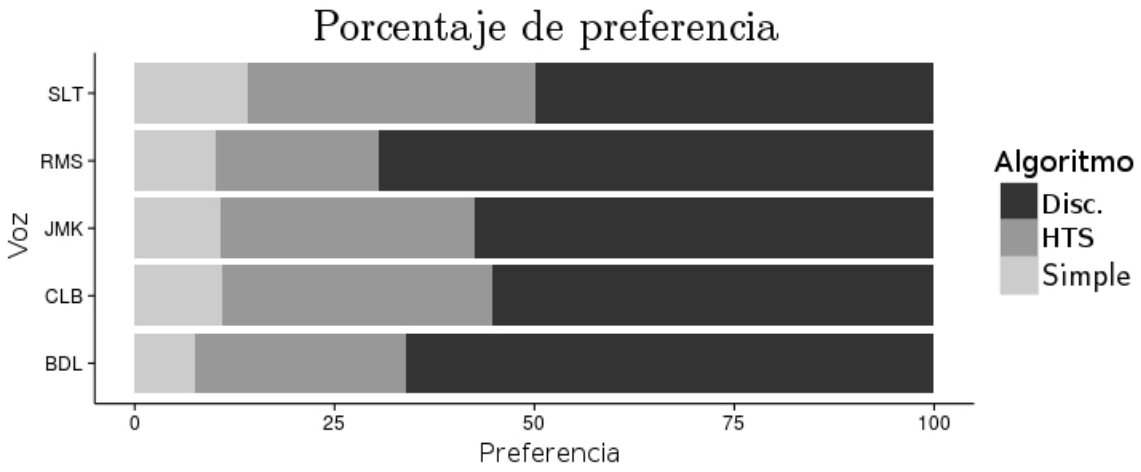



**Figura 6.5:** Porcentaje de preferencia para las voces generadas con HTS y los post-filtros discriminativos y no discriminativos.

realizó una extensa comparación utilizando cinco voces, tanto masculinas como femeninas, en bases de datos de idioma inglés. Para efectos de evaluar los resultados de forma objetiva se utilizaron tres conocidas medidas previamente consideradas en distintas publicaciones de mejora de señales, y en los capítulos precedentes de esta tesis, así como la media de la distancia absoluta entre los coeficientes MFCC de los distintos sistemas propuestos y el habla natural.

Para las medidas objetivas se realizaron pruebas estadísticas para determinar no solamente cuál es la mejor, sino cuándo la mejoría con respecto a la voz generada con HTS es significativa. Uno de los resultados más importantes de la aplicación en forma discriminativa de los post-filtros es la mejora lograda con los sistemas LSTM-3, los cuales realizan un mapeo solamente entre regiones que contienen $f_0$. Por esta razón tienen mejores posibilidades de acercar el contorno de $f_0$ de las frases sintetizadas hacia los valores del habla natural, en comparación con los sistemas no discriminativos.

La principal motivación para realizar el proceso de separación de acuerdo con las regiones sonoras y no sonoras de las frases para el entrenamiento y posteriormente el conjunto de prueba, radica en que la compleja relación que existe entre el habla de HTS y la natural se puede modelar mejor a partir de grupos semejantes de parámetros, en lugar de las frases completas. Los resultados de todas las medidas objetivas muestran los beneficios de aplicar los post-filtros de esta manera, en lugar de los enfoques que se han aplicado hasta el momento. La mejora también es notoria en la evaluación subjetiva realizada, con escuchas humanos que determinaron su preferencia de acuerdo con la naturalidad de los resultados en las cinco voces.

# Parte II

## Otras aplicaciones de los sistemas propuestos

# 7

# Introducción a la segunda parte

*En este capítulo se introducen las aplicaciones en las cuales los sistemas propuestos para mejorar la síntesis de voz han dado buenos resultados. Estas áreas son el mejoramiento de señales con ruido, y la aplicación de mapeos entre HMM para realizar cambio de acento en voces.*

## Índice

## 7.1 Introducción

En los capítulos precedentes se introdujo la idea de utilizar colecciones de post-filtros para mejorar los resultados de la síntesis de voz basada en Modelos Ocultos de Markov (*Hidden Markov Models*, HMM). La idea de mapear parámetros entre voces de distinta naturaleza o calidad ha sido previamente explorada en dos ámbitos:

- La síntesis de voz por adaptación: Se realiza a partir del mapeo de una voz bien entrenada (con suficientes datos) hacia una voz mal entrenada (con pocos datos), de manera que se mejora la calidad de la segunda.
- La mejora de señales de voz con ruido: El mapeo se realiza a partir de un conjunto de entrenamiento donde se encuentran frases de habla con ruido y frases de habla limpia, con la intención de que algoritmos (como las redes neuronales profundas) aprendan el mapeo entre ambas. De esta manera se les dota de la capacidad para eliminar niveles de ruido en nuevas frases.

En las siguientes secciones se describen estas dos aplicaciones.

## 7.2 Adaptación de HMM

En la síntesis de voz basada en la adaptación de HMM, se crea en primer lugar una voz promedio a partir de los parámetros de varios hablantes (incluyendo distintos géneros), con el objetivo de que esta voz contenga suficiente información y sea factible su transformación hacia otra voz deseada. En una segunda etapa, estos modelos se adaptan, mediante transformaciones lineales, para asemejarse a los de la voz objetivo, de la cual usualmente se tienen menos datos (como unos pocos minutos en comparación con varias horas de grabaciones para los modelos promedio). Cuando la voz promedio está en el mismo idioma que la voz objetivo, este procedimiento recibe el nombre de adaptación intra-lenguaje.

En la Figura 7.1 se ilustra este proceso, en el cual los parámetros de los HMM de la voz promedio (especialmente las distribuciones de probabilidad para las emisiones de símbolos) se ajustan para asemejarse a las distribuciones de la voz objetivo.

En el proceso de adaptación, los vectores de medias y matrices de covarianza de las distribuciones de probabilidad en los estados de los HMM, se transforman para coincidir (idealmente) con los

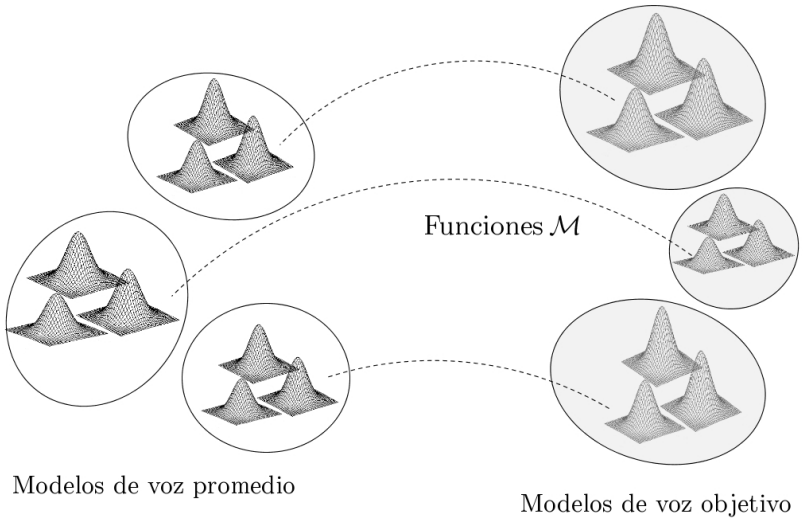



**Figura 7.1:** Ilustración del proceso de adaptación mediante un conjunto de transformaciones lineales $\mathcal{M}$

del hablante objetivo. Para realizarlo, una transformación lineal se puede utilizar, por ejemplo la Regresión lineal de máxima probabilidad (*Maximum Likelihood Linear Regression*, MLLR) para distribuciones gaussianas:

$$\overline{\mu}_i = \zeta_k \mu_i + \varepsilon_k, \tag{7.1}$$

o la Regresión lineal de máxima probabilidad restringida (*Constrained Maximum Likelihood Linear Regression*, CMLLR) que considera media y covarianza:

$$\overline{\mu}_i = \zeta_k \mu_i + \varepsilon_k \tag{7.2}$$

$$\overline{\Sigma}_i = \zeta_k \Sigma_i \varepsilon_k, \tag{7.3}$$

donde $\mu_i$ es el vector de medias, $\zeta_k$ es la matriz de transformación lineal, $\Sigma_i$ es la matriz de covarianza, y $\varepsilon$ el vector bias. Un árbol de regresión se genera en el proceso, basado en las similaridades de las distribuciones gaussianas, para compartir la misma transformación entre ellas.

Este técnica de adaptación surgió para mejorar el proceso de reconocimiento de habla [70][71], considerando que si se tiene un conjunto de HMM entrenados adecuadamente para reconocer el habla de un individuo, con transformaciones lineales podrían modificarse para reconocer

adecuadamente un nuevo hablante. Posteriormente, la idea de transformar las distribuciones se llevó a la síntesis de voz con la finalidad de modificar una voz a partir de otra de referencia.

Más recientemente, la idea de utilizar la adaptación en el reconocimiento de voz ha sido propuesta para mejorar la tasa de reconocimiento ante cambios de acento [72]. Este tipo de transformación, en síntesis, ha probado su capacidad de cambiar las características prosódicas del habla [73].

En nuestra propuesta, se explora la transformación de las distribuciones de los HMM entre una voz promedio que tiene un determinado acento de castellano, y una voz objetivo con otro acento. Para esto es necesario explorar las diferencias existentes entre los fonemas de ambos acentos, y determinar la forma de proceder para la transformación de los fonemas no coincidentes en ambos acentos.

## 7.3 Mejora de señales de voz en presencia de ruido

Durante las últimas décadas se han propuesto una gran cantidad de algoritmos para mejorar el desempeño de los sistemas modernos de comunicación en entornos ruidosos, para una importante variedad de aplicaciones, por ejemplo:

- Sistemas de reconocimiento de voz: Los sistemas iniciales de reconocimiento de voz han evolucionado desde el reconocimiento de palabras aisladas en condiciones controladas de laboratorio, a incorporarse en dispositivos móviles que puedan utilizarse en diversos entornos, aún en condiciones de ruido (tales como lluvia, viento, multitudes, ecos, distancia al micrófono). Estas condiciones adversas requieren una mejora de la señal ruidosa para que pueda ser procesada y reconocida adecuadamente en el dispositivo.
- Redes telefónicas móviles: Semejante al caso anterior, la emisión de voz en redes móviles puede realizarse en diversos entornos, además de distorsiones provocadas por la codificación y transmisión de la señal. La mejora de la calidad del habla en el receptor es deseable para completar la comunicación de forma adecuada.
- Dispositivos de asistencia: En aquellas personas con dispositivos de asistencia para escuchar, existe el interés de poder filtrar la señal de habla y limpiar los posibles ruidos que la puedan estar afectando, de manera que el dispositivo pueda amplificar la voz que se desea escuchar y no los ruidos que la acompañan.

Los algoritmos destinados a mejorar el habla en estas aplicaciones tienen la finalidad de corregir las características de la señal para incrementar la inteligibilidad y calidad general de las ondas. En años recientes se han desarrollado sistemas para este fin que incluyen algoritmos de aprendizaje

profundo, los cuales surgen al combinar varias capas o etapas de modelos más simples. En particular, sistemas que incluyes redes neuronales con memoria a corto y largo plazo (LSTM) han superado otros sistemas basados en procesamiento digital de señales y otros tipos de redes neuronales.

El método principal utilizado para implementar este tipo de algoritmos es en forma de *autoencoders*, donde la red es entrenada para realizar el proceso de eliminación de ruido o distorsiones, aproximando de esta manera una función de regresión. En aplicaciones como reducción de ruido para sistemas de reconocimiento de habla, se han implementado sobre una parametrización de la señal que incluye coeficientes MFCC, ya que los sistemas de reconocimiento tradicionalmente funcionan con estos parámetros. Dado que el funcionamiento de estos sistemas se degrada al incluirse ruido en la señal, al mejorar la calidad de los coeficientes con los *autoencoders* se puede mejorar la calidad del sistema en términos de tasa de reconocimiento.

En esta segunda parte de la tesis se presentan dos propuestas para mejorar señales de voz degradada con distintos tipos de ruido a diversos niveles. Los tipos de ruido contemplan tanto aquellos generados de forma artificial, así como un ruido natural, y niveles que permiten analizar degradación desde ligera hasta severa de la señal de voz.

Las propuestas tienen como propósito ampliar la manera en que las redes neuronales profundas han sido aplicadas para mejorar la señal. En una primera dirección, contemplando todos los parámetros de la señal de voz, no solamente los MFCC, de forma que la señal ruidosa es parametrizada, los parámetros son mejorados por separado con distintos algoritmos, y finalmente reunidos y el habla re-sintetizada.

El método principal combina dos tipos de arquitectura de redes neuronales: los *denoising autoencoders* (AE), y las memorias auto-asociativas (AAM), ambos implementados con unidades LSTM. Esta combinación constituye un enfoque de flujo múltiple (*multi-stream*), aplicado a la parametrización de la señal de habla. En los siguientes tres capítulos se desarrollan propuestas para ambas aplicaciones.

# 8

# Modificación de acento en voces

*En este capítulo se plantea la aplicación de la técnica de adaptación de HMM para producir un cambio de acento entre dos voces en variantes del castellano. Se describe la experimentación y evaluación de la propuesta, con medida objetivas y subjetivas.*

## Índice

## 8.1 Introducción

En la comunicación verbal, existe una gran cantidad de aspectos para-lingüísticos de diversas categorías, tales como edad, emoción y género, así como acentos de acuerdo con aspectos sociales o regionales del hablante [74]. Las variaciones en los acentos están definidos por múltiples aspectos en la pronunciación, cambios de ritmo y variaciones de tono [75] [76].

Una de las características más prominentes de la síntesis de voz basada en Modelos Ocultos de Markov (HMM) es su capacidad para cambiar las características del habla generada. Esto es posible al modificar los parámetros de los HMM, por ejemplo, con la técnica de adaptación presentada en [77]. En ésta, se pueden obtener voces de calidad a partir de una cantidad pequeña de datos, gracias a la transformación de modelos adecuadamente ajustados para una voz en particular, de manera que se asemejen a los de la voz de la cual se cuenta con pocos datos.

Esta manera de entrenar y transformar los HMM ha sido utilizada en diversas publicaciones, cuyo fin ha sido acercarse a la conversión de voces entre distintos idiomas [78][79][80][81][82][83]. En este caso, una voz que ha sido parametrizada y para la cual se cuenta con un conjunto de HMM, se utiliza como referencia para transformar los HMM de un idioma distinto, de manera que lo dicho en un idioma puede expresarse en uno nuevo conservando las características del hablante del primero.

Para algunas aplicaciones de síntesis de voz, incluyendo la conversión voz a voz, el acento del lenguaje objetivo puede ser importante, por ejemplo, cuando se realiza doblaje de películas para distintas regiones donde se habla castellano, para representar mejor la forma de hablar de cada región. Se han estudiado características como las diferencias a nivel fonético y de contorno de $f_0$ entre castellano europeo y latinoamericano [84]. Idealmente, un sistema que realice conversión voz a voz de un lenguaje a otro que preserve las características del hablante, debe tomar en cuenta el acento del escucha en el otro extremo de la comunicación.

En este capítulo se explora la técnica de adaptación de HMM para cambiar el acento de voces grabadas en castellano de Europa hacia el acento mexicano y viceversa, analizando en primer lugar las diferencias fonéticas entre ambos acentos e introduciendo transformaciones lineales para completar aquellos sonidos inexistentes en el acento original, o su modificación en el acento objetivo.

## 8.2 Transformación de distribuciones de HMM

Para realizar el proceso de adaptación entre diferentes lenguajes para producir síntesis de voz, la primera propuesta de mapeo entre los estados de los HMM se propuso en [77]. En ésta, la adaptación se realiza al introducir una voz objetivo que está en un lenguaje distinto al de la voz promedio [85][80]. Al establecer reglas para las transformaciones lineales $\mathcal{M}$ entre los lenguajes L1 y L2, estas transformaciones pueden relacionar los dos diferentes lenguajes.

En el caso de transformación entre castellano europeo y castellano mexicano, nuestra propuesta parte de dos bases de datos grabadas para ambos acentos, pronunciadas por distintos hablantes. Se realiza de esta manera para que la cantidad de datos entre ambas sea igual, además de que la mayoría de los fonemas entre ambos sean coincidentes. Esto significa que la mayoría de los HMM $\lambda$ tendrán su equivalente en el otro acento, pero las diferencias deben ser tomadas en cuenta para el caso en que los fonemas difieran.

El castellano europeo tiene una descripción de 29 símbolos fonéticos en SAMPA [86]. Tal como se ha descrito en [87], para los acentos latinoamericanos, incluyendo el mexicano, los grafemas [j] y [g] seguidos de [e] o [i] se transcriben fonéticamente como /h/, mientras que para el castellano europeo, como [x]. Las otras dos diferencias significativas se encuentran en los grafemas [s] y [z], los cuales son trascritos fonéticamente para el castellano europeo como /s/ y /T/, respectivamente. En castellano mexicano, todos los casos de [s], [z] y [c] seguida de [e] o [i] se transcribe como /s/.

Para realizar el proceso de modificación de acento con la síntesis HMM basada en adaptación, sean $\mathbb{S}_{in}$ y $\mathbb{S}_{out}$ las distribuciones de los estados de los HMM en el acento promedio y el objetivo, respectivamente. Como se mencionó previamente, la mayoría de las distribuciones tendrán un equivalente en el otro acento. La propuesta es realizar el mapeo entre un acento A1 y el otro A2 a partir de las siguientes reglas:

- De castellano europeo a mexicano: El proceso de adaptación establece reglas de mapeo entre $\mathbb{S}_{in}$ hacia $\mathbb{S}_{out}$:

$$\mathcal{M}_d : \mathcal{M}_d(\mathbb{S}_{in}) = \mathbb{S}_{out}. \tag{8.1}$$

  En este caso, la frase a sintetizar requiere la transcripción fonética del castellano mexicano. Como éste tiene una cantidad mayor de fonemas, se hace necesario sustituir las transcripciones fonéticas de las frases que contienen los sonidos /T/ y /s/ por /s/ en todos los datos. De esta manera, el mapeo ignora las distribuciones gaussianas que describen los sonidos inexistentes en el acento mexicano. Las nuevas distribuciones gaussianas para los

fonemas se crean con el mapeo desde el castellano europeo promedio. En la Figura 8.1 se ilustra el proceso de mapeo entre el castellano europeo al mexicano.

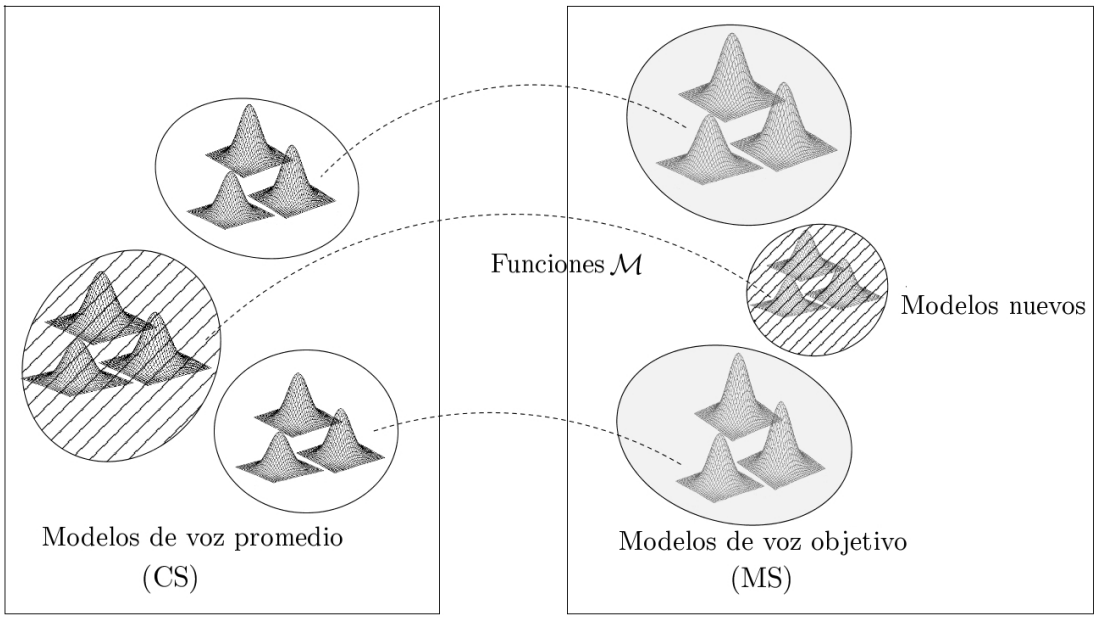



**Figura 8.1:** Mapeo de distribuciones del acento castellano europeo (CS) al acento mexicano (MS)

- De acento mexicano a castellano europeo: El proceso de adaptación establece reglas de mapeo entre distribuciones en $\mathbb{S}_{out}$. Todos las distribuciones en $\mathbb{S}_{in}$ serán transformadas en $\mathbb{S}_{out}$. En este caso, la frase a sintetizar requiere la transcripción fonética del castellano europeo. Las nuevas distribuciones de los fonemas se crean mediante el mapeo desde la voz promedio. En la Figura 8.2 se ilustra este proceso.

El número de transformaciones requeridas para el proceso de adaptación se estiman directamente a partir de la cantidad de datos disponible. Si solamente se cuenta con una base de datos de pocas frases, una transformación global se aplica a todas las distribuciones. El procedimiento general se describe en el Algoritmo 9.

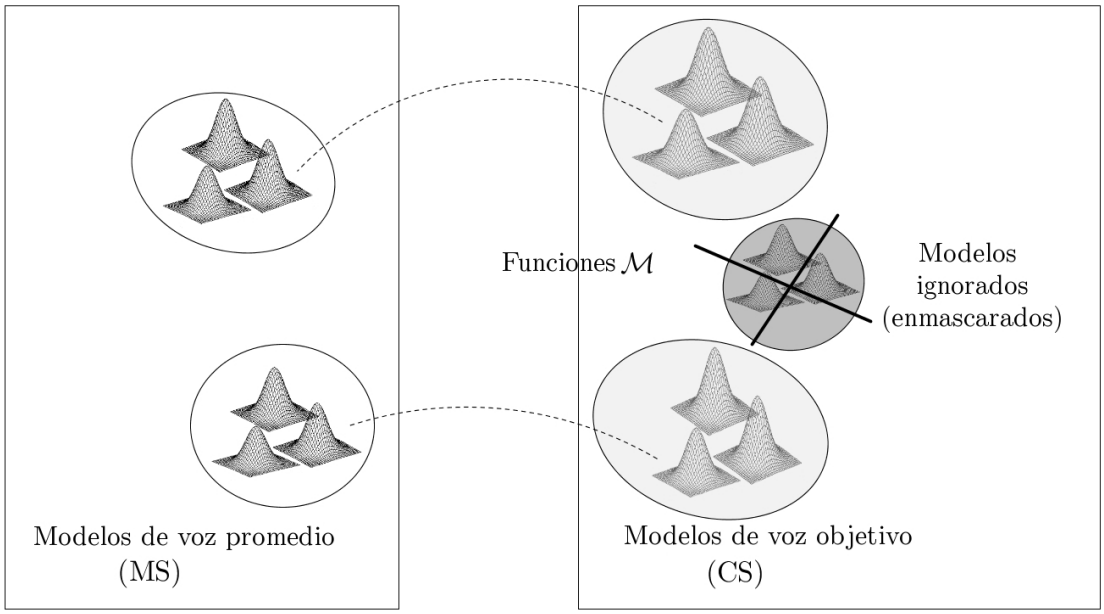



**Figura 8.2:** Mapeo de distribuciones del acento mexicano al castellano europeo

**Algoritmo 9** Entrenamiento para realizar cambio de acento

**Entrada:** $A_1$: Frases de voz en acento 1
**Entrada:** $A_2$ : Frases de voz en acento objetivo
**Salida:** Cambio de acento de la voz $A_1$

**mientras** Exista frase **hacer**
  extraer características
  // Crear HMMs de la voz promedio con el acento objetivo, con el sistema HTS //
  estimar $\lambda^1_{A_2}, \lambda^2_{A_2}, \ldots, \lambda^n_{A_2}$
  // Crear HMMs de la voz que se desea cambiar de acento, con el sistema HTS //
  estimar $\lambda^1_{A_1}, \lambda^2_{A_1}, \ldots, \lambda^n_{A_1}$
  // agrupar estados HMM para voz $A_1$ y $A_2$ usando árboles //
  definir $P = \{A_i : i \in I\}$
  // estimar funciones MLLR entre grupos de estados //
  $f_1(P_1(A_1)) \approx P_1A_2, \ldots, f_I(P_1(A_1)) \approx P_IA_2$
**fin mientras**
guardar mejor estimación de funciones MLLR
**devolver** conjunto de funciones MLLR

## 8.3 Procedimiento experimental

### 8.3.1 Descripción de los datos

Dos actores profesionales de doblaje de nacionalidad mexicana, género masculino y femenino, realizaron grabaciones de 184 frases en este acento con siete emociones: neutral, enojo, alegría, miedo, sorpresa, tristeza y disgusto. Estas frases incluyen palabras aisladas así como frases afirmativas y exclamativas, se muestra en la Tabla 8.1.

**Tabla 8.1:** Contenido de las frases de la base de datos, por género y hablante

| Número de frase | Contenido |
|---|---|
| 1-100 | Frases afirmativas |
| 101-134 | Frases interrogativas |
| 135-150 | Párrafos |
| 151-160 | Dígitos |
| 161-184 | Palabras aisladas |

La selección de palabras, frases y párrafos son las mismas diseñadas en [88]. Las grabaciones se realizaron en un estudio profesional donde todos los aspectos técnicos y condiciones de grabación estuvieron completamente controlados. En condiciones similares fueron realizadas las grabaciones para las voces masculinas y femeninas en acento castellano europeo del catálogo ELRA.

Para crear las voces se utilizaron HMM de cinco estados, con distribuciones de probabilidad gaussianas estándar en el sistema HTS. Se utilizaron como parámetros la frecuencia fundamental ($f_0$), 39 coeficientes MFCC, así como sus aproximaciones a primer y segunda derivada, $\Delta$ y $\Delta\Delta$. Para realizar la transformación entre distribuciones se utilizó el algoritmo CMLLR. Para la voz que se desea el cambio de se utilizaron ciento treinta y cinco frases, mientras que para la voz de promedio con el otro acento se utilizaron más de setecientas frases.

Los árboles de decisión y regresión compartieron las mismas preguntas definidas en el proceso, para lograr un equilibrio entre la calidad obtenida en ambos casos.

### 8.3.2 Evaluación subjetiva

Para evaluar los resultados obtenidos de cambio de acento con el proceso de adaptación, se procedió a la elaboración de un sitio web, donde cada frase resultante fue evaluada acorde con tres opciones: acento castellano europeo, acento mexicano o acento no identificable. La evaluación consistió en cuatro frases por cada caso, y se contó para ello con más de treinta personas. Todas son adultos, con edades entre los 25 y los 55 años. Los casos considerados fueron:

- Acento castellano europeo sin adaptación (CS)
- Acento mexicano sin adaptación (MS)
- Modificación de acento de mexicano a castellano europeo (MS2CS)
- Modificación de acento de castellano europeo a mexicano (CS2MS)

Si bien el proceso de evaluación subjetiva es necesario para determinar el grado de éxito en la conversión de acento, el realizado para esta capítulo tiene algunas limitaciones. La principal es que las variables ambientales no fueron controladas, y la selección de la muestra no tiene una correlación con la población general.

### 8.3.3 Evaluación objetiva

Se ha estudiado que los acentos se pueden explicar parcialmente por las diferencias entre las duraciones de las vocales y la tasa de habla [89]. Para evaluar la modificación de acento realizada, el propósito es determinar la calidad de la conversión, explorando la duración de cada vocal, la tasa de habla y cómo los contornos de $f_0$ han sido afectados.

## 8.4 Resultados y discusión

Los resultados de la evaluación subjetiva de identificación de acento se muestran en la Figura 8.3. Se puede observar cómo la gran mayoría de opiniones identificaron de forma correcta el acento de la voz en castellano europeo, con un pequeño porcentaje de identificación incorrecta y ninguna opinión de acento no identificable. La voz con acento mexicano convertida a acento

castellano europeo también obtuvo una mayoría de opiniones que la identifican correctamente, aunque han aparecido algunas opiniones donde el acento no ha sido identificado.

En cuanto al acento mexicano, los resultados son menos contundentes, a pesar de que la voz convertida de acento castellano europeo a acento mexicano tiene una mayoría de opiniones que la identifican con el acento buscado. La baja tasa de identificaciones correctas de acento en la voz mexicana puede deberse a que el resultado de voz artificial tiene menor calidad con respecto a la voz castellana.

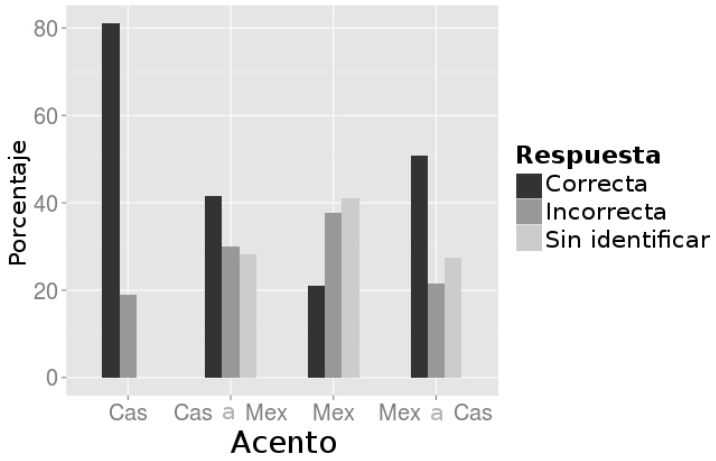



**Figura 8.3:** Porcentaje de idenficaciones correctas, incorrectas o sin identificar en la evaluación subjetiva. Cas: Voz castellana, Mex: Voz mexicana.

Los resultados de las evaluaciones objetivas basadas en duración de vocales se pueden observar en la Figura 8.4. En ésta, las diferencias más importantes se encuentran en la vocal <i>, cuya duración en español mexicano tiende a ser menor. La voz mexicana convertida a acento español incrementa su duración media, aunque los valores han quedado aún por encima de las vocales españolas. Un efecto semejante se encuentra en la vocal <u>, con tendencia a mayor rango de duraciones en el acento español mexicano, por lo que la conversión tiende a incrementar el rango de duración de esta vocal al convertir la voz castellana a mexicana, así como a reducir el rango de valores en la conversión en el otro sentido.

En cuanto a la tasa de habla, la voz de español castellano tiene una tasa menor de fonemas por segundo que la mexicana, como se muestra en la Tabla 8.2. La conversión de acento afecta positivamente la tasa de habla en el sentido de acercarlo hacia el acento que se desea convertir (objetivo).

Finalmente, en la Figura 8.5 se muestra cómo un contorno de $f_0$ se modifica de la voz castellana para hacerla más parecido a la voz mexicana mediante la conversión de acento por adaptación.

Todos los contornos de $f_0$ tienen la misma escala vertical y se trata de la misma frase pronunciada en las tres voces.

Estas medidas objetivas coinciden con los resultados objetivos en evidenciar un cambio en las características de las voces con el proceso de adaptación, el cual las hace asemejarse más a las características de las voces en el acento objetivo. De esta manera se verifica la capacidad del proceso para realizar cambio de acento entre variantes del castellano.

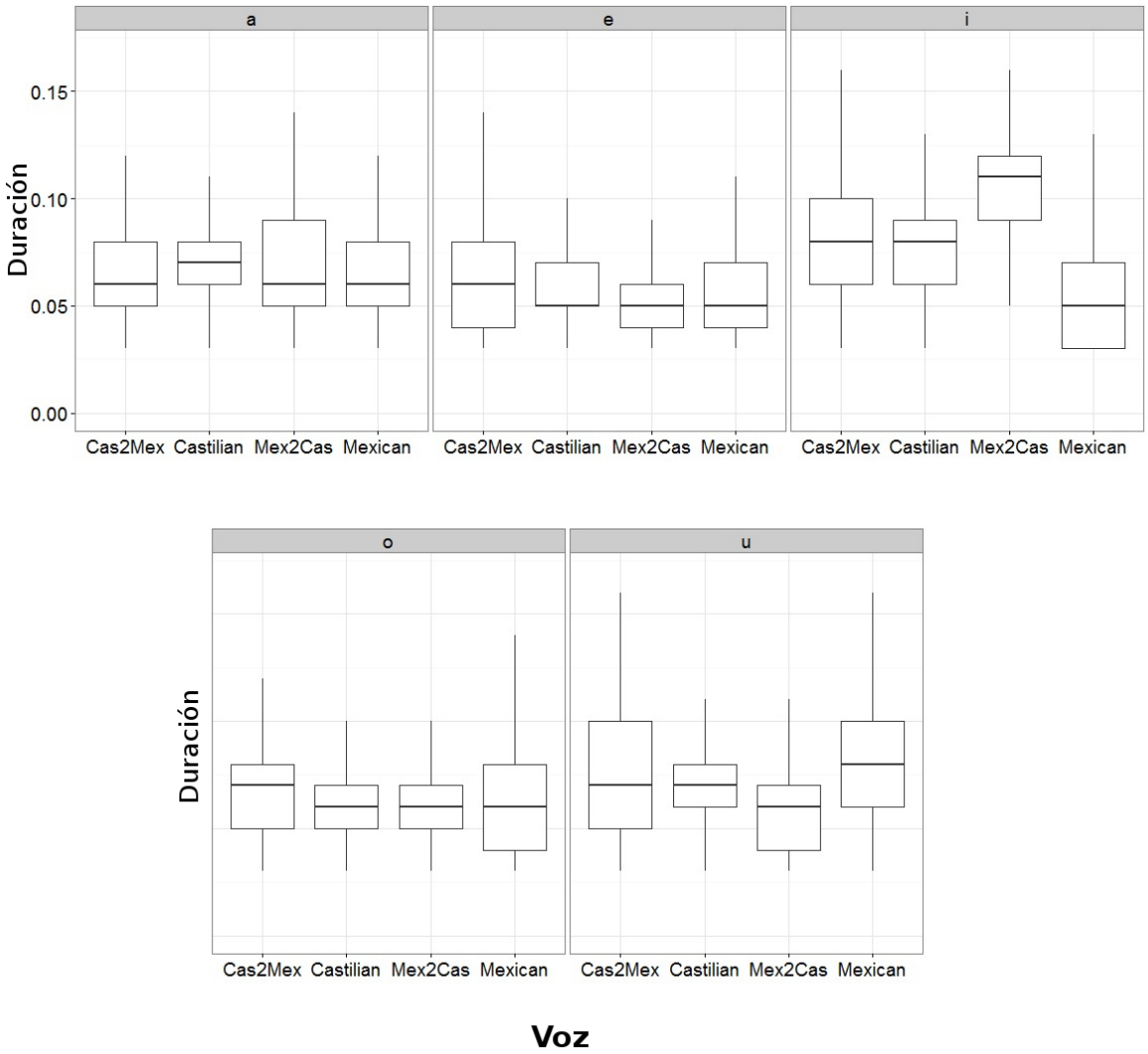
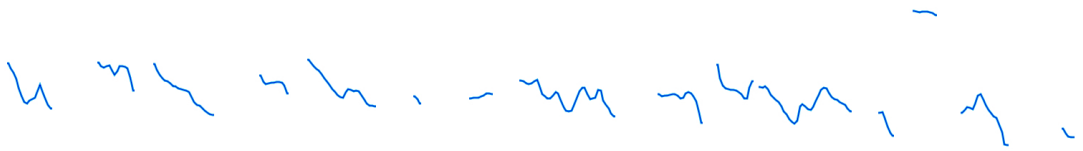



**Figura 8.4:** Diagramas de caja de las duraciones de las vocales para los acentos y conversiones realizadas

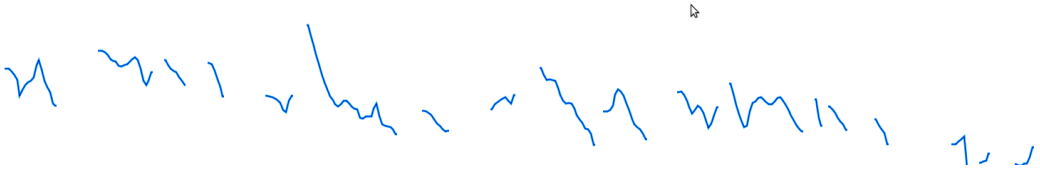
**Tabla 8.2:** Tasa de habla de 100 frases en cada acento y conversión. MS: Acento mexicano. CS: Acento castellano europeo

| Voz | Razón de habla |
|---|---|
| MS | 13.44 |
| CS | 11.07 |
| MS2CS | 12.99 |
| CS2MS | 11.30 |

**(a)** Frase con acento castellano europeo



**(b)** Frase convertida de acento castellano europeo a acento mexicano



**(c)** Frase con acento mexicano

**Figura 8.5:** Contornos de $f_0$ de la frase "Con estoico respeto a la justicia adyacente guardó sus flechas". El eje horizontal tiene escala de tiempo, y el vertical el valor correspondiente de $f_0$.

## 8.5 Resumen de contribuciones

En esta sección se ha investigado la manera en que la técnica de adaptación para síntesis HMM se pude utilizar para modificar el acento del habla de castellano europeo hacia castellano mexicano y viceversa. Esta investigación es pionera en el campo de modificación de acento para variantes del castellano.

Los resultados subjetivos han mostrado una identificación adecuada del cambio de acento, especialmente en la conversión del acento mexicano al castellano europeo. En la otra dirección los resultados no han sido tan claros, lo cual se puede deber a la calidad de las voces obtenidas de las bases de datos con acento mexicano.

Las evaluaciones objetivas han mostrado cómo los elementos prosódicos de las voces se ven afectados, haciendo parecerse estas características hacia el acento objetivo. Esto es coincidente con los resultados de las evaluaciones subjetivas, lo cual muestra que la técnica de adaptación ayuda a cambiar la prosodia acorde con la modificación de acento.

# 9

# Mejora de señales de voz en presencia de ruido

*En este capítulo se plantea la aplicación de múltiples arquitecturas de redes LSTM para mejorar señales de voz que han sido degradadas con ruido de diferente naturaleza e intensidad.*

## Índice

## 9.1 Introducción

En el tema de mejora de señales de voz, una señal degradada con ruido aditivo se procesa para mejorar su calidad con respecto a factores como la inteligibilidad y la calidad percibida. Se puede asumir que una señal con ruido $y$, es la suma de una señal de habla pura, $x$ y un ruido $d$, expresada como

$$y(t) = x(t) + d(t) \tag{9.1}$$

Aplicando la Transformada Discreta de Fourier, la formulación en el dominio espectral se expresa como:

$$Y_k(n) = X_k(n) + D_k(n) \tag{9.2}$$

En los métodos clásicos, principalmente relacionados con técnicas de procesamiento de señales, $x(t)$ se considera no correlacionado con $d(t)$, de manera que una amplia cantidad de estos algoritmos estiman $X_k(n)$ a partir del espectro de $y(t)$ y $d(t)$. En aplicaciones prácticas, no se asume ningún conocimiento previo de las características de estas señales, de manera que los algoritmos dependen en una estimación adecuada de ambos.

En aquellos basados en aprendizaje profundo, $x(t)$ (o $X_k(n)$) se puede estimar a través de una función $f(\cdot)$ entre los datos ruidosos y limpios, de la forma

$$\tilde{x}(k) = f\left(y(k)\right). \tag{9.3}$$

La precisión de la aproximación $f(\cdot)$ usualmente depende de la cantidad de datos disponibles para el entrenamiento del algoritmo seleccionado. En el presente capítulo se aplica una colección de redes LSTM para esta función $f$, semejante a las utilizadas para mejorar el habla artificial en los capítulos 4 a 6.

## 9.2 Trabajo relacionado

Recientemente se han presentado una diversidad de métodos basados en algoritmos de aprendizaje profundo para mejorar señales de voz. La gran mayoría de éstos abordan el problema de

mejorar características derivadas del espectro, típicamente MFCC. Esto se debe principalmente a la influencia del área de investigación que se ocupa del reconocimiento de habla, el cual se ha basado en las últimas décadas principalmente en la extracción y clasificación de MFCC, por su relación con la escala de percepción humana del sonido.

Uno ejemplo de esto se encuentra en [90][91], donde el procedimiento de mejora del habla se basa en la extracción de coeficientes MFCC, así como las aproximaciones discretas de primera y segunda derivadas ($\Delta$ y $\Delta\Delta$). Con esto, los algoritmos realizan la tarea de regresión entre pares ruidosos y limpios de la señal, y se integra en el reconocimiento características dinámicas del habla.

En [92] se experimentó con una combinación de parámetros: el uso de MFCC del audio y características provenientes de vídeo para realizar reconocimiento de voz, utilizando como algoritmo las redes de creencia profunda (DBN). Este tipo de algoritmos han superado otros considerados como clásicos para eliminar ruido de señales, a distintos SNR y con distintos tipos de ruido [93][94][48][49].

Como se ha mencionado, el mecanismo principal de acción de los algoritmos de aprendizaje profundo es su utilización para eliminar ruido a partir de una función de regresión, la cual es estimada a partir de los datos, tomando ejemplos de señales con y sin ruido. Si bien su efectividad ha sido probada, entrenar una red neuronal se realiza tradicionalmente para un tipo específico de ruido, sobre el cual se cuenta con ejemplos suficientes. Si la red entrenada se utiliza con otro tipo de ruido u otro nivel, su eficacia disminuye.

Por esta razón, aunque el conocimiento del tipo de ruido o su intensidad no son necesarios para realizar el entrenamiento, sí es importante poder determinar el nivel de ruido en aplicaciones prácticas, para utilizar la red entrenada con las condiciones más próximas a la señal de entrada. Esto ha llevado a implementar sistemas para estimar el ruido en tiempo real, y así ajustar el modelo o seleccionarlo adecuadamente [95].

Otras distorsiones en la señal de habla, como las reverberaciones producidas al rebotar la onda en el espacio e incidir en el micrófono han sido analizadas utilizando algoritmos semejantes, planteando un problema de regresión entre los MFCC de una señal distorsionada y los de una limpia [96][97]. Redes recurrentes y mezclas con otros algoritmos para el reconocimiento (como los HMM) han sido también entrenados y probados con éxito para este propósito [98][99][100].

Por otra parte, además de la parametrización del habla con MFCC, otras representaciones, tal como la Predicción Perceptual Lineal (PLP) y sus derivadas hasta el tercer orden han sido probadas con algoritmos de aprendizaje profundo [101], utilizando procedimientos semejantes para el entrenamiento a los utilizados con MFCC.

También en sistemas híbridos los algoritmos de aprendizaje profundo han probado su utilidad. Por ejemplo, para potenciar otros algoritmos, como el filtro Wiener, el cual requiere una

estimación de las características del ruido para su utilización. En este caso las redes neuronales se emplean como una primera etapa para mejorar esta estimación [102].

Finalmente, es importante destacar la utilización más reciente de redes neuronales en las cuales las neuronas pueden sustituirse con unidades más complejas, como aquellas capaces de almacenar valores en el corto o en el largo plazo (LSTM). Estas se han presentado con gran éxito para sistemas de eliminación de ruido y reverberación en coeficientes MFCC [103][98] y en arquitecturas del tipo *autoencoders*, eliminando también señales con la complejidad de música de fondo [104], y probadas en cascada, donde una red recibe y realiza la tarea de regresión con la salida de la anterior [51].

En estas propuestas se observa que se ha privilegiado la mejora de señales en sus coeficientes de espectro, y la utilización de un tipo de red neuronal para realizar la operación de regresión. Si bien los sistemas híbridos han sido probados, se han implementado para estimar los parámetros de un algoritmo, no en varias etapas que combinen algoritmos de distinta naturaleza.

Por esta razón, en el presente capítulo se probará un caso previamente no considerado en las referencias, el cual consiste en la combinación de varios algoritmos de aprendizaje profundo para tratar de forma separada distintos grupos de coeficientes de una parametrización de la señal de habla.

## 9.3 Sistema propuesto

La propuesta contempla la extensión de las técnicas de mejora de señales de habla que utilizan algoritmos de aprendizaje profundo, las cuales utilizan tradicionalmente características espectrales, para abarcar otras características de una parametrización completa de la señal, como $f_0$ y energía. Para realizarlo, se sigue un procedimiento en dos etapas, semejante al planteado para los post-filtros de síntesis de voz: En la primer etapa se utiliza un *autoencoder* para mejorar los coeficientes MFCC al aprender una función de regresión directamente de los datos. En la segunda etapa, se aplica una variante de la arquitectura de las memorias auto-asociativas para mejorar los coeficientes de energía y $f_0$.

Al terminar el proceso de mejora de los coeficientes por separado, se reconstruye la señal de habla, la cual se espera tenga características más cercanas a las de la voz sin ruido. La intención es determinar si este sistema de mejora, a partir de una colección de redes neuronales entrenadas por separado en lugar de una sola, puede mejorar los resultados de la aplicación los sistemas basados en algoritmos de aprendizaje profundo tal como se han realizado hasta el momento. De igual manera que en el caso de habla sintetizada, el modelo elegido para las redes profundas es el LSTM.

**Algoritmo 10** Entrenamiento de red LSTM para eliminar ruido en MFCC

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: frases de habla con ruido
**Entrada:** $L$: red inicializada
**Entrada:** $N$: número de iteraciones
**Entrada:** $K$: número de ventanas
**Salida:** $L_{ae}$: Red entrenada para eliminar ruido en 39 MFCC

**mientras** iteración < $N$ **hacer**
  **mientras** ventana < $K$ **hacer**
    extraer características: $M_1$: 39 MFCC de $n$ y $M_2$: 39 MFCC de $r$
    // propagar 39 MFCC de $r$ de la entrada hacia la salida de $L$ (ecuaciones 2.8 a 2.16) //
    calcular $f_L(M_2)$
    calcular error: $E(f_L(M_2), M_1)$
    **si** error < error mínimo **entonces**
      // ajustar pesos de $L$ (ecuaciones 2.17 a 2.22) //
      $L_{ae} \leftarrow L$
    **fin si**
  **fin mientras**
**fin mientras**
**devolver** $L_{ae}$

**Algoritmo 11** Entrenamiento de red LSTM para eliminar ruido en $f_0$

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: frases de habla con ruido
**Entrada:** $L$: red inicializada
**Entrada:** $N$: número de iteraciones
**Entrada:** $K$: número de ventanas
**Salida:** $L_{maa_1}$: Red entrenada para eliminar ruido en parámetro $f_0$

**mientras** iteración < $N$ **hacer**
  **mientras** ventana < $K$ **hacer**
    extraer características: $M_1$: 39 MFCC y $f_0$ de $n$ y $M_2$: 39 MFCC de $n$ y $f_0$ de $r$
    // propagar 39 MFCC y $f_0$ de $r$ de la entrada hacia la salida de $L$ (ecuaciones 2.8 a 2.16) //
    calcular $f_L(M_2)$
    calcular error: $E(f_L(M_2), M_1)$
    **si** error < error mínimo **entonces**
      *a*justar pesos de $L$ (ecuaciones 2.17 a 2.22)
      $L_{maa_1} \leftarrow L$
    **fin si**
  **fin mientras**
**fin mientras**
**devolver** $L_{maa_1}$

**Algoritmo 12** Entrenamiento de red LSTM para eliminar ruido en coeficiente de energía

**Entrada:** $n$: Frases de habla natural
**Entrada:** $r$: frases de habla con ruido
**Entrada:** $L$: red inicializada
**Entrada:** $N$: número de iteraciones
**Entrada:** $K$: número de ventanas
**Salida:** $L_{maa_2}$: Red entrenada para eliminar ruido en parámetro de energía

```
mientras iteración < N hacer
    mientras ventana < K hacer
        extraer características: M1: 39 MFCC y energía de n y M2: 39 MFCC de n y energía de r
        // propagar 39 MFCC y energía de r de la entrada hacia la salida de L (ecuaciones 2.8 a 2.16) //
        calcular f_L(M_2)
        calcular error: E(f_L(M_2), M_1)
        si error < error mínimo entonces
            ajustar pesos de L (ecuaciones 2.17 a 2.22)
            L_maa2 ← L
        fin si
    fin mientras
fin mientras
devolver L_maa2
```

Como primera etapa del proceso se deben extraer los parámetros del habla con ruido y del habla limpia. A diferencia del habla sintetizada, la correspondencia entre los parámetros es directa, por lo que de forma natural se constituyen las entradas (parámetros con ruido) y las salidas (parámetros limpios) en las redes LSTM. Este proceso se ilustra en la Figura 9.1 para el caso de mejorar los coeficientes MFCC, y en la Figura 9.2 para el coeficiente de energía y $f_0$.

Por su parte, los cuatro tipos de sistemas que se aplicarán a la señal de habla ruidosa se describen a continuación:

- DLSMT-S: Se aplica un único *autoencoder* a todos los parámetros al mismo tiempo: energía, $f_0$ y los 39 coeficientes MFCC
- DLSTM-1: Un *autoencoder* se aplica a los 39 MFCC, dejando la energía y el $f_0$ sin procesar. Con la red entrenada, se reconstruye la señal con los MFCC mejorados y la energía y $f_0$ de la señal ruidosa. El proceso de entrenamiento se describe en el Algoritmo 10.
- DLSTM-2: Adicional al *autoencoder* considerado en el caso anterior, una memoria auto-asociativa adicional se entrena para mejorar el parámetro de energía. En la etapa de

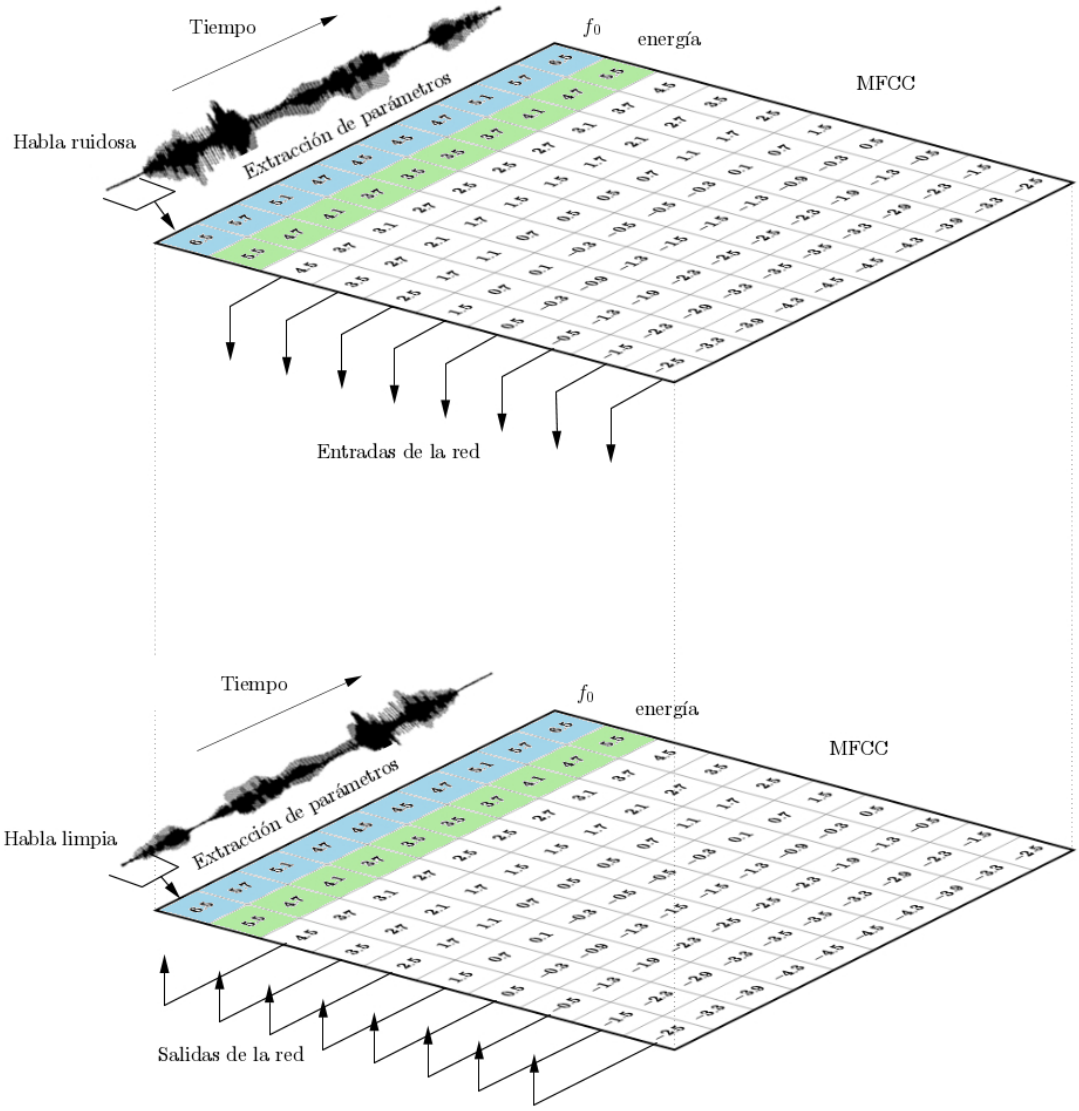



**Figura 9.1:** Ilustración del proceso de extracción de parámetros para el entrenamiento del *autoencoder* con MFCC del habla ruidosa y limpia.

reconstrucción de la señal, estos dos coeficientes mejorados se unen con el $f_0$ proveniente de la señal ruidosa. El proceso de entrenamiento se describe en el Algoritmo 11.

- DLSTM-3: Además de las dos redes contempladas en DLSTM-2, una memoria auto-asociativa adicional se utiliza para mejorar el parámetros de $f_0$. El proceso de entrenamiento se describe en el Algoritmo 12.

Cada LSTM utilizado se entrena para mapear los parámetros ruidosos ($y$) a los correspondientes del habla limpia ($x$). En el caso de las memorias auto-asociativas, se trata de una variante del

**Algoritmo 13** Eliminación de ruido en frases de prueba con redes LSTM

**Entrada:** $r$: Frases de habla con ruido
**Entrada:** $L_{ae}$
**Entrada:** $L_{maa_1}$
**Entrada:** $L_{maa_2}$
**Salida:** Onda procesada con reducción de ruido

**mientras** exista frase **hacer**
  extraer características: $f_0$, $e$: energía, $g$: 39 MFCC de $r$
  // predecir 39 MFCC mejoradas con entradas 39 MFCC de $r$ en $L_{ae}$ //
  $P_g = f_{L_{ae}}(g)$
  // predecir $f_0$ mejorada con entrada ($f_0$ de $r$, 39 MFCC mejoradas) en $L_{maa_1}$ //
  $P_{f_0} = f_{L_{maa_1}}(f_0, P_g)$
  // predecir energía mejorada con entrada (energía de $r$, 39 MFCC mejoradas) en $L_{maa_2}$ //
  $P_e = f_{L_{maa_2}}(e, P_g)$
**fin mientras**
Reconstruir habla con ($f_0$ mejorada, energía mejorada, 39 MFCC mejorados)
**devolver** Onda de habla reconstruida

modelo usual que realiza una aproximación de la función identidad, pues se entrena con los mismos 39 datos en las entradas y las salidas, pero con una entrada diferente, correspondiente a la energía (o a $f_0$) ruidosa, mientras que a la salida se ubica el mismo parámetro de la señal limpia. De esta manera, la memoria auto-asociativa realiza una reconstrucción del parámetro a partir de los MFCC. Los cuatro sistemas contemplados en esta sección se muestran en las figuras 9.3 a 9.6. La reconstrucción de la señal se describe en el Algoritmo 13.

## 9.4 Algoritmos de comparación

Se describirán brevemente algunas técnicas conocidas, basadas en procesamiento digital de señales, para comparar los resultados del sistema propuesto:

### 9.4.1 *Spectral Subtraction*

Este método asume que el espectro de potencia de la señal degradada con ruido puede modelarse como la suma del espectro de la señal limpia más el del ruido. El algoritmo consiste en la implementación de la relación definida como:

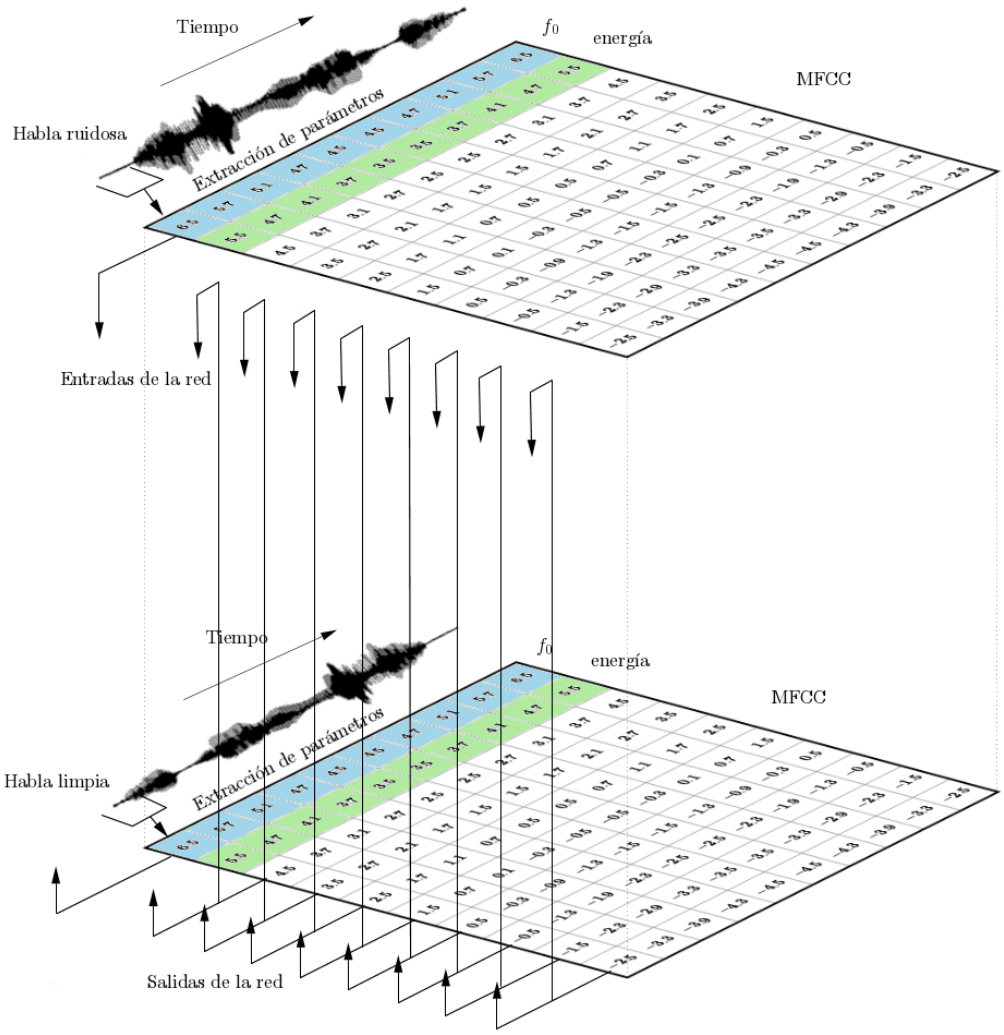



**Figura 9.2:** Ilustración del proceso de extracción de parámetros para el entrenamiento de la memoria auto-asociativa para la mejora del parámetro $f_0$.

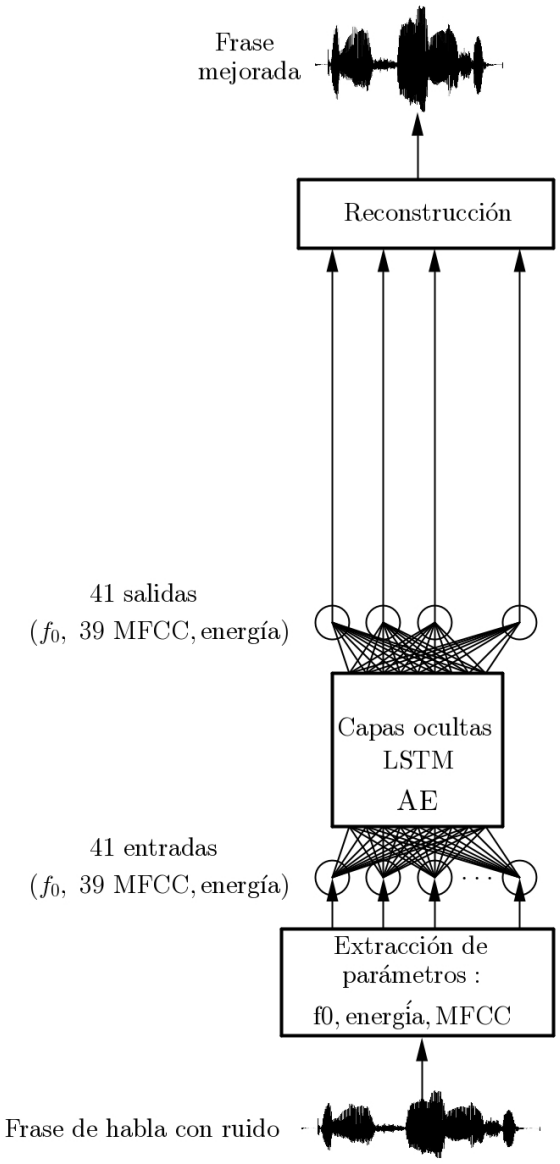



**Figura 9.3:** Sistema DLSTM-S para mejorar todos los parámetros del habla ruidosa al mismo tiempo

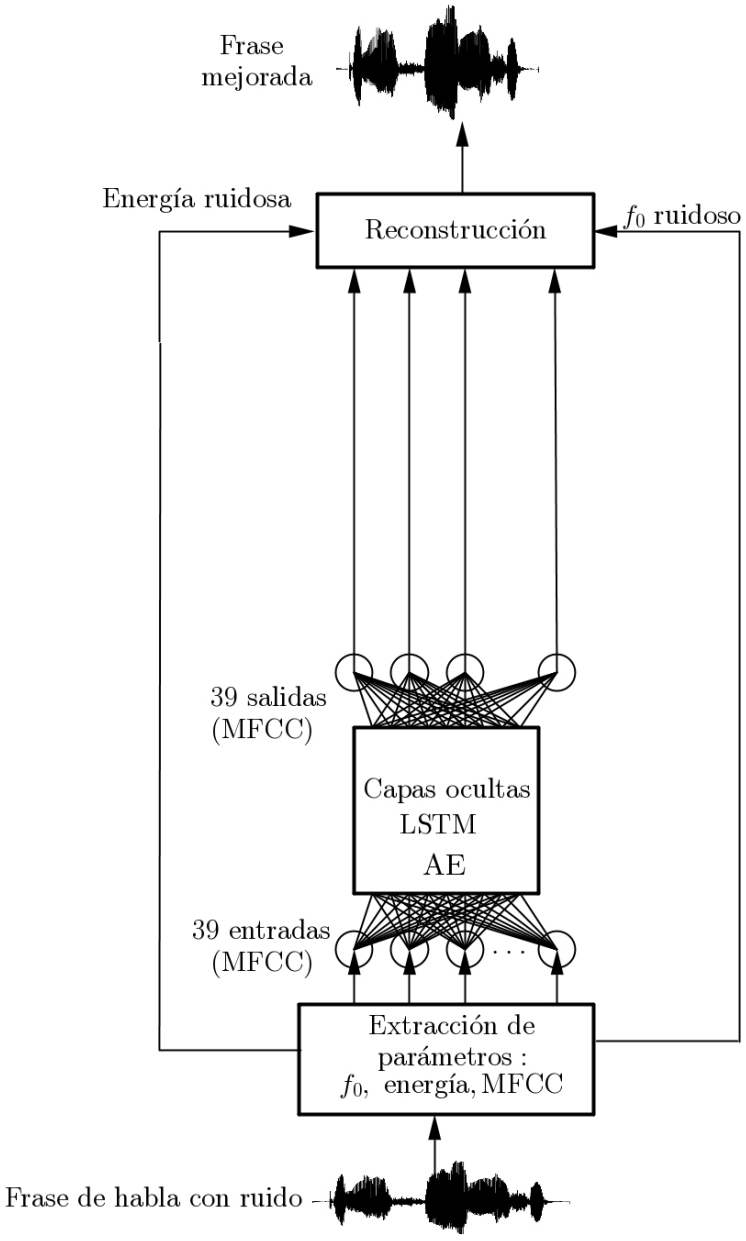



**Figura 9.4:** Sistema DLSTM-1 para mejorar los coeficientes MFCC del habla ruidosa con un *autoencoder*

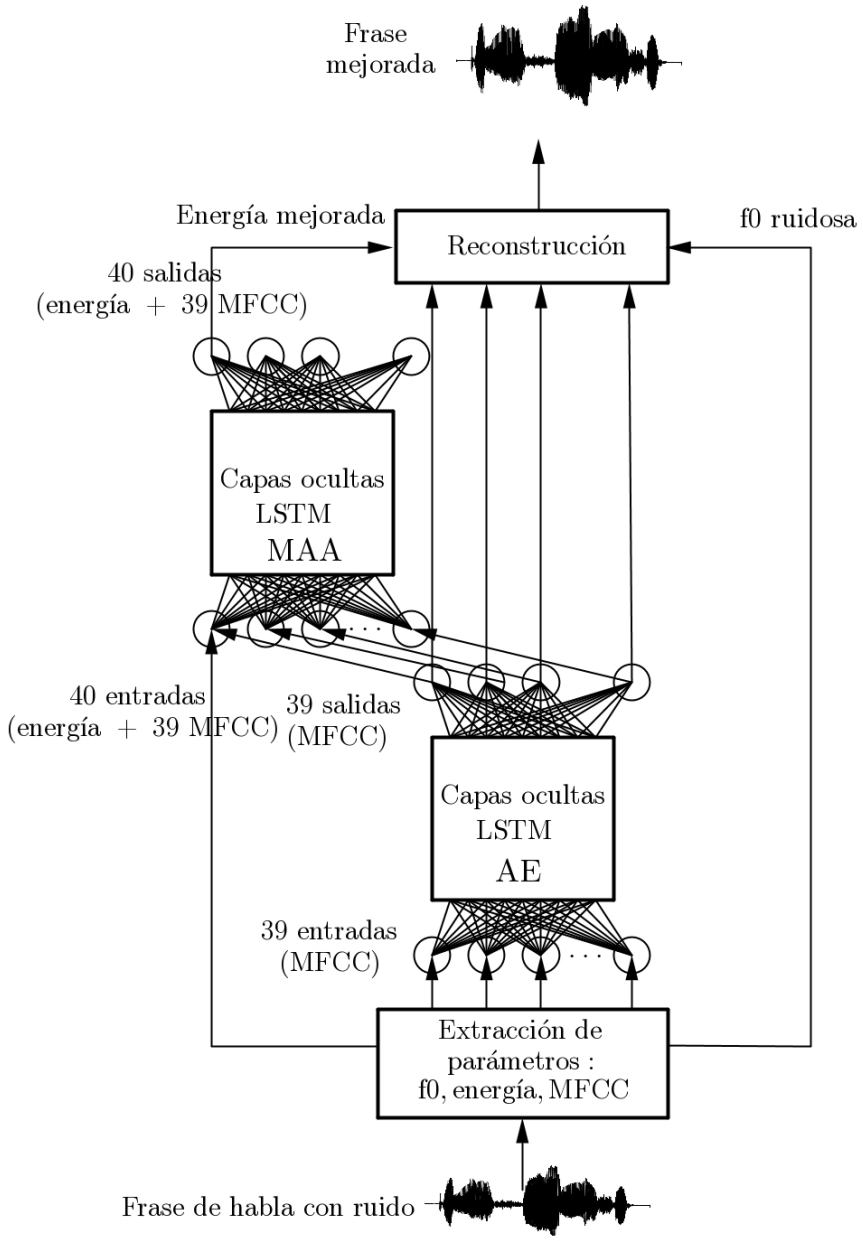



**Figura 9.5:** Sistema DLSTM-2 para mejorar los coeficientes MFCC del habla ruidosa con un *autoencoder* y el coeficiente de energía con una memoria auto-asociativa

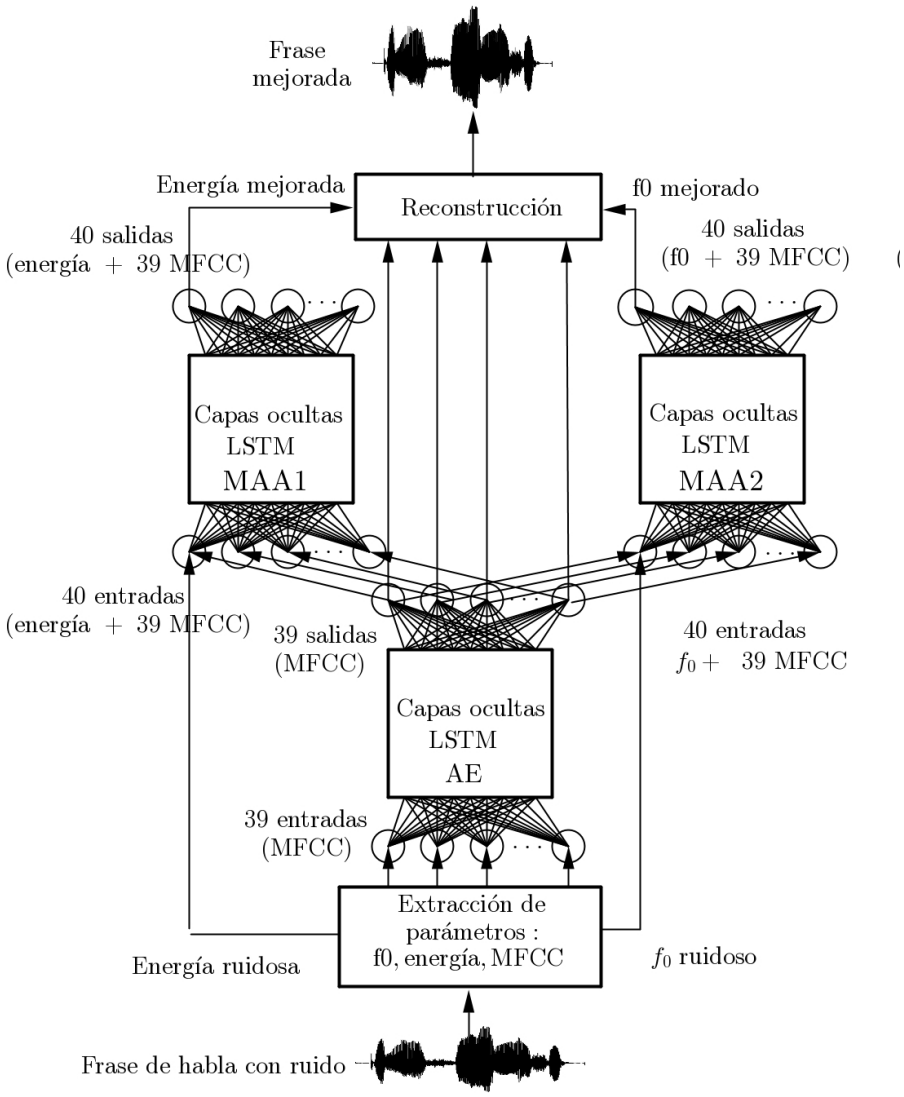



**Figura 9.6:** Sistema DLSTM-3 para mejorar los coeficientes MFCC del habla ruidosa con un *autoencoder* y los de energía y $f_0$ con memorias auto-asociativas

$$\text{Sea } D(W) = P_S(w) - P_n(w) \tag{9.4}$$

$$P'_S(w) = \begin{cases} D(w), & \text{if } D(W) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{9.5}$$

donde $P'_S(w)$ es el espectro modificado, $P_s(w)$ es el espectro de la señal de entrada (ruidosa), y $P_n(w)$ es el espectro estimado del ruido.

Desde la década de 1970, varias implementaciones de este algoritmo han aparecido en la literatura. Una implementación exitosa fue presentada en [105], al modificar la formulación original introduciendo restricciones en la aproximación del espectro del ruido, haciéndolo más robusto a los picos de esta representación. La formulación modifica las ecuaciones 9.4 y 9.5 al introducir ajustes y normalizaciones, expresados de la siguiente manera:

$$D(W) = G\left[P_s^{\gamma}(w) - \alpha P_n^{\gamma}(w)\right] \tag{9.6}$$

$$P'(s) = \begin{cases} D^{\frac{1}{\gamma}}(w), & \text{si } D^{\frac{1}{\gamma}}(w) > \beta P_n(w) \\ \beta P_n(w), & \text{otherwise} \end{cases} \tag{9.7}$$

Detalles de la implementación se pueden encontrar en la referencia [105]. Para el propósito de esta tesis, se utilizaron los siguientes parámetros, también tomados de la referencia: $\alpha$ adaptativo, el parámetro $\beta = 0.002$, la ganancia $G = 2$ y el exponente del espectro de potencia de la entrada $\gamma = 2$.

### 9.4.2 Filtro Wiener adaptativo

Las técnicas de mejora de señales de habla comúnmente estiman un factor de supresión, ajustado para cada componente en frecuencia con un SNR calculado *a posteriori*. Algunas técnicas recientes han incluido una estimación *a priori* del SNR para el cálculo del ajuste en el proceso de eliminación de ruido [64]. Estos factores se pueden estimar utilizando varias técnicas, por ejemplo una estimación Wiener.

Se utiliza en la presente tesis una implementación de esta aproximación que define una densidad del espectro de potencia del ruido $\hat{P}^t(\cdot)$ en cada componente en frecuencia $f_k$ como:

$$\hat{P}^t(f_k) = \lambda \hat{P}_B^{t-1}(f_k) + (1-\lambda)|B^t(f_k)|^2 \tag{9.8}$$

donde $B$ es el espectro del ruido, $P[\cdot]$ denota una rectificación de media onda en el intervalo de tiempo $t$.

La estimación *a priori* del SNR se define como:

$$\widetilde{SNR}^t_{prio}(f_k) = (1-\beta)P[\widetilde{SNR}^t_{prio}(f_k) - 1] + \beta \frac{|\hat{S}^{t-1}(f_k)|^2}{\hat{P}_b(f_k)} \tag{9.9}$$

Mayores detalles se pueden encontrar en la referencia [64]. En la implementación utilizada en esta tesis, los parámetros principales se han fijado $\lambda = \beta = 0.98$.

### 9.4.3 *Multi-band Spectral Subtraction*

Este método, presentado en [106] considera ruidos de colores, los cuales afectan la señal de habla de forma distinta en diferentes bandas de frecuencia. Para este propósito, el espectro es dividido en bandas, y una técnica semejante a *Spectral subtraction* se aplica de forma independiente en cada banda.

En los experimentos de este capítulo se han utiliza ocho bandas espaciadas de forma lineal para calcular y realizar el procedimiento de *Spectral subtraction*.

### 9.4.4 *Generalized Subspace Approach*

Una estimación lineal de la señal de habla limpia $X$ a partir de una versión ruidosa de ésta se puede escribir $\hat{X} = H \cdot Y$, donde $H$ es una matriz. La idea tras este método es encontrar $H$ de manera que el error $\hat{X} - X$ se minimice. En otras palabras, estimar la señal limpia al remover los componentes del ruido y reteniendo solamente los de la señal limpia [107].

Varios métodos se han propuesto para estimar $H$ [108][109]. Para la presente tesis, empleamos un enfoque reciente basado en matrices de covarianza de la señal de habla y el ruido. Utilizamos los mismos parámetros descritos en [107].

### 9.4.5 *Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator*

Hay una clase de sistemas para el mejoramiento de habla ruidosa que dan una mayor importancia al análisis del espectro en el corto tiempo (STSA, por las siglas de *Short-Time Spectral Amplitude*). Los métodos basados en una descomposición de la señal, como los descritos en las subsecciones previas, no están optimizados para STSA.

Algunos modelos estadísticos se han propuesto para derivar un estimador, asumiendo las distribuciones de probabilidad de los coeficientes de Fourier. Por ejemplo, el modelo presentado en [110], el cual incluye una estimación de la exponencial compleja de la fase. Los parámetros principales del método son la probabilidad de ausencia de señal en el componente espectral $k-$ésimo $q_k$ y el factor de ajuste $\alpha$. Para los experimentos realizados en este tesis, se utiliza $q_k = 0.3$ y $\alpha = 0.98$.

### 9.4.6 *Log-Spectral Amplitude Estimator*

Este método se basa en una estimación del espectro de la señal limpia, utilizando un modelo de la señal en ventanas $X$, junto con un ruido aditivo $D$, de manera que la señal ruidosa se puede expresar $Y(k) = X(k) + D(k)$. Consideraciones estadísticas se introducen incorporando una probabilidad *a priori* de la ausencia de habla y una estimación del SNR para estimar el espectro de la señal en presencia o ausencia de habla.

La implementación utilizada en esta tesis fue presentada en [111], donde un estimador modificado óptimamente usando parámetros de la energía del habla en ventanas vecinas fue introducido. En nuestros experimentos se utilizan los mismos parámetros presentados en dicha referencia para la probabilidad $q_k$, con $\alpha = 0.98$.

### 9.4.7 Procedimiento experimental

Se describirán a continuación las condiciones experimentales seguidas en la presente sección. El procedimiento total, desde la generación de los datos hasta la evaluación, se puede resumir en los siguientes pasos:

1. Base de datos: La base de datos utilizada para la experimentación fue la CMU ARCTIC del *Language Technologies Institute* de la Universidad Carnegie Mellon [112]. Como se describió en la Sección 4.3.1, esta base de datos cuenta con cinco voces. Para efectos de la presente sección, se seleccionó una de las voces disponibles: SLT, la cual es una voz femenina.

2. Generación del ruido: La información de los diferentes tipos de ruido fue generada y aplicada a los archivos de la base de datos para un SNR específico. Tres tipos de ruido fueron generados: Ruido Blanco, Ruido Rosa y Ruido Babble. Para cada uno de éstos, se añadieron cinco niveles SNR, de manera que los audios fueran afectados desde un nivel leve hasta uno fuerte. Tanto el Ruido Blanco como el Rosa son ruidos artificiales, mientras que Babble es un ruido producido a partir de una grabación de un ambiente natural. La descripción de este último se puede encontrar en [113].

3. Extracción de características y correspondencia de entrada y salida: Para procesar los archivos de audio, la base de datos fue resampleada a una frecuencia de 16 kHz, con 16 bits, de manera que pudiera aplicarse el sistema Ahocoder. Un conjunto de parámetros fueron extraídos tanto del habla limpia como la ruidosa: 39 coeficientes MFCC que representan el espectro, un coeficiente de energía y uno de $f_0$. Cada ventana de audio está representada por un vector de dimensión 41:

$$V_k = [f_0^k, e^k, mfcc_1^k, \ldots, mfcc_{39}^k] \tag{9.10}$$

   Los parámetros del habla ruidosa fueron utilizados como entradas a las redes LSTM, mientras que sus correspondientes limpios constituyen las salidas. Al finalizar el proceso, con los parámetros procesados es posible reconstruir una forma de onda con el sistema Ahodecoder.

4. Entrenamiento: Durante el proceso de entrenamiento, los pesos de las redes LSTM se ajustan conforme se van presentando los ejemplos de habla ruidosa y limpia a la entrada y la salida. Dado que los audios tienen distinta duración, los pares entrada/salida constituyen secuencia de longitud variable.

5. Prueba: Un subconjunto de cincuenta frases fueron seleccionadas para realizar la evaluación. Estas frases no tomaron parte del proceso de entrenamiento, para garantizar independencia entre los procesos de entrenamiento y prueba. Las mismas cincuenta frases fueron seleccionadas para aplicarles los algoritmos de comparación considerados en la presente sección.

### 9.4.8 Evaluación

Las medidas objetivas utilizadas para evaluar los resultados de la presente sección coinciden con las descritas en la Sección 4.4.

### 9.4.9 Nomenclatura

Para definir la cantidad de unidades en las capas ocultas de las redes LSTM se siguió un proceso de prueba y error. Dadas las redes definidas en la Sección 4.3.3, se partió de una cantidad de tres capas ocultas y se seleccionaron doscientas unidades en cada una. El total de algoritmos considerados en la propuesta y a manera de comparación se muestra en la Tabla 9.1. En aquellos casos (DLSTM-1 y DLSTM-2) en los cuales se procesa solamente un subconjunto de los parámetros, en la reconstrucción de la señal se toman los parámetros ruidosos que hacen falta para generar la forma de onda.

**Tabla 9.1:** Nomenclatura de los algoritmos

| Algoritmo | Nomenclatura |
|---|---|
| specsub | *Spectral subtraction* |
| wiener | Filtro Wiener adaptativo |
| mss | *Multi-band Spectral Subtraction* |
| klt | *Generalized Subspace Approach* |
| mmse | *Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator* |
| logmmse | *Log-Spectral Amplitude Estimator* |
| DLSTM-S | Un solo *autoencoder* LSTM para procesar simultáneamente todos los parámetros |
| DLSTM-1 | Un *autoencoder* para procesar solamente los MFCC |
| DLSTM-2 | Un *autoencoder* para los MFCC y una memoria auto-asociativa para el coeficiente de energía |
| DLSTM-3 | Un *autoencoder* para los MFCC, y dos memorias auto-asociativas para los coeficientes de energía y $f_0$ |

## 9.5 Resultados y discusión

Los resultados están organizados en dos partes. En la primera parte (Sección 9.5.1) se presentan los resultados únicamente de los sistemas propuestos basados en LSTM, para comparar su desempeño de acuerdo con las medidas objetivas propuestas. En la segunda parte (Sección 9.5.2) los mejores resultados de los LSTM se comparan con los seis algoritmos basados en procesamiento digital de señales.

### 9.5.1 Sistemas LSTM

Los resultados de la medida WSS para los tres tipos de ruido y los cinco niveles de intensidad se presentan en la Tabla 9.2. DLSTM-2 obtiene los mejores resultados para la mayoría de los casos, pero es importante notar cómo DLSTM-3 lo supera en el nivel más alto de ruido para los tres tipos considerados.

La propuesta DLSTM-S, la cual transforma todos los parámetros de forma simultánea con una sola red LSTM, da resultados desfavorables comparado con el resto de propuestas, con excepción de un nivel de Ruido Babble (SNR -10).

Estos resultados indican en los niveles más altos de ruido, donde la detección de $f_0$ se ve afectada (y produce valores 0 en la mayoría de los casos), todos los sistemas, con excepción de DLSTM-3 se ven afectados en la reconstrucción. En contraste con este hecho, DLSTM-3 es capaz de reconstruir una versión de $f_0$ a partir de los MFCC mejorados, la cual no es exactamente la versión original, sino una aproximada. Esto se puede observar en la Figura 9.7, donde se realiza la reconstrucción de $f_0$ a partir de los MFCC aún cuando no se haya podido detectar ningún valor de este parámetro.

La versión reconstruida de $f_0$ en DLSTM-3 puede significar la obtención de una versión alterna del habla que tiene desventaja cuando se aplican las medidas objetivas al habla procesada. Sin embargo, puede observarse una ventaja en el sistema ASR, dado que los resultados de DLSTM-3 tienen un WER mejor en presencia de ruidos altos que los demás sistemas. Esto se puede observar en la Tabla 9.3, donde los sistemas DLSTM-2 y DLSTM-3 presentan los mejores resultados en todos los casos.

Se puede notar, sin embargo, que para el nivel de ruido menor (SNR-10), ninguno de los sistemas propuestos basados en LSTM mejoran el WER de la señal ruidosa. Esto puede indicar una limitación de la propuesta con respecto aesta medida de evaluación. En la Figura 9.8 se realiza

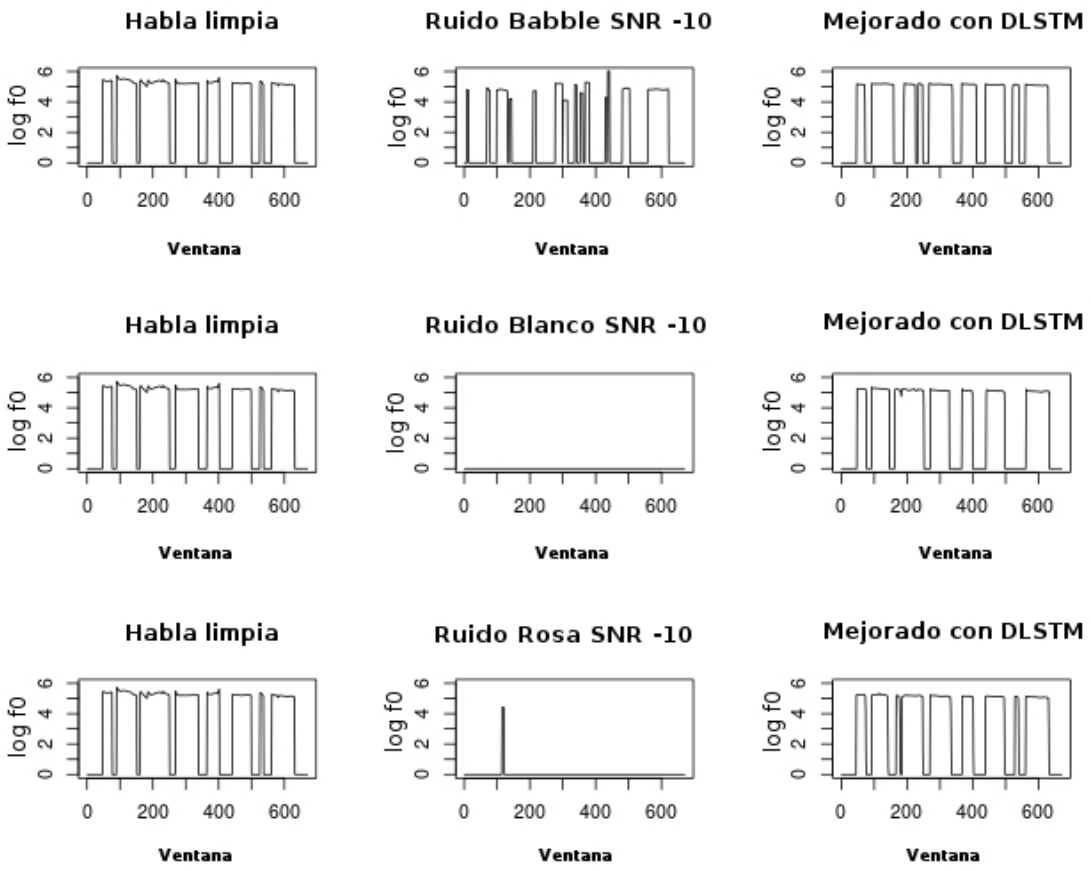



**Figura 9.7:** Reconstrucción de $f_0$ con la memoria auto-asociativa para niveles altos de ruido

una comparación de los espectrogramas para los sistemas LSTM propuestos. Se evidencia cómo DLSTM-2 y DLSTM-3 reducen el ruido en los espectrogramas, en comparación con DLSTM-1.

Los resultados de la medida PESQ se muestran en la Tabla 9.4. En ésta, DLSTM-2 da los mejores resultados para la mayoría de los casos, con excepción del Ruido Blanco con SNR-10, donde DLSTM-3 tiene mejor resultado. En todos los casos, DLSTM-2 y DLSTM-3 mejoran los valores PESQ de la señal ruidosa.

Los resultados de las propuestas con respecto a la medida $\text{SegSNR}_\text{f}$ se muestran en la Tabla 9.5. Éstos son semejantes a los presentados en las medidas previas: DLSTM-3 obtiene mejores resultados en los niveles más altos de ruido, mientras que DLSTM-2 los obtiene para el resto de los casos.

Como se mencionó anteriormente, una posible explicación para estos resultados es el hecho de que DLSTM-3 requiere la extracción de $f_0$, y este proceso falla para algunos niveles de ruido. Por ejemplo, en la Figura 9.9, se muestran los contornos de $f_0$ para una de las frases, a diferentes niveles de Ruido Blanco. Se puede observar cómo a niveles de ruido SNR-10, SNR-5 y SNR 0

**Tabla 9.2:** Resultados WSS para los sistemas basados en LSTM. Los valores menores son mejores resultados. * es el mejor.

| | | Propuesta | | | | |
|---|---|---|---|---|---|---|
| Ruido | SNR | Ninguno | DLSTM-S | DLSTM-1 | DLSTM-2 | DLSTM-3 |
| Blanco | -10 | 63.88 | 52.89 | 75.89 | 72.86 | 41.40* |
| | -5 | 50.60 | 72.04 | 58.72 | 55.69* | 70.26 |
| | 0 | 30.71 | 60.82 | 30.47 | 27.47* | 47.88 |
| | 5 | 24.47 | 64.80 | 21.51 | 20.94* | 52.67 |
| | 10 | 19.60 | 53.29 | 16.80* | 16.85 | 24.96 |
| | Prom. | 37.85 | 60.77 | 40.68 | 38.76* | 47.44 |
| Rosa | -10 | 63.86 | 66.89 | 73.38 | 71.11 | 59.51* |
| | -5 | 50.60 | 47.47 | 46.02 | 41.06* | 72.07 |
| | 0 | 30.72 | 59.07 | 24.95 | 23.31* | 77.18 |
| | 5 | 24.49 | 37.06 | 19.74 | 19.50* | 60.75 |
| | 10 | 19.61 | 33.86 | 14.55* | 15.09 | 49.48 |
| | Prom. | 37.86 | 48.87 | 35.73 | 34.02* | 63.80 |
| Babble | -10 | 63.87 | 64.92* | 72.48 | 73.82 | 72.00 |
| | -5 | 50.61 | 81.93 | 61.35 | 60.61* | 78.93 |
| | 0 | 30.71 | 94.68 | 48.76* | 49.94 | 102.60 |
| | 5 | 24.48 | 39.05 | 30.01* | 30.65 | 48.18 |
| | 10 | 19.60 | 45.20 | 17.57 | 17.18* | 40.98 |
| | Prom. | 37.85 | 65.16 | 46.04* | 46.44 | 68.54 |

mucha de la información sobre este parámetro se pierde. Resultados semejantes se obtienen para los otros tipos de ruido.

Dado este problema de determinar el valor de $f_0$ para niveles altos de ruido, la mayoría de la información de este parámetro se infiere de los MFCC en la memoria auto-asociativa que se incorpora en DLSTM-3. En la Figura 9.10 se muestra cómo esta memoria auto-asociativa reconstruye los valores de $f_0$ en el nivel de ruido más alto, a pesar de que el resultado difiere del original en su variabilidad. Esto puede llegar a afectar los resultados de las medidas objetivas cuando se trata de DLSTM-3.

Las siguientes dos secciones presentan los resultados obtenidos con los seis algoritmos utilizados para comparación. Estos resultados se compararán con los de DLSTM-3 para los niveles de ruido DLSTM-3, y con los de DLSTM-2 para el resto de éstos. En las tablas, esta selección se denotará como DLSTM. En la primera sección se compararan los algoritmos entre sí para determinar cuál es el mejor y en qué casos los demás no presentan resultados que difieran de forma significativa. En la sección final se aborda el problema de determinar en qué casos los algoritmos mejoran significativamente la señal ruidosa.

**Tabla 9.3:** Resultados WER para los sistemas basados en LSTM. Los valores menores indican mejores resultados. * es el mejor resultado.

| | | Propuesta | | | | |
|---|---|---|---|---|---|---|
| Ruido | SNR | Ninguno | DLSTM-S | DLSTM-1 | DLSTM-2 | DLSTM-3 |
| Blanco | -10 | 100 | 100 | 97.46 | 100 | 93.01* |
| | -5 | 100 | 92.37 | 92.58 | 89.62* | 91.1 |
| | 0 | 97.23 | 93.22 | 91.31 | 82.42 | 81.36* |
| | 5 | 68.87 | 67.37 | 72.25 | 58.47 | 57.84* |
| | 10 | 28.36 | 34.75 | 51.48 | 28.81* | 29.03 |
| | Prom. | 78.89 | 77.54 | 81.02 | 71.86 | 70.47* |
| Rosa | -10 | 100 | 100 | 96.82 | 95.97 | 94.07* |
| | -5 | 100 | 89.83 | 91.1 | 91.1 | 88.98* |
| | 0 | 85.07 | 73.09 | 79.24 | 52.12* | 58.05 |
| | 5 | 27.29 | 53.18 | 53.6 | 26.06* | 31.36 |
| | 10 | 9.17 | 17.37 | 23.09 | 13.77* | 21.61 |
| | Prom. | 64.31 | 66.69 | 68.77 | 55.80* | 58.81 |
| Babble | -10 | 100 | 99.15 | 95.76 | 96.19 | 94.49* |
| | -5 | 95.13 | 92.58 | 88.56 | 81.36 | 80.51* |
| | 0 | 51.06 | 76.48 | 75.85 | 60.81* | 65.25 |
| | 5 | 16.74 | 43.43 | 37.29 | 23.73* | 26.06 |
| | 10 | 7.42 | 17.16 | 12.71 | 8.69* | 9.96 |
| | Prom. | 54.07 | 65.76 | 62.03 | 54.16* | 55.25 |

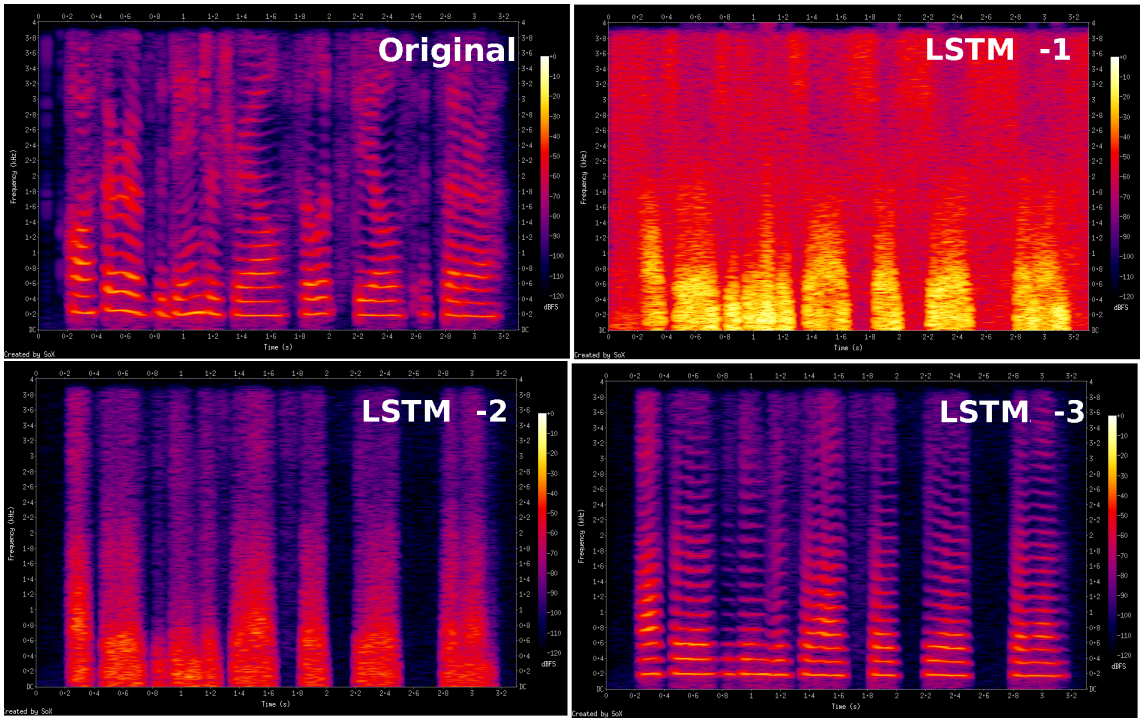



**Figura 9.8:** Espectrogramas de los tipos de DLSTM para Ruido Blanco con SNR -10

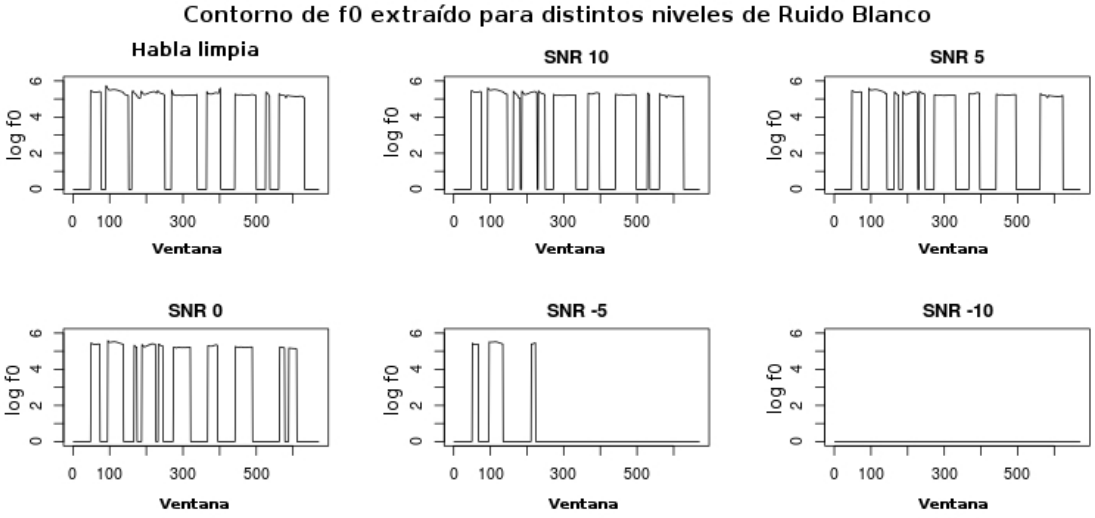



**Figura 9.9:** Detección de $f_0$ para diferentes niveles de Ruido Blanco

**Tabla 9.4:** Resultados de PESQ para los sistemas basados en LSTM. Los valores mayores representan mejores resultados. * es el mejor.

| | | Procedimiento | | | | |
|---|---|---|---|---|---|---|
| Ruido | SNR | Ninguno | DLSTM-S | DLSTM-1 | DLSTM-2 | DLSTM-3 |
| Blanco | -10 | 0.83 | 0.97 | 0.68 | 0.80 | 1.10* |
| | -5 | 0.96 | 0.67 | 1.17 | 1.30* | 0.70 |
| | 0 | 1.30 | 0.79 | 1.85 | 2.03* | 0.83 |
| | 5 | 1.62 | 0.69 | 2.36 | 2.57* | 0.85 |
| | 10 | 2.10 | 0.86 | 2.73 | 3.00* | 2.20 |
| | Prom. | 1.36 | 0.79 | 1.76 | 1.94* | 1.14 |
| Rosa | -10 | 0.83 | 0.64 | 0.67 | 0.80* | 0.67 |
| | -5 | 0.96 | 1.00 | 0.54 | 1.51* | 1.31 |
| | 0 | 1.30 | 0.75 | 0.61 | 2.32* | 2.06 |
| | 5 | 1.62 | 1.32 | 0.72 | 2.74* | 2.48 |
| | 10 | 2.10 | 1.47 | 1.02 | 3.33* | 2.96 |
| | Prom. | 1.36 | 1.03 | 0.71 | 2.14* | 1.90 |
| Babble | -10 | 0.82 | 0.54 | 0.48 | 0.70* | 0.59 |
| | -5 | 0.96 | 0.45 | 0.48 | 0.99* | 0.82 |
| | 0 | 1.30 | 0.37 | 0.43 | 1.33* | 1.17 |
| | 5 | 1.77 | 1.34 | 0.94 | 1.95* | 1.73 |
| | 10 | 2.35 | 0.98 | 1.08 | 3.08* | 2.79 |
| | Prom. | 1.44 | 0.73 | 0.68 | 1.61* | 1.42 |

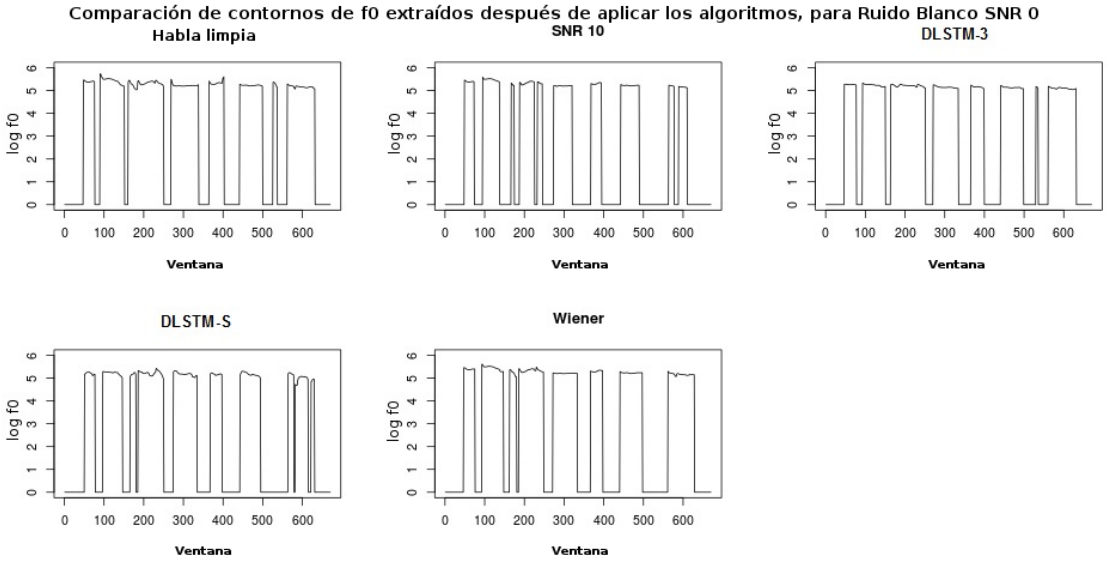



**Figura 9.10:** Reconstrucción de $f_0$ con diversos algoritmos

**Tabla 9.5:** Resultados de $\text{SegSNR}_{\text{f}}$ para los sistemas basados en LSTM. Los valores mayores representan mejores resultados. * es el mejor.

| | | Procedimiento | | | | |
|---|---|---|---|---|---|---|
| Ruido | SNR | Ninguno | DLSTM-S | DLSTM-1 | DLSTM-2 | DLSTM-3 |
| Blanco | -10 | -3.81 | 0.88 | -7.39 | -0.16 | 2.57* |
| | -5 | -0.53 | 0.87 | -2.92 | 0.95* | 0.76 |
| | 0 | 4.85 | 1.04 | 1.84 | 5.07* | 2.50 |
| | 5 | 10.67 | 1.19 | 5.72 | 10.49* | 2.31 |
| | 10 | 15.24 | 2.49 | 7.01 | 12.55* | 7.28 |
| | Prom. | 5.28 | 1.29 | 0.85 | 5.78* | 3.08 |
| Rosa | -10 | -3.81 | -0.24 | -3.14 | -0.15 | 0.88* |
| | -5 | -0.53 | 1.93 | 0.59 | 1.46* | 0.56 |
| | 0 | 4.85 | 1.46 | 4.03 | 7.38* | 0.67 |
| | 5 | 10.67 | 3.89 | 5.37 | 10.66* | 1.57 |
| | 10 | 15.24 | 4.76 | 7.41 | 13.10* | 3.31 |
| | Prom. | 5.28 | 2.36 | 2.85 | 6.49* | 1.40 |
| Babble | -10 | -3.81 | 0.29 | -0.08 | 0.37* | 0.37* |
| | -5 | 0.53 | -0.47 | 0.96 | 1.21* | 0.20 |
| | 0 | 4.85 | -0.01 | 1.53 | 1.94* | -0.53 |
| | 5 | 6.13 | 2.92 | 3.19 | 5.07* | 2.44 |
| | 10 | 15.24 | 2.81 | 7.72 | 11.15* | 3.31 |
| | Prom. | 4.59 | 1.11 | 2.67 | 3.95* | 1.16 |

### 9.5.2 Comparación con algoritmos basados en procesamiento de señales

En esta sección se aplican las medidas WSS, PESQ, $\text{SegSNR}_f$ y WER a los seis algoritmos considerados para comparación. Esta comparación se realiza para determinar las diferencias entre ellos.

Para determinar la significancia de estas diferencias, se realiza una prueba ANOVA a los conjuntos de medidas de cada algoritmo y las medidas del algoritmo que presentó el mejor resultado en las cincuenta frases de prueba. Como se realizó previamente, en las tablas se presentan los resultados de acuerdo con la medida objetiva, el tipo de ruido y el SNR.

En la Tabla 9.6 se muestran los resultados de la medida WSS. Se puede observar que DLSTM y el algoritmo mmse obtienen los mejores resultados para todos los niveles de Ruido Blanco y Rosa, mientras que para el Ruido Babble, ambos algoritmos, junto con mss, obtuvieron los mejores.

**Tabla 9.6:** Resultados WSS para todos los algoritmos. Los valores más bajos representan mejores resultados. * es el mejor resultado. En negrita las medidas que no difieren significativamente del mejor.

| | | | Algoritmo | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | SNR | Ninguno | DLSTM | mss | klt | logmse | mmse | specsub | wiener |
| Blanco | -10 | 63.88 | 41.40* | 63.32 | 59.29 | 60.83 | 46.31 | 79.22 | 50.66 |
| | -5 | 50.61 | 55.69 | 47.60 | 40.39 | 50.28 | 33.81* | 68.07 | 39.10 |
| | 0 | 30.71 | **27.47** | 30.32 | 29.57 | 36.13 | 26.68* | 50.31 | 30.19 |
| | 5 | 24.47 | 20.94* | **23.41** | **22.16** | 27.58 | **23.50** | 29.61 | **23.13** |
| | 10 | 19.60 | 16.85* | 19.94 | **16.72** | 23.76 | 19.37 | 20.76 | **17.54** |
| | Prom. | 37.85 | 32.47 | 36.92 | 33.63 | 39.72 | 29.93* | 49.59 | 32.12 |
| Rosa | -10 | 63.86 | 59.51 | 64.12 | 67.78 | 63.86 | 55.70* | 81.56 | **56.93** |
| | -5 | 50.60 | 46.02 | 45.31 | 47.48 | 59.96 | 41.67* | 74.49 | **42.91** |
| | 0 | 30.72 | 23.31* | 29.48 | 34.89 | 55.82 | 32.84 | 52.68 | 33.85 |
| | 5 | 24.49 | 19.50* | 24.28 | 25.18 | 38.98 | 25.02 | 31.62 | 25.19 |
| | 10 | 19.61 | 15.09* | 20.71 | 18.37 | 25.49 | 20.26 | 20.85 | 17.35 |
| | Prom. | 37.86 | 34.02* | 36.78 | 38.74 | 48.82 | 35.10 | 52.24 | 35.25 |
| Babble | -10 | 63.87 | 72.00* | 76.46 | 77.81 | 86.43 | 77.80 | 85.38 | 79.56 |
| | -5 | 50.61 | 60.61 | **56.29** | 62.31 | 67.07 | 55.32* | 72.14 | 61.51 |
| | 0 | 30.71 | 49.94 | 39.78* | 44.68 | 48.10 | **40.62** | 53.42 | 44.62 |
| | 5 | 24.48 | 30.65 | 28.94* | 30.84 | 32.04 | **30.38** | 32.74 | 31.30 |
| | 10 | 19.60 | 17.18* | 21.68 | 21.39 | 22.77 | 22.31 | 22.64 | 22.27 |
| | Prom. | 37.85 | 46.44 | 44.63* | 47.41 | 51.28 | 45.29 | 53.26 | 47.85 |

Los resultados para WER en el sistema ASR se muestran en la Tabla 9.7. En ésta, DLSTM obtiene los mejores resultados para los niveles altos de ruido, mientras que el filtro Wiener mejora los resultados de éste en casi todos los demás casos. Se puede observara también que para los valores de baja intensidad de ruido Babble, ninguna de los algoritmos mejoró el WER de la señal ruidosa. Dado que el WER se evalúa sobre el conjunto de cincuenta frases, en este caso no es posible realizar una prueba estadística para determinar diferencias significativas entre algoritmos.

**Tabla 9.7:** Resultados de WER para todos los algoritmos de eliminación de ruido. Los valores menores representan mejores resultados. * es el mejor.

| | | | Algoritmo | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | SNR | Ninguno | DLSTM | mss | klt | logmse | mmse | specsub | wiener |
| Blanco | -10 | 100 | 93.01* | 100 | 100 | 100 | 100 | 100 | 100 |
| | -5 | 100 | 89.62* | 97.23 | 100 | 90.83 | 100 | 100 | 92.96 |
| | 0 | 97.23 | 82.42 | 91.26 | 98.72 | 80.6 | 95.74 | 92.58 | 79.32* |
| | 5 | 68.87 | 58.47 | 68.23 | 86.99 | 59.49 | 75.27 | 84.53 | 42.86* |
| | 10 | 28.36 | 28.81 | 33.69 | 62.9 | 22.17 | 43.28 | 42.16 | 17.48* |
| | Prom. | 78.89 | 71.86 | 78.08 | 89.72 | 70.62 | 82.86 | 83.85 | 66.52* |
| Rosa | -10 | 100 | 94.07* | 100 | 100 | 100 | 100 | 100 | 99.15 |
| | -5 | 100 | 91.1* | 100 | 100 | 94.88 | 100 | 100 | 93.18 |
| | 0 | 85.07 | 52.12* | 78.89 | 98.08 | 70.36 | 94.24 | 89.41 | 56.08 |
| | 5 | 27.29 | 26.06 | 26.23 | 80.17 | 25.8 | 47.97 | 59.53 | 20.68* |
| | 10 | 9.17 | 13.77 | 10.87 | 37.31 | 11.51 | 24.52 | 24.15 | 8.53* |
| | Prom. | 64.31 | 55.80 | 63.20 | 83.11 | 60.51 | 73.35 | 74.62 | 55.52* |
| Babble | -10 | 100 | 94.49* | 100 | 100 | 100 | 100 | 100 | 100 |
| | -5 | 95.13 | 81.36* | 99.36 | 98.52 | 96.19 | 100 | 94.7 | 97.25 |
| | 0 | 51.06 | 60.81 | 76.06 | 75.64 | 76.06 | 89.83 | 59.32* | 62.29 |
| | 5 | 16.74 | 23.73 | 30.08 | 33.9 | 22.25 | 34.11 | 27.97 | 17.37* |
| | 10 | 7.42 | 8.69 | 11.44 | 18.86 | 7.42* | 12.71 | 9.32 | 8.05 |
| | Prom. | 54.07 | 54.16* | 63.39 | 65.38 | 60.38 | 67.33 | 58.26 | 56.99 |

Los espectrogramas que se muestran en la Figura 9.11 reflejan la mejora que se obtiene con DLSTM en los niveles altos de ruido, en comparación con el Filtro Wiener.

Los resultados de la medida PESQ se presentan en la Tabla 9.8. Para el caso de Ruido Blanco, el algoritmo klt obtuvo el mejor resultado para todos los niveles, con excepción de SNR -10, para el cual DLSTM fue el mejor. Para el Ruido Rosa, DLSTM obtiene el mejor resultado para todos los niveles, con excepción de SNR -10. Para Ruido Babble, mss es el mejor en tres de los cinco niveles SNR, a pesar de que mmse y Wiener no son significativamente diferentes del mejor, y de forma semejante con DLSTM.

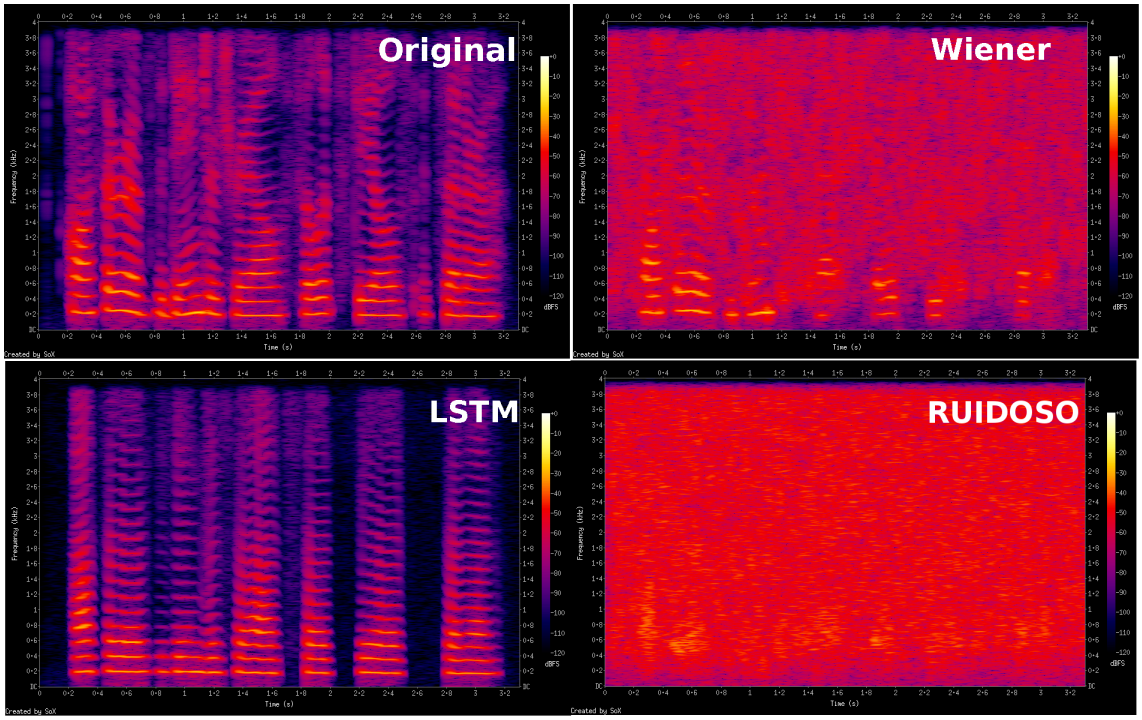



**Figura 9.11:** Espectrogramas de la señal original, ruidosa y procesada con los algoritmos Wiener y DLSTM para el Ruido Blanco con SNR -10

Los resultados para $\text{SegSNR}_\text{f}$ se muestran en la Tabla 9.9. En ésta, se observa cómo el filtro Wiener generalmente obtiene los mejores resultados, a pesar de que DLSTM continúa dando buenos resultados para Ruido Rosa, así como los mejores para todos los niveles altos de ruido.

### 9.5.3 Análisis de significancia estadística con respecto a la señal ruidosa

En esta sección se presenta un análisis estadístico para determinar cuáles de los resultados de los algoritmos utilizados en las secciones previas realizan una mejora estadísticamente significativa con respecto a la señal ruidosa, en las medidas objetivas utilizadas. Este análisis se realiza para distinguir aquellos casos para los que se han identificado los mejores resultados, pero estos valores no son una mejoría de las medidas en la señal con ruido, de aquellos que sí representan una mejora. Para el análisis estadístico se aplicó la Prueba HSD de Tukey, con comparaciones entre el conjunto que obtuvo la mejor media, con el resto.

En la Tabla 9.10 se observa cómo DLSTM obtiene más mejoras estadísticamente significativas para la medida WSS que el resto de algoritmos. Más aún, realiza mejoras para al menos un nivel

**Tabla 9.8:** Resultados para PESQ para todos los algoritmos de eliminación de ruido. Los valores mayores representan mejores resultados. * es el mejor. En negrita las medidas que no difieren significativamente del mejor.

| | | | Algoritmo | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | SNR | Ninguno | DLSTM | mss | klt | logmse | mmse | specsub | wiener |
| Blanco | -10 | 0.83 | 1.10* | 0.84 | 0.75 | 0.34 | 0.70 | 0.50 | 0.90 |
| | -5 | 0.96 | 1.30 | 0.93 | 1.75* | 0.75 | 1.54 | 0.61 | 1.32 |
| | 0 | 1.30 | 2.03 | 1.28 | 2.30* | 1.56 | **2.19** | 1.22 | 1.85 |
| | 5 | 1.62 | 2.57 | 1.92 | 2.71* | 2.12 | 2.49 | 2.16 | 2.33 |
| | 10 | 2.10 | **3.00** | 2.43 | 3.06* | 2.37 | 2.85 | 2.79 | 2.70 |
| | Prom. | 1.36 | 1.94 | 1.48 | 2.11* | 1.43 | 1.95 | 1.46 | 1.82 |
| Rosa | -10 | 0.83 | 0.67 | 0.79 | 0.74 | 1.02* | **0.97** | 0.60 | 0.92 |
| | -5 | 0.96 | 1.51* | 0.99 | 1.60 | 1.30 | **1.43** | 0.85 | 1.32 |
| | 0 | 1.30 | 2.32* | 1.52 | **2.16** | 1.00 | 1.93 | 1.40 | 1.87 |
| | 5 | 1.62 | 2.74* | 2.06 | **2.63** | 1.51 | 2.42 | 2.30 | 2.41 |
| | 10 | 2.10 | 3.33* | 2.52 | **3.00** | 2.36 | 2.90 | 2.87 | 2.84 |
| | Prom. | 1.36 | 2.14* | 1.58 | 2.03 | 1.44 | 1.93 | 1.61 | 1.87 |
| Babble | -10 | 0.82 | 0.48 | **0.59** | 0.39 | 0.29 | 0.62* | 0.27 | **0.58** |
| | -5 | 0.96 | **0.99** | 1.16* | 0.93 | 0.71 | **1.11** | 0.72 | **1.08** |
| | 0 | 1.30 | **1.33** | 1.65* | **1.50** | **1.35** | **1.62** | **1.38** | **1.61** |
| | 5 | 1.77 | 1.95 | 2.21* | **2.11** | 2.06 | **2.17** | **2.13** | **2.08** |
| | 10 | 2.35 | 3.08* | 2.62 | 2.55 | 2.53 | 2.58 | 2.62 | 2.53 |
| | Prom. | 1.44 | 1.61 | 1.64* | 1.50 | 1.39 | 1.62 | 1.42 | 1.58 |

**Tabla 9.9:** Resultados para $\text{SegSNR}_\text{f}$ para todos los algoritmos de eliminación de ruido. Los valores mayores representan mejores resultados. * es el mejor. En negrita las medidas que no difieren significativamente del mejor.

| | | | Algoritmo | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | SNR | Ninguno | DLSTM | mss | klt | logmse | mmse | specsub | wiener |
| Blanco | -10 | -3.81 | 2.57* | -2.45 | 0.50 | 0.11 | 0.43 | 0.04 | -0.04 |
| | -5 | -0.53 | 0.95 | -0.17 | **4.06** | 0.50 | 4.49* | 0.15 | **4.30** |
| | 0 | 4.85 | 5.07 | 2.48 | 8.38 | 4.34 | **9.62** | 1.08 | 9.88* |
| | 5 | 10.67 | 10.49 | 4.67 | 12.08 | 10.18 | 12.60 | 8.29 | 13.80* |
| | 10 | 15.24 | 12.55 | 11.83 | 15.26 | 12.27 | 14.81 | 14.71 | 16.39* |
| | Prom. | 5.28 | 5.78 | 3.27 | 8.06 | 5.48 | 8.39 | 4.85 | 8.87* |
| Rosa | -10 | -3.81 | 0.88* | -1.62 | 0.57 | 0.70 | 0.40 | -0.17 | 0.35 |
| | -5 | -0.53 | 1.46 | 0.29 | 1.98 | 0.16 | 1.32 | 0.33 | 2.87* |
| | 0 | 4.85 | 7.38* | 3.64 | 3.67 | 0.39 | 3.79 | 1.43 | 5.87 |
| | 5 | 10.67 | **10.66** | 3.13 | 6.28 | 2.82 | 9.77 | 7.92 | 10.90* |
| | 10 | 15.24 | **13.10** | 7.60 | 10.32 | 10.79 | **13.26** | **13.95** | 15.07* |
| | Prom. | 5.28 | 6.49 | 2.61 | 4.56 | 2.97 | 5.71 | 4.69 | 7.01* |
| Babble | -10 | -3.81 | 0.37* | -0.05 | 0.08 | -0.15 | -0.04 | 0.10 | -0.32 |
| | -5 | 0.53 | 1.21* | 0.84 | 0.37 | 0.11 | **1.07** | 0.317 | 0.90 |
| | 0 | 4.85 | 1.94 | 3.38 | 1.93 | 2.257 | 4.48* | 1.51 | **4.46** |
| | 5 | 6.13 | 5.07 | 8.46 | 5.02 | 6.80 | **8.62** | 7.29 | 9.17* |
| | 10 | 15.24 | 11.15 | 12.98 | 9.96 | 11.52 | 12.37 | **12.85** | 13.46* |
| | Prom. | 4.59 | 3.95 | 5.12 | 3.47 | 4.11 | 5.30 | 4.41 | 5.53* |

de ruido en todos los tipos de ruido, características que supera a los demás. Es importante notar que logmse y specsub no obtuvieron mejoras significativas para ningún tipo de ruido o nivel, y que mss solo lo hizo para el caso de Ruido Rosa con SNR -5.

**Tabla 9.10:** Resultados para la medida WSS. ✓ indica una mejora significativa con respecto a la señal ruidosa

| | | Algoritmo | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SNR | DLSTM | mss | klt | logmse | mmse | specsub | wiener |
| Blanco | -10 | ✓ | | ✓ | | ✓ | | ✓ |
| | -5 | | | ✓ | | ✓ | | ✓ |
| | 0 | | | | | ✓ | | |
| | 5 | ✓ | | ✓ | | | | |
| | 10 | ✓ | | ✓ | | | | |
| Rosa | -10 | | | | | ✓ | | ✓ |
| | -5 | | ✓ | | | ✓ | | ✓ |
| | 0 | ✓ | | | | | | |
| | 5 | ✓ | | | | | | |
| | 10 | ✓ | | | | | | ✓ |
| Babble | -10 | | | | | | | |
| | -5 | | | | | | | |
| | 0 | | | | | | | |
| | 5 | | | | | | | |
| | 10 | ✓ | | | | | | |

Para el caso de la medida PESQ, en la Tabla 9.11 se puede observar que todos los algoritmos obtienen mejores resultados que para el caso de la medida WSS. DLSTM es el único de los algoritmos que mejora todos los niveles de Ruido Blanco, y junto con Wiener y mmse, obtuvieron la mayor cantidad de mejoras estadísticamente significativas.

En cuanto a la medida $\text{SegSNR}_{\text{f}}$, en los resultados de significancia estadística mostrados en la Tabla 9.12, el filtro Wiener alcanza la mayor cantidad de mejoras, seguido de mmse y luego DLSTM. Se destaca nuevamente que DLSTM obtiene mejoras significativas en los niveles altos de ruido. Por otra parte, ninguno de los algoritmos logró realizar mejoras significativas en los niveles bajos de Ruido Rosa o Ruido Babble.

Con estos resultados se observa que los sistemas DLSTM obtienen mejoras en las señales de habla con ruido que son en general competitivas con respecto a los mejores algoritmos tomados en las comparaciones. Para algunos tipos de ruido presentan resultados notablemente superiores en cuanto a mayor cantidad de mejoras en las medidas y cantidad de éstas que son estadísticamente significativas.

**Tabla 9.11:** Resultados para PESQ. ✓ indica una mejora significativa con respecto a la señal ruidosa

| | | Algoritmo | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SNR | DLSTM | mss | klt | logmse | mmse | specsub | wiener |
| Blanco | -10 | ✓ | | | | | | |
| | -5 | ✓ | | ✓ | | ✓ | | ✓ |
| | 0 | ✓ | | ✓ | ✓ | ✓ | | ✓ |
| | 5 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 10 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Rosa | -10 | | | | ✓ | | | |
| | -5 | ✓ | | ✓ | | ✓ | | ✓ |
| | 0 | ✓ | ✓ | ✓ | | ✓ | | ✓ |
| | 5 | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ |
| | 10 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Babble | -10 | | | | | | | |
| | -5 | | ✓ | | | | | |
| | 0 | | ✓ | ✓ | | ✓ | | ✓ |
| | 5 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 10 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Tabla 9.12:** Resultados para $\text{SegSNR}_{\text{f}}$. ✓ indica una mejora significativa con respecto a la señal ruidosa

| | | Enhancement | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SNR | DLSTM | mss | klt | logmse | mmse | specsub | wiener |
| Blanco | -10 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | -5 | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 0 | | | ✓ | | ✓ | | ✓ |
| | 5 | | | ✓ | | ✓ | | ✓ |
| | 10 | | | | | | | |
| Rosa | -10 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | -5 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 0 | ✓ | | | | | | ✓ |
| | 5 | | | | | | | |
| | 10 | | | | | | | |
| Babble | -10 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | -5 | ✓ | ✓ | | | ✓ | | ✓ |
| | 0 | | | | | | | |
| | 5 | | | | | | | |
| | 10 | | | | | | | |

## 9.6 Resumen de contribuciones

En este capítulo se han aplicado los sistemas propuestos para mejorar habla sintetizada, los cuales contemplan una combinación de arquitecturas de redes profundas LSTM, en la mejora de señales degradadas con diversos tipos de ruido. Se realizó una extensa comparación entre los sistemas que mejoran algunos o todos los parámetros del habla de forma separada o combinada. Los resultados también se han comparado con seis conocidos métodos de reducción de ruido basados en procesamiento digital de señales. Se realizaron varias pruebas estadísticas para establecer las diferencias significativas entre las propuestos que utilizan redes LSTM, así como la capacidad de éstas para mejorar la señal de habla con ruido.

La forma de entrenar y utilizar el conjunto de arquitecturas de redes LSTM para aplicarlo a las señales ruidosas coincide con lo propuesto para los post-filtros en síntesis de habla de esta tesis. A diferencia de lo realizado en síntesis, la degradación de la señal de voz en este caso se da para ruidos e intensidades específicos.

Dado que la propuesta basada en LSTM conlleva la parametrización y posterior reconstrucción de la forma de onda, al tratar el parámetro $f_0$ se observó que en la reconstrucción la nueva versión que se infiere principalmente de los coeficientes MFCC difiere de la original, lo cual afecta sensiblemente algunas de las medidas aplicadas. Por esta razón, se incluyó en este caso el WER de un reconocedor de propósito general, de manera que se realice una comparación orientada hacia la inteligibilidad de los resultados, con lo cual se comprobó la mayor inteligibilidad de la versión con $f_0$ reconstruida con respecto a la de los otros algoritmos.

Las medidas aplicadas muestran cómo DLSTM-2 y DLSTM-3 obtienen resultados mucho mejores que los sistemas base DLSTM-S y DLSTM-1 en la vasta mayoría de niveles y tipos de ruido. En particular, para PESQ, $\text{SegSNR}_\text{f}$, y WER, todos los mejores resultados fueron obtenidos con DLSTM-2 o DLSTM-3, en los cuales el segundo de estos destaca para los niveles más altos de ruido. Esta mejoría también ha sido constatada en los espectrogramas.

En los resultados que se comparan con los algoritmos basados en procesamiento digital de señales, se puede observar que la propuesta DLSTM es competitiva en obtener mejoras de la señal ruidosa, en comparación con los mejores casos entre los algoritmos. Esto sugiere que la propuesta realizada es efectiva en obtener mejoras para todos los tipos de ruido y los niveles considerados.

# 10

# Mejora de señales de voz con sistemas híbridos

*Este capítulo se plantea la aplicación de un sistema en dos etapas para mejorar señales de voz degradadas con ruido, combinando filtros Wiener y redes LSTM. Este combinación dio buenos resultados previamente en la mejora de voz sintetizada, y en este caso presente considerables beneficios.*

## 10.1 Introducción

En el capítulo precedente se mostró la aplicación de una combinación de arquitecturas de redes de memoria a corto y largo plazo (LSTM) para mejorar señales de voz degradadas con ruido. En éste, la propuesta se extiende para abarcar un sistema híbrido en dos etapas: En la primera se utiliza un filtro Wiener, el cual mostró buenos resultados en el capítulo anterior en la reducción de ruido, y en la segunda se aplica el conjunto de redes LSTM para mejorar la salida del filtro Wiener, resolviendo el problema de regresión necesario para mapear esta salida hacia el habla sin ruido.

Con esta extensión se desea realizar mejoras a la señal de habla con ruido, investigando en qué tipos y niveles de ruido el enfoque híbrido puede presentar mejores resultados que los sistemas individuales constituidos por Wiener y de redes LSTM.

## 10.2 Sistema propuesto

Para realizar la mejora de las frases de habla que contienen ruido, en una primera etapa se aplica el filtro Wiener, con la intención de realizar la reducción de ruido que ofrece esta técnica. La salida del filtro será en la mayoría de las casos una señal cuyos parámetros se encuentran más cercanos a los de habla natural que los ruidosos.

Por esta razón, se considera una segunda etapa, para la cual se requiere parametrizar la salida del filtro Wiener, $\overline{y}$, de manera que sus parámetros puedan ser utilizados con la colección de *autoencoders* y memorias auto-asociativas, de forma semejante a lo realizado en el capítulo anterior, pero con la versión mejorada de la señal en lugar de la señal original $y$. El proceso se esquematiza en la Figura 10.1.

Para poder realizar una comparación de la propuesta y verificar los beneficios que puede presentar, se conservará la salida del filtro Wiener, y se entrenará el conjunto de redes LSTM semejante a como se realizó en la sección anterior, para mapear directamente del habla ruidosa a la limpia. De acuerdo con la notación introducida previamente, serán tres tipos de sistemas LSTM con los cuales se realizará la comparación:

- DLSTM-1: Se utiliza una red LSTM con arquitectura *autoencoder* para mejorar los coeficientes MFCC. Se conservan tanto el coeficiente de energía como el coeficiente de $f_0$ los proveniente de la señal ruidosa.

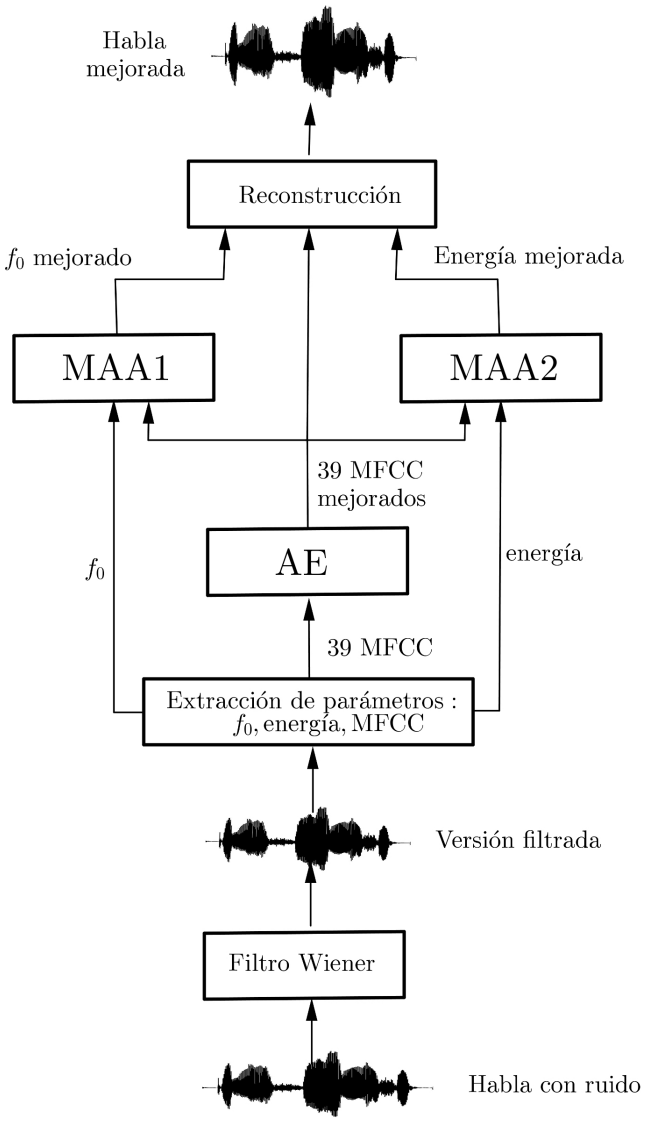



**Figura 10.1:** Sistema híbrido propuesto que contempla dos etapas para la mejora en señales de habla con ruido.

- DLSTM-2: Adicionalmente al *autoencoder* LSTM considerado en DLSTM-1, una memoria auto-asociativa se entrena de forma separada para mejorar el coeficiente de energía. El coeficiente $f_0$ se preserva de la señal ruidosa.
- DLSTM-3: Además de las dos redes consideradas en DLSTM-1 y DLSTM-2, una memoria auto-asociativa adicional se entrena para mejorar el parámetro $f_0$.

Para red LSTM se entrena a partir de los ejemplos de habla con ruido y habla limpia. En esta ocasión, cada una de las redes contiene la misma cantidad de neuronas en las tres capas ocultas.

Las propuestas de sistema híbrido presentadas en esta sección son:

- HW-DLSTM-1: En una primera etapa, la señal de habla ruidosa es filtrada utilizando Wiener. En la segunda etapa, se aplica el sistema DLSTM-1. La forma de onda finalmente se reconstruye con los coeficientes de energía y $f_0$ provenientes del filtro Wiener.

- HW-DLSTM-2: Tal como el anterior, se aplica en la primera etapa el filtro Wiener. En la segunda etapa se aplica el sistema DLSTM-2. Para la reconstrucción de la señal se utiliza el coeficiente $f_0$ proveniente del filtro Wiener.

- HW-DLSTM-3: Luego de aplicar el filtro Wiener, como en los dos casos previos, se aplica el sistema DLSTM-3.

En todos los casos híbridos se requiere un nuevo proceso de entrenamiento de todas las redes LSTM, para aprender el mapeo de los coeficientes $\overline{x}$ provenientes del filtro Wiener hacia los coeficientes $y$ del habla sin ruido, en lugar del mapeo de los coeficientes ruidosos hacia los limpios, como se realiza para los sistemas DLSTM-1, DLSTM-2 y DLSTM-3.

La forma de entrenar y aplicar las redes LSTM de los sistemas híbridos coincide con lo mostrado en los algoritmos 10 a 13 del Capítulo 9, sustituyendo el habla ruidosa por su versión filtrada utilizando Wiener.

## 10.3 Procedimiento experimental

La base de datos utilizada, la cantidad de datos y los tipos y niveles de ruido considerados coinciden con los presentados en la Sección 9.4.7. Cada frase se parametriza utilizando el sistema Ahocoder, extrayendo los vectores de parámetros de ventanas de 10 ms. El procedimiento para entrenar las redes LSTM se aceleró con hardware NVIDIA GPU, el cual requirió aproximadamente siete horas para entrenar cada red. El criterio de paro del entrenamiento se estableció con un máximo de épocas (2000) o bien el no obtener mejoras después de veinte iteraciones.

Se consideraron cinco niveles de ruido por cada tipo, a los cuales se aplicó cada uno de los siete sistemas mostrados en la Tabla 10.1.

**Tabla 10.1:** Nomenclatura de los algoritmos para los sistemas híbridos de eliminación de ruido

| Algoritmo | Nomenclatura |
|---|---|
| Wiener | Filtro Wiener adaptativo |
| DLSTM-1 | Un único *autoencoder* para los MFCC |
| DLSTM-2 | Un *autoencoder* para los MFCC y una memoria auto-asociativa para el parámetro de energía |
| DLSTM-3 | Un *autoencoder* y dos memorias auto-asociativas para los parámetros $f_0$ y energía |
| HW-DLSTM-1 | Filtro Wiener en la primera etapa, y sistema DLSTM-1 en la segunda |
| HW-DLSTM-2 | Filtro Wiener en la primera etapa, y sistema DLSTM-2 en la segunda |
| HW-DLSTM-3 | Filtro Wiener en la primera etapa, y sistema DLSTM-3 en la segunda |

## 10.4 Resultados y discusión

Los resultados están organizados en dos partes. En la primera (Sección 10.4.1) se presentan los resultados de los sistemas híbridos propuestos, así como una comparación con el filtro Wiener y los sistemas LSTM no híbridos. Se determina adicionalmente cuál de ellos resulta mejor en cada medida objetiva y cuáles de los otros casos no difieren significativamente, de acuerdo con una prueba ANOVA. En la segunda parte (sección 10.4.2) se realiza una Prueba HSD de Tukey, para determinar, en relación con la señal ruidosa, cuáles de los sistemas presentan mejoras significativas. Todas las pruebas estadísticas fueron realizadas con un nivel de significancia de 0.95.

### 10.4.1 Medidas objetivas

Los resultados para le medida WSS en los tres tipos de ruido se presentan en la Tabla 10.2. El sistema híbrido HW-DLSTM-2 obtuvo el mejor resultado para todos los niveles de ruido blanco, incluyendo mejoras significativamente mayores a los demás sistemas para los niveles SNR -10, SNR -5 y SNR 0.

Resultados similares se obtuvieron para los casos de ruido Rosa y ruido Babble, con excepción de ruido Rosa para el nivel más alto, donde DLSTM-3 obtuvo el mejor resultado, pero los sistemas híbridos HW-DLSTM-1 y HW-DLSTM-2 no tuvieron diferencias significativas con éste. Para el resto de casos en ruidos Rosa y Babble, los mejores resultados fueron obtenidos con los sistemas híbridos HW-DLSTM-2 y HW-DLSTM-3.

**Tabla 10.2:** Resultados de la medida WSS. Los valores más bajos indican mejor resultado. * es el mejor resultado. En negrita las medidas que no difieren significativamente del mejor.

| | Ruido Rosa | | | | |
|---|---|---|---|---|---|
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | 49.7 | 42.6 | 37.1 | 32.9 | 30.0 |
| Wiener | 68.8 | 57.8 | 47.8 | 40.2 | 34.5 |
| DLSTM-1 | 72.0 | 54.9 | 27.8 | **21.0** | **17.0** |
| DLSTM-2 | 75.1 | 57.0 | 29.6 | **21.4** | **16.7** |
| DLSTM-3 | 74.5 | 64.0 | 47.3 | 68.9 | 27.3 |
| HW-DLSTM-1 | 38.5 | 30.6 | 26.6 | 23.7 | 20.0 |
| HW-DLSTM-2 | 28.9* | 22.1* | 19.2* | 16.8* | 15.1* |
| HW-DLSTM-3 | 65.8 | 92.6 | 39.2 | 28.4 | 27.2 |
| | Ruido Rosa | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | 63.6 | 55.1 | 47.0 | 39.4 | 33.3 |
| Wiener | 79.2 | 67.5 | 33.7 | 41.9 | 34.1 |
| DLSTM-1 | 70.9 | **41.3** | **28.3** | **19.7** | **15.2** |
| DLSTM-2 | 73.6 | 45.9 | **24.8** | **19.9** | **14.5** |
| DLSTM-3 | 59.5* | 71.9 | 77.3 | 60.6 | 49.2 |
| HW-DLSTM-1 | **64.8** | 45.0 | **29.5** | **22.8** | **18.1** |
| HW-DLSTM-2 | **60.9** | 35.4* | 21.5* | 17.4* | 13.9* |
| HW-DLSTM-3 | 77.2 | 59.8 | 77.4 | 65.1 | 56.4 |
| | Ruido Babble | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | 71.6 | 57.8 | 43.4 | 31.7 | 21.0 |
| Wiener | 79.6 | **61.5** | 44.5 | 31.4 | 22.3 |
| DLSTM-1 | 72.5 | **61.1** | 48.6 | 30.1 | **17.5** |
| DLSTM-2 | 73.5 | **60.4** | 49.6 | 30.3 | **17.1** |
| DLSTM-3 | 72.1 | 79.3 | 102.2 | 48.1 | 41.0 |
| HW-DLSTM-1 | 82.8 | **59.4** | **35.7** | **24.7** | **17.3** |
| HW-DLSTM-2 | 77.8 | 53.6* | 30.3* | 21.0* | 15.3* |
| HW-DLSTM-3 | 60.2* | 88.0 | 50.2 | 74.6 | 42.3 |

Se puede observar cómo los sistemas híbridos presentaron mejores resultados del parámetro WSS en catorce de los quince casos analizados. En el restante, la diferencia no es significativa con el mejor.

Los resultados para la medida PESQ se muestran en la Tabla 10.3. Para esta medida objetiva, nuevamente el híbrido HW-DLSTM-2 obtuvo el mejor resultado para todos los niveles de Ruido Blanco. En la mayoría de los casos en este tipo de ruido las diferencias con el resto de sistemas son significativas, con excepción de SNR 10, donde el sistema no híbrido DLSTM-2 obtuvo un valor que no tiene diferencia significativa.

Para el caso de Ruido Rosa, el filtro Wiener obtuvo los mejores resultados para el nivel más alto de ruido, pero para el resto, el híbrido HW-DLSTM-2 presentó el mejor, mientras que en los sistemas no híbridos, DLSTM-2 obtuvo resultados que no difieren significativamente del mejor, con excepción de SNR -10.

En el Ruido Babble, el filtro Wiener obtuvo el mejor resultado en para uno de los niveles (SNR -5), aunque DLSTM-2 y el híbrido HW-DLSTM-2 no difieren de este resultado significativamente. Los mejores resultados para este tipo de ruido se obtuvieron con el sistema híbrido HW-DLSTM-2, especialmente en los niveles SNR 0, SNR 5 y SNR 10.

En resumen, para el caso de la medida PESQ, de los quince casos analizados, los sistemas híbridos obtuvieron el mejor resultado en doce de ellos. En los tres restantes, su diferencia no es significativa con el mejor resultado.

Para la medida $\text{SegSNR}_\text{f}$, los resultados se muestran en la Tabla 10.4. En el caso de Ruido Blanco, el sistema híbrido HW-DLSTM-2 obtuvo resultados que son significativamente mejores que el resto en el caso de los niveles más altos de ruido (SNR-10, SNR -5, SNR 0), mientras que para SNR -5 su diferencia con el mejor, DLSTM-2 no es significativa. Para el nivel de ruido más bajo, DLSTM-2 obtuvo un resultado significativamente mejor que el resto.

En el Ruido Rosa, el sistema híbrido HW-DLSTM-2 presenta dos mejores resultados. Destaca, para este medida en Ruido Rosa, el sistema DLSTM-2 en los niveles bajos de ruido, y DLSTM-3 en el más alto.

En Ruido Babble y el parámetro $\text{SegSNR}_\text{f}$ se dan los resultados menos favorables para los sistemas híbridos, pues solamente tiene un mejor resultado y otro que no difiere significativamente del mejor. Destaca aquí el filtro Wiener, con los mejores resultados para los tres niveles de ruido más bajos.

### 10.4.2 Análisis de significancia estadística

En esta sección se presenta un análisis estadístico para determinar cuáles de los resultados obtenidos con los sistemas híbridos o los considerados para comparación, obtienen mejoras significativas con respecto a la señal ruidosa. Esto debido a que un valor presentado en la tablas

**Tabla 10.3:** Resultados de la medida PESQ. Los valores más altos indican mejor resultado. * es el mejor resultado. En negrita las medidas que no difieren significativamente del mejor.

| | Ruido Blanco | | | | |
|---|---|---|---|---|---|
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | 1.0 | 1.1 | 1.3 | 1.6 | 1.9 |
| Wiener | 1.0 | 1.3 | 1.7 | 2.1 | 2.4 |
| DLSTM-1 | 0.8 | 1.1 | 1.8 | 2.4 | 2.7 |
| DLSTM-2 | 0.8 | 1.3 | 2.0 | 2.6 | **3.0** |
| DLSTM-3 | 0.4 | 0.6 | 0.9 | 0.6 | 2.0 |
| HW-DLSTM-1 | 1.5 | 2.0 | 2.2 | 2.4 | 2.7 |
| HW-DLSTM-2 | 1.8* | 2.3* | 2.6* | 2.9* | 3.1* |
| HW-DLSTM-3 | 0.5 | 0.3 | 1.1 | 1.7 | 1.8 |
| | Ruido Pink | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | 0.8 | 1.1 | 1.3 | 1.7 | 2.0 |
| Wiener | 1.0* | 1.3 | 1.9 | 2.1 | 2.5 |
| DLSTM-1 | 0.7 | 1.3 | 1.9 | 2.5 | 2.9 |
| DLSTM-2 | 0.8 | **1.5** | **2.3** | **2.7** | 3.3* |
| DLSTM-3 | 0.7 | 0.6 | 0.6 | 0.7 | 1.0 |
| HW-DLSTM-1 | 0.7 | 1.3 | 2.1 | 2.5 | 2.9 |
| HW-DLSTM-2 | **0.9** | 1.6* | 2.4* | 2.9* | 3.3* |
| HW-DLSTM-3 | 0.3 | 0.7 | 0.6 | 0.6 | 0.8 |
| | Ruido Babble | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | 0.6 | 0.9 | 1.3 | 1.8 | 2.3 |
| Wiener | **0.6** | 1.1* | **1.6** | 2.1 | 2.5 |
| DLSTM-1 | **0.6** | 0.8 | 1.1 | 1.7 | 2.8 |
| DLSTM-2 | **0.7*** | **1.0** | 1.3 | 1.9 | **3.1** |
| DLSTM-3 | 0.4 | 0.5 | 0.4 | 0.9 | 1.1 |
| HW-DLSTM-1 | 0.1 | 0.7 | **1.5** | **2.4** | **3.0** |
| HW-DLSTM-2 | **0.6** | **1.0** | 1.7* | 2.6* | 3.2* |
| HW-DLSTM-3 | **0.6** | 0.4 | 0.9 | 0.6 | 1.1 |

anteriores puede ser el mejor entre los sistemas comparados, pero aún así no mejora de forma suficiente la señal de habla ruidosa. En las tablas 10.5 a 10.7 se muestran los resultados.

En la Tabla 10.5 se muestra que los sistemas híbridos HW-DLSTM-1 y HW-DLSTM-2 mejoraron significativamente el valor de la medida WSS en todos los casos de Ruido Blanco. Se pueden considerar, para este caso, como los sistemas de mejores resultados. Para el caso de Ruido Rosa, la cantidad de mejoras obtenidas con los sistemas híbridos HW-DLSTM-1 y HW-LSTM-2 son comparables a las obtenidas por DLSTM-1 y DLSTM-2. En Ruido Babble, solamente los

**Tabla 10.4:** Resultados de la medida $\text{SegSNR}_{\text{f}}$. Los valores más altos indican mejor resultado. * es el mejor resultado. En negrita las medidas que no difieren significativamente del mejor.

| | Ruido Blanco | | | | |
|---|---|---|---|---|---|
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | -3.4 | -0.64 | 2.5 | 5.2 | 6.6 |
| Wiener | 0.6 | 4.1 | 7.4 | 8.8 | 9.4 |
| DLSTM-1 | -4.9 | -3.3 | 2.3 | 5.4 | 6.7 |
| DLSTM-2 | 0.0 | 0.8 | 5.3 | 10.5* | 12.3* |
| DLSTM-3 | 0.4 | 1.1 | 2.5 | 1.2 | 6.1 |
| HW-DLSTM-1 | 2.7 | 3.8 | 3.9 | 3.7 | 4.0 |
| HW-DLSTM-2 | 4.6* | 8.1* | 8.7* | **9.6** | 10.9 |
| HW-DLSTM-3 | 0.6 | -0.2 | 3.0 | 5.3 | 6.1 |
| | Ruido Rosa | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | -3.8 | -2.0 | 0.8 | 4.0 | 6.0 |
| Wiener | -1.1 | 1.8 | 5.9 | 7.8 | 9.2 |
| DLSTM-1 | -3.1 | 0.6 | 3.9 | 5.4 | 7.4 |
| DLSTM-2 | -0.2 | 1.4 | **7.4** | 10.6* | 13.1* |
| DLSTM-3 | 0.9* | 0.6 | 0.7 | 1.6 | 3.3 |
| HW-DLSTM-1 | 0.4 | 1.4 | 3.1 | 3.7 | 4.1 |
| HW-DLSTM-2 | 0.3 | 3.1* | 7.8* | 9.0 | 10.8 |
| HW-DLSTM-3 | 0.1 | 0.5 | 0.1 | 0.7 | 1.8 |
| | Ruido Babble | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Ninguno | -0.3 | 0.2 | 2.3 | 6.1 | 12.7 |
| Wiener | -0.3 | 0.9 | 4.5* | 9.2* | 13.4* |
| DLSTMA-1 | -0.1 | **1.0** | 1.5 | 3.2 | 7.7 |
| DLSTMA-2 | 0.4* | 1.2* | 1.9 | 5.1 | 11.1 |
| DLSTMA-3 | 0.4* | 0.2 | -0.5 | 2.4 | 3.3 |
| HW-DLSTMA-1 | **0.2** | 0.5 | 2.1 | 4.4 | 7.8 |
| HW-DLSTMA-2 | -0.6 | 0.8 | **4.2** | 8.1 | 10.7 |
| HW-DLSTMA-3 | 0.4* | -0.1 | 1.6 | 0.6 | 3.6 |

sistemas híbridos presentaron al menos una mejora significativa en todos los niveles de ruido.

Para la medida PESQ, en la Tabla 10.6, los resultados para el Ruido Blanco son semejantes al caso anterior, en cuanto los sistemas híbridos HW-DLSTM-1 y HW-DLSTM-2 obtuvieron mejoras significativas en todos los niveles. Los sistemas en los cuales se aplicó solamente el filtro Wiener y los DLSTM obtuvieron mejoras solamente en los niveles altos de ruido.

En el caso de Ruido Rosa, en cuanto a mejoras de la señal ruidosa todos los sistemas obtuvieron igual cantidad de mejoras. Se destaca también en este ruido el hecho de que DLSTM-3 y

**Tabla 10.5:** Resultados de la medida WSS. ✓ indica una mejora estadísticamente significativa.

| | Ruido Blanco | | | | |
|---|---|---|---|---|---|
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | | | | | |
| DLSTM-1 | | | ✓ | ✓ | ✓ |
| DLSTM-2 | | | ✓ | ✓ | ✓ |
| DLSTM-3 | | | | | |
| HW-DLSTM-1 | ✓ | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-3 | | | | | ✓ |
| | Ruido Rosa | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | | | ✓ | | |
| DLSTM-1 | | ✓ | ✓ | ✓ | ✓ |
| DLSTM-2 | | ✓ | ✓ | ✓ | ✓ |
| DLSTM-3 | | | ✓ | | |
| HW-DLSTM-1 | | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-2 | | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-3 | | | | | |
| | Ruido Babble | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | | ✓ | | | |
| DLSTM-1 | | ✓ | | | |
| DLSTM-2 | | ✓ | | | ✓ |
| DLSTM-3 | | | | | ✓ |
| HW-DLSTM-1 | | ✓ | ✓ | ✓ | |
| HW-DLSTM-2 | | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-3 | ✓ | | | | ✓ |

HW-DLSTM-3 no obtuvieron ninguna mejora.

Para el ruido Babble, la mayor cantidad de mejoras las obtuvieron el filtro Wiener y los híbridos HW-DLSTM-1 y HW-DLSTM-2.

Finalmente, en la Tabla 10.7 se muestran los resultados de las mejoras significativas para la medida $\text{SegSNR}_{\text{f}}$. La mayor cantidad de mejoras fueron obtenidas con el filtro Wiener y el híbrido HW-DLSTM-2. La excepción para el caso de los sistemas híbridos se dio en los niveles más alto y más bajo de ruido, en los cuales las mejoras no son significativas.

**Tabla 10.6:** Resultados de la medida PESQ. ✓ indica una mejora estadísticamente significativa.

| | Ruido Blanco | | | | |
|---|---|---|---|---|---|
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | | | ✓ | ✓ | ✓ |
| DLSTM-1 | | | ✓ | ✓ | ✓ |
| DLSTM-2 | | | ✓ | ✓ | ✓ |
| DLSTM-3 | | | | | |
| HW-DLSTM-1 | ✓ | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-3 | | | | | |
| | Ruido Rosa | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | | ✓ | ✓ | ✓ | ✓ |
| DLSTM-1 | | ✓ | ✓ | ✓ | ✓ |
| DLSTM-2 | | ✓ | ✓ | ✓ | ✓ |
| DLSTM-3 | | | | | |
| HW-DLSTM-1 | | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-2 | | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-3 | | | | | |
| | Ruido Babble | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | | ✓ | ✓ | ✓ | |
| DLSTMA-1 | | | | | ✓ |
| DLSTMA-2 | | | | | ✓ |
| DLSTMA-3 | | | | | |
| HW-DLSTMA-1 | | | ✓ | ✓ | ✓ |
| HW-DLSTMA-2 | | | ✓ | ✓ | ✓ |
| HW-DLSTMA-3 | | | | | |

## 10.5 Resumen de contribuciones

En este capítulo se realizó una propuesta de aplicación de las colecciones de redes LSTM aplicadas para mejorar las señales de voz en conjunto con los filtros Wiener, para mejorar señales de habla que contienen ruido. Se realizó una comparación con la propuesta no híbrida en redes LSTM y el filtro Wiener aplicado en una única etapa.

Fueron tres tipos de sistemas híbridos presentados, coincidentes con las tres propuestas de post-filtros realizadas para mejorar las señales de habla sintetizada. La evaluación se realizó

**Tabla 10.7:** Resultados de la medida $\text{SegSNR}_{\text{f}}$. ✓ indica una mejora estadísticamente significativa.

| | Ruido blanco | | | | |
|---|---|---|---|---|---|
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | ✓ | ✓ | ✓ | ✓ | ✓ |
| DLSTMA-1 | | | | | |
| DLSTMA-2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| DLSTMA-3 | ✓ | ✓ | | | |
| HW-DLSTMA-1 | ✓ | ✓ | ✓ | | |
| HW-DLSTMA-2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTMA-3 | ✓ | | | | |
| | Ruido Rosa | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | | ✓ | ✓ | ✓ | ✓ |
| DLSTM-1 | | ✓ | ✓ | | ✓ |
| DLSTM-2 | | ✓ | ✓ | ✓ | ✓ |
| DLSTM-3 | ✓ | ✓ | | | |
| HW-DLSTM-1 | ✓ | ✓ | ✓ | | |
| HW-DLSTM-2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| HW-DLSTM-3 | ✓ | ✓ | | | |
| | Ruido Babble | | | | |
| Sistema | SNR -10 | SNR -5 | SNR 0 | SNR 5 | SNR 10 |
| Wiener | | ✓ | ✓ | ✓ | |
| DLSTM-1 | | ✓ | | | |
| DLSTM-2 | ✓ | ✓ | | | |
| DLSTM-3 | ✓ | | | | |
| HW-DLSTM-1 | ✓ | | | | |
| HW-DLSTM-2 | | ✓ | ✓ | ✓ | |
| HW-DLSTM-3 | ✓ | | | | |

utilizando tres medidas objetivas y tres tipos de ruido con cinco niveles de intensidad.

Los resultados muestran cómo la propuesta de un sistema híbrido que combina dos tipos de filtrado de diferente naturaleza obtienen mejores resultados en la gran mayoría de los casos que aquellos sistemas de una sola etapa. En particular, se destaca las medidas WSS y PESQ, para los cuales la propuesta híbrida tiene resultados que son significativamente mejores que el resto de sistemas considerados para comparación.

# 11

# Conclusiones y perspectivas de la tesis

Con el propósito de mejorar los resultados que se producen hasta el momento con la síntesis estadística paramétrica de voz basada en HMM, en esta tesis se presentaron tres propuestas basadas en la utilización de algoritmos de aprendizaje profundo. Para determinar el modelo que se incorporó en la propuesta, se analizó la evolución de las redes neuronales presentadas en las referencias, las cuales iniciaron a publicarse en el año 2013. Es claro que estas primeras referencias se centraban en la mejora de las características espectrales, dejando de lado otros parámetros como el $f_0$, por las limitaciones que tienen las redes neuronales para tratar con un parámetro que oscila entre intervalos donde su valor es cero, y otros intervalos con valores positivos (segmentos no sonoros y sonoros).

La decisión de utilizar los LSTM provino del estudio realizado en áreas afines a la síntesis de voz, tales como el reconocimiento de habla y la eliminación de ruido en señales de voz, además de la generación de letra manuscrita. En éstos, los LSTM probaron ser el modelo con mejores resultados en diversas tareas que comparten con la síntesis de voz su naturaleza secuencial y la dependencia de valores previos.

Por otra parte, una dificultad importante para la aplicación de post-filtros se encontraba en la no correspondencia entre habla natural y sintetizada, debida al proceso de promediado de parámetros de duración en el entrenamiento de HMM. Para solventar este problema, se creó una variante del sistema HTS (*H*TS-Parallel), en la cual se utiliza la información del proceso de segmentación para producir frases sintetizadas alineadas con las naturales, con las cuales se puede trabajar en una correspondencia directa ventana a ventana de información.

La principal arquitectura implementada en las aplicaciones de LSTM relacionadas con habla es el *autoencoder*. Por esta razón se consideró la primera opción para realizar las mejoras en los parámetros generados en el habla sintetizada. Sin embargo, su aplicación en la mejora de todos los parámetros generados por los HMM resultó insuficiente, por lo cual se realizaron tres propuestas independientes que consideran combinaciones de dos arquitecturas, cuyas conclusiones se muestran en la siguiente sección.

## 11.1 Propuestas de mejora del habla sintetizada con HMM

La primer propuesta de post-filtros para mejorar el habla sintetizada con HMM realizó una comparación entre cuatro sistemas, abarcando desde un solo *autoencoder* para mejorar todos los parámetros, hasta un *autoencoder* y dos memorias auto-asociativas para los parámetros MFCC, de energía y $f_0$. A diferencia de una memoria auto-asociativa regular, la cual se entrena para aproximar la función identidad, la propuesta contempla una variante de esta arquitectura con treinta y nueve entradas y salidas con este mismo objetivo, mientras que la entrada y salida número cuarenta se incorporan para reconstruir el parámetro correspondiente a partir de la información de las restantes entradas.

Esta novedad en la arquitectura mostró buenos resultados en comparación con los sistemas de una sola red neuronal hasta el momento presentados en la literatura. La ventaja principal de este sistema es que puede mejorar parámetros cuya escala difiere de la de los coeficientes MFCC, potenciando de esta manera la aplicación de LSTM en post-filtros.

A pesar de los resultados favorables en comparación con la voz HMM utilizando varias medidas objetivas, esta primer propuesta cuenta con dos limitaciones importantes. La primera de éstas es que requiere de una sola red LSTM para modelar todos los elementos que se distorsionan o degradan en la voz artificial, en comparación con la voz natural. Esto representa un alto grado de complejidad en la relación entre ambas. La segunda limitación proviene del hecho que las distorsiones pueden diferir de acuerdo con los segmentos del habla, ya que estos segmentos se producen a partir de cantidades distintas de información, por lo cual las redes deben modelar también una función dependiente del sonido emitido, del cual no tienen información.

Para solventar el primer problema, se propuso un sistema de filtros en dos etapas. En la primera etapa se aplicó un filtro Wiener para reducir el componente de ruido observado en la señal de habla sintetizada, y de esta manera reducir la complejidad en el mapeo requerido por el conjunto de redes LSTM aplicados a la señal después del filtro. Los resultados de estas dos etapas fueron comparados con los del sistema que aplica únicamente el conjunto de redes LSTM, y mostraron mejoría en gran parte de las medidas objetivas aplicadas. Estos resultados son un indicio del beneficio obtenido al aplicar combinaciones de filtros, tanto clásicos como filtros basados en aprendizaje profundo.

En cuanto al segundo problema, sobre las desventajas producidas al aplicar un mismo conjunto de arquitecturas LSTM para todos los segmentos del habla, se propuso un sistema discriminativo para los segmentos sonoros y no sonoros. En éstos se procedió con un entrenamiento independiente del conjunto de redes neuronales para los distintos segmentos. Los resultados mostraron mejoras en las medidas objetivas tanto en relación con el habla HTS como con los

post-filtros propuestos originalmente. Las mejoras obtenidas en esta última propuesta fueron verificadas con una evaluación subjetiva, en la cual claramente las voces procesadas con este sistema discriminativo superaron en naturalidad a las producidas por HMM y por los post-filtros no discriminativos.

Uno de los resultados más importantes de esta propuesta es la mejora obtenida en el parámetro $f_0$, lo cual no había sido logrado en los sistemas anteriores. En esta ocasión, el logro se debe al mapeo directo entre coeficientes estrictamente positivos, excluyendo el mapeo de valores cero y las consecuentes transiciones a valores positivos.

Entre las tres propuestas, la que mejores resultados generales presentó fue la discriminativa. Esto abre nuevas perspectivas para el desarrollo de post-filtros, lo cual se discutirá en la Sección 11.3.

## 11.2 Otras aplicaciones

La producción de voces con HMM y el sistema de adaptación para generar nuevas voces a partir de modelos promedio, ha permitido desarrollar, entre otras aplicaciones, sistemas capaces de producir cambios de lenguaje y acento en otros idiomas. Por esta razón, en esta tesis se ha mostrado cómo el procedimiento de adaptación puede producir un cambio de acento entre castellano europeo y mexicano, a partir de funciones lineales que cambian parcialmente las distribuciones de probabilidad de una voz en un acento hacia el otro.

Los resultados, basados tanto en evaluaciones objetivas como subjetivas muestran que la voz en el primer acento es modificada en características prosódicas y de percepción que la hacen asemejarse a la voz en el segundo acento. Estos resultados han sido más contundentes en el cambio de acento del español mexicano al europeo, posiblemente por la calidad de datos disponibles en cada uno para el proceso de adaptación, en cuanto a su poca variabilidad y homogeneidad de características.

Por otra parte, los buenos resultados obtenidos con el conjunto de post-filtros aplicados en la voz artificial generada con HMM, tiene una extensión inmediata hacia la mejora en voces degradadas con ruido. Esta área, en la cual existen gran cantidad de algoritmos con distintas formas de abordar el problema, ofrece posibilidades de probar las propuestas a partir de tipos y niveles específicos de ruido.

Con éstos se mostró la competitividad de la propuesta de combinación de arquitecturas de redes LSTM para la misma parametrización utilizada en síntesis de voz, en la eliminación del

ruido. Para algunos de los tipos y niveles de ruido considerados, los conjuntos de redes LSTM mostraron superar otros algoritmos clásicos.

Debido al éxito de los post-filtros en dos etapas con filtros Wiener y las redes LSTM, ésta propuesta fue trasladada al problema de reducción de ruido. Los resultados superaron a los de los algoritmos individuales, comprobando la utilidad de combinar algoritmos clásicos y basados en algoritmos de aprendizaje profundo.

## 11.3 Líneas de investigación

Una primer línea de investigación que se puede señalar a partir del presente trabajo es la aplicación de post-filtros a nuevas parametrizaciones del habla, en las cuales exista una menor pérdida de información en el proceso de reconstrucción. Por ejemplo, contemplando coeficientes de información aperiódica que posiblemente requieran nuevos conjuntos de arquitecturas LSTM. De esta manera, el alcance puede extenderse y la calidad de los resultados incrementarse aún más.

Los sistemas de post-filtros propuestos pueden aplicarse a áreas donde la síntesis de voz basada en HMM no ha producido buenos resultados, tales como en voces con emociones. La limitación principal se debe a que en el proceso de entrenamiento de HMM la información de la expresión es promediada, por lo que el resultado pierde las características que lo hacen identificable con una emoción. Los post-filtros pueden devolver al habla sintetizada la variabilidad en parámetros que se refleje en la prosodia propia del habla con emociones.

En cuanto a la combinación de algoritmos para el proceso de post-filtros, en la presente tesis se probó una combinación de dos tipos de filtros. Se abre entonces la posibilidad de combinar muchos más tipos de algoritmos en el proceso, incluyendo un sistema inverso al aplicado en la presente tesis: primero el conjunto de post-filtros LSTM y luego otros filtros clásicos o basados en otro tipo de procedimientos.

Los post-filtros discriminativos propuestos para mejorar voces artificiales pueden considerarse solamente un paso en la jerarquía de clasificación de sonidos del habla. En el futuro se pueden desarrollar sistemas discriminativos para conjuntos más específicos, como consonantes plosivas, vocales, consonantes líquidas, entre otras, y aplicar los conjuntos de post-filtros de forma independiente en cada grupo.

Para la mejora de señales que han sido degradadas con ruidos específicos, una extensión natural de la propuesta es la utilización de otras parametrizaciones del habla, y el establecimiento de nuevos conjuntos de arquitecturas LSTM para determinados subconjuntos de parámetros.

Para la aplicación de cambio de acento, posterior al proceso de modificación a partir del mapeo entre las distribuciones de probabilidad de los HMM, se pueden utilizar post-filtros para mejorar los resultados y clarificar el acento. Esto requiere de una extensa experimentación para determinar la conveniencia de aplicar post-filtros entrenados para un acento específico o directamente en el proceso de cambio de acento.

Con excepción de la aplicación de cambio de acento, los bases de datos utilizadas en esta tesis han sido en idioma inglés. Dado que los sistemas son independientes del idioma (aún más claramente en el caso de reducción de ruido), lo propuesto puede probarse en otros idiomas, iniciando por el castellano o lenguas autóctonas americanas.

Finalmente, existe un área de oportunidad ligada a generar los parámetros del habla directamente a partir del texto, aplicando conjuntos de redes LSTM de varias arquitecturas que tengan como entrada especificaciones lingüísticas. Esto ha sido explorado en diversas referencias, pero aún no ha sido presentado un sistema que abarque distintos tipos de redes para la generación individual de parámetros.

# REFERENCIAS


[1] Koichi Shinoda and Takao Watanabe. Mdl-based context-dependent subword modeling for speech recognition. *Acoustical Science and Technology*, 21(2):79–86, 2001.

[2] Faizan Shaikh. Introduction to gradient descent algorithm (along with variants) in machine learning. http://https://www.analyticsvidhya.com/blog/2017/03/introduction-to-gradient-descent-algorithm-along-its-variants/. Accessed: 2017-04-30.

[3] Alan W Black. Unit selection and emotional speech. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'03*, pages 1649–1652, 2003.

[4] Junichi Yamagishi. An introduction to hmm-based speech synthesis. *Technical Report*, 2006.

[5] Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura. Speech synthesis based on hidden markov models. *Proceedings of the IEEE*, 101(5):1234–1252, 2013.

[6] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W Black, and Keiichi Tokuda. The hmm-based speech synthesis system (hts) version 2.0. In *SSW*, pages 294–299, 2007.

[7] Keiichi Tokuda, Takashi Masuko, Noboru Miyazaki, and Takao Kobayashi. Hidden markov models based on multi-space probability distribution for pitch pattern modeling. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 1, pages 229–232. IEEE, 1999.

[8] Steve J Young, Julian J Odell, and Philip C Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology*, pages 307–312. Association for Computational Linguistics, 1994.

[9] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In *Sixth European Conference on Speech Communication and Technology*, pages 2347–2350, 1999.

[10] Satoshi Imai. Cepstral analysis synthesis on the mel frequency scale. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'83.*, volume 8, pages 93–96. IEEE, 1983.

[11] Keiichi Tokuda, Takao Kobayashi, Takashi Masuko, and Satoshi Imai. Mel-generalized cepstral analysis-a unified approach to speech spectral estimation. In *ICSLP*, volume 94, pages 18–22, 1994.

[12] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Mixed excitation for hmm-based speech synthesis. In *Seventh European Conference on Speech Communication and Technology*, 2001.

[13] Krichi Mohamed Khalil and Cherif Adnan. Implementation of speech synthesis based on hmm using padas database. In *Systems, Signals & Devices (SSD), 2015 12th International Multi-Conference on*, pages 1–6. IEEE, 2015.

[14] Michael Pucher, Dietmar Schabus, Junichi Yamagishi, Friedrich Neubarth, and Volker Strom. Modeling and interpolation of austrian german and viennese dialect in hmm-based speech synthesis. *Speech Communication*, 52(2):164–179, 2010.

[15] Daniel Erro, Iñaki Sainz, Iker Luengo, Igor Odriozola, Jon Sánchez, Ibon Saratxaga, Eva Navas, and Inma Hernáez. Hmm-based speech synthesis in basque language using hts. *Proc. FALA*, pages 67–70, 2010.

[16] Adriana Stan, Junichi Yamagishi, Simon King, and Matthew Aylett. The romanian speech synthesis (rss) corpus: Building a high quality hmm-based speech synthesis system using a high sampling rate. *Speech Communication*, 53(3):442–450, 2011.

[17] Tomasz Kuczmarski. Hmm-based speech synthesis applied to polish. *Speech and Language Technology*, 12:13, 2010.

[18] Zdeněk Hanzlíček. Czech hmm-based speech synthesis. In *International Conference on Text, Speech and Dialogue*, pages 291–298. Springer, 2010.

[19] Ya Li, Shifeng Pan, and Jianhua Tao. Hmm-based speech synthesis with a flexible mandarin stress adaptation model. In *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, pages 625–628. IEEE, 2010.

[20] Son Thanh Phan, Thang Tat Vu, Cuong Tu Duong, and Mai Chi Luong. A study in vietnamese statistical parametric speech synthesis based on hmm. *International Journal*, 2(1):1–6, 2013.

[21] Ramani Boothalingam, V Sherlin Solomi, Anushiya Rachel Gladston, S Lilly Christina, P Vijayalakshmi, Nagarajan Thangavelu, and Hema A Murthy. Development and evaluation of unit selection and hmm-based speech synthesis systems for tamil. In *Communications (NCC), 2013 National Conference on*, pages 1–5. IEEE, 2013.

[22] Krichi Mohamed Khalil and Cherif Adnan. Implementation of speech synthesis based on hmm using padas database. In *Systems, Signals & Devices (SSD), 2015 12th International Multi-Conference on*, pages 1–6. IEEE, 2015.

[23] Kazuhiro Nakamura, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Hmm-based singing voice synthesis and its application to japanese and english. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 265–269. IEEE, 2014.

[24] Sophie Roekhaut, Sandrine Brognaux, Richard Beaufort, and Thierry Dutoit. elite-hts: A nlp tool for french hmm-based speech synthesis. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'14*, pages 2136–2137, 2014.

[25] Keijiro Saino, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda. An hmm-based singing voice synthesis system. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'06*, pages 2274–2277, 2006.

[26] Takeshi Saitou, Masataka Goto, Masashi Unoki, and Masato Akagi. Speech-to-singing synthesis: Converting speaking voices to singing voices by controlling acoustic features unique to singing voices. In *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*, pages 215–218. IEEE, 2007.

[27] Simon King. An introduction to statistical parametric speech synthesis. *Sadhana*, 36(5):837–852, 2011.

[28] Heiga Ze, Andrew Senior, and Mike Schuster. Statistical parametric speech synthesis using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7962–7966. IEEE, 2013.

[29] S Prahallad Kishore and Alan W Black. Unit size in unit selection speech synthesis. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[31] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[32] Heiga Zen and Haşim Sak. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4470–4474. IEEE, 2015.

[33] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013.

[34] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*, pages 753–753, 2005.

[35] Timit database. `https://catalog.ldc.upenn.edu/ldc93s1`. Accessed: 2017-04-30.

[36] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K Soong. Tts synthesis with bidirectional lstm based recurrent neural networks. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'14*, pages 1964–1968, 2014.

[37] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[38] Ling-Hui Chen, Tuomo Raitio, Cassia Valentini-Botinhao, Junichi Yamagishi, and Zhen-Hua Ling. Dnn-based stochastic postfilter for hmm-based speech synthesis. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'14*, pages 1954–1958, 2014.

[39] Shinnosuke Takamichi, Tomoki Toda, Alan W Black, and Satoshi Nakamura. Modified post-filter to recover modulation spectrum for hmm-based speech synthesis. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 547–551. IEEE, 2014.

[40] Prasanna Kumar Muthukumar and Alan W Black. Recurrent neural network postfilters for statistical parametric speech synthesis. *arXiv preprint arXiv:1601.07215*, 2016.

[41] Shinnosuke Takamichi, Tomoki Toda, Alan W Black, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura. Postfilters to modify the modulation spectrum for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):755–767, 2016.

[42] Ling-Hui Chen, Tuomo Raitio, Cassia Valentini-Botinhao, Zhen-Hua Ling, and Junichi Yamagishi. A deep generative architecture for postfiltering in statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(11):2003–2014, 2015.

[43] Alan W Black. Clustergen: A statistical parametric synthesizer using trajectory modeling. In *Ninth International Conference on Spoken Language Processing*, 2006.

[44] Toru Nakashika, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. Voice conversion in high-order eigen space using deep belief nets. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'14*, pages 369–372, 2013.

[45] Prasanna Kumar Muthukumar and Alan W Black. Recurrent neural network postfilters for statistical parametric speech synthesis. *arXiv preprint arXiv:1601.07215*, 2016.

[46] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Duration modeling for hmm-based speech synthesis. In *ICSLP*, volume 98, pages 29–32, 1998.

[47] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[48] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[49] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. An experimental study on speech enhancement based on deep neural networks. *IEEE Signal processing letters*, 21(1):65–68, 2014.

[50] Xueheng Qiu, Le Zhang, Ye Ren, Ponnuthurai N Suganthan, and Gehan Amaratunga. Ensemble deep learning for regression and time series forecasting. In *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on*, pages 1–6. IEEE, 2014.

[51] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'13*, pages 436–440, 2013.

[52] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2016.

[53] Daniel Erro, Iñaki Sainz, Eva Navas, and Inma Hernáez. Improved hnm-based vocoder for statistical synthesizers. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'11*, pages 1809–1812, 2011.

[54] John Kominek, Alan W Black, and Ver Ver. Cmu arctic databases for speech synthesis. 2003.

[55] International Telecommunications Unit. Recomendación itu-t p.862 (pesq). `https://www.itu.int/rec/T-REC-P.862-200102-I/es`. Accessed: 2017-04-30.

[56] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 2, pages 749–752. IEEE, 2001.

[57] Yi Hu and Philipos C Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 16(1):229–238, 2008.

[58] Dennis Klatt. Prediction of perceived phonetic distance from critical-band spectra: A first step. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.*, volume 7, pages 1278–1281. IEEE, 1982.

[59] Speechmatics. `http://speechmatics.com`. Accessed: 2017-04-30.

[60] Meng Zhang, Jianhua Tao, Huibin Jia, and Xia Wang. Improving hmm based speech synthesis by reducing over-smoothing problems. In *Chinese Spoken Language Processing, 2008. ISCSLP'08. 6th International Symposium on*, pages 1–4. IEEE, 2008.

[61] Peter J. Sherman. Discrete wiener filter methods in deterministic signal estimation with applications to evoked responses. *University of Wisconsin–Madison*, 1985.

[62] Pawan Patidar, Manoj Gupta, Sumit Srivastava, and Ashok Kumar Nagawat. Image de-noising by various filters for different noise. *International journal of computer applications*, 9(4), 2010.

[63] María Antonia Martí Antonín and Joaquim Llisterri Boix. *Tecnologías del texto y del habla*, volume 72. Edicions Universitat Barcelona, 2004.

[64] Pascal Scalart et al. Speech enhancement based on a priori signal to noise estimation. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 629–632. IEEE, 1996.

[65] Jingdong Chen, Jacob Benesty, Yiteng Huang, and Simon Doclo. New insights into the noise reduction wiener filter. *IEEE Transactions on audio, speech, and language processing*, 14(4):1218–1234, 2006.

[66] Jacob Benesty, Jingdong Chen, Yiteng Arden Huang, and Simon Doclo. Study of the wiener filter for noise reduction. In *Speech Enhancement*, pages 9–41. Springer, 2005.

[67] Joerg Meyer and Klaus Uwe Simmer. Multi-channel speech enhancement in a car environment using wiener filtering and spectral subtraction. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1167–1170. IEEE, 1997.

[68] Chung-Chien Hsu, Kah-Meng Cheong, Jen-Tzung Chien, and Tai-Shih Chi. Modulation wiener filter for improving speech intelligibility. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 370–374. IEEE, 2015.

[69] Hervé Abdi and Lynne J Williams. Tukey's honestly significant difference (hsd) test. *Encyclopedia of Research Design. Thousand Oaks, CA: Sage*, pages 1–5, 2010.

[70] Mark JF Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2):75–98, 1998.

[71] Alex Acero, Li Deng, Trausti Kristjansson, and Jerry Zhang. Hmm adaptation using vector taylor series for noisy speech recognition. In *Sixth International Conference on Spoken Language Processing*, 2000.

[72] Petr Motlicek, Philip N Garner, Namhoon Kim, and Jeongmi Cho. Accent adaptation using subspace gaussian mixture models. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7170–7174. IEEE, 2013.

[73] Masatsune Tamura, Takashi Masuko, Keiichi Tokuda, and Takao Kobayashi. Adaptation of pitch and spectrum for hmm-based speech synthesis using mllr. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 2, pages 805–808. IEEE, 2001.

[74] Alexandros Lazaridis, Elie Khoury, Jean-Philippe Goldman, Mathieu Avanzi, Sébastien Marcel, and Philip N Garner. Swiss french regional accent identification. In *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, pages 106–111, 2014.

[75] Cécile Woehrling and Philippe Boula de Mareüil. Identification of regional accents in french: perception and categorization. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'06*, pages 1511–1514, 2006.

[76] Adrian Leemann. *Comparative analysis of voice fundamental frequency behavior of four swiss german dialects: Elektronische daten*. PhD thesis, Selbstverlag, 2009.

[77] Yi-Jian Wu, Yoshihiko Nankaku, Keiichi Tokuda, et al. State mapping based method for cross-lingual speaker adaptation in hmm-based speech synthesis. pages 528–531, 2009.

[78] Yi-Jian Wu, Simon King, and Keiichi Tokuda. Cross-lingual speaker adaptation for hmm-based speech synthesis. In *Chinese Spoken Language Processing, 2008. ISCSLP'08. 6th International Symposium on*, pages 1–4. IEEE, 2008.

[79] Xianglin Peng, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Cross-lingual speaker adaptation for hmm-based speech synthesis considering differences between language-dependent average voices. In *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, pages 605–608. IEEE, 2010.

[80] Hui Liang and John Dines. An analysis of language mismatch in hmm state mapping-based cross-lingual speaker adaptation. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'10*, number EPFL-CONF-150600, 2010.

[81] Keiichiro Oura, Keiichi Tokuda, Junichi Yamagishi, Simon King, and Mirjam Wester. Unsupervised cross-lingual speaker adaptation for hmm-based speech synthesis. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 4594–4597. IEEE, 2010.

[82] Takenori Yoshimura, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Cross-lingual speaker adaptation based on factor analysis using bilingual speech data for hmm-based speech synthesis. In *Eighth ISCA Workshop on Speech Synthesis*, pages 297–302, 2013.

[83] Daiki Nagahama, Takashi Nose, Tomoki Koriyama, and Takao Kobayashi. Transform mapping using shared decision tree context clustering for hmm-based cross-lingual speech synthesis. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'14*, 2014.

[84] Mary Beckman, Manuel Díaz-Campos, Julia Tevis McGory, and Terrell A Morgan. Intonation across spanish, in the tones and break indices framework. *Probus*, 14(1):9–36, 2002.

[85] Hui Liang and John Dines. An analysis of language mismatch in hmm state mapping-based cross-lingual speaker adaptation. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'10*, number EPFL-CONF-150600, 2010.

[86] Joaquim Llisterri and José B Mariño. Spanish adaptation of sampa and automatic phonetic transcription. *Reporte técnico del ESPRIT PROJECT*, 6819, 1993.

[87] Mónica Caballero, Asunción Moreno, and Albino Nogueiras. Data driven multidialectal phone set for spanish dialects. In *Eighth International Conference on Spoken Language Processing*, 2004.

[88] Elra catalogue. emotional speech synthesis database. `http://catalog.elra.info`. Accessed: 2017-04-30.

[89] Qin Yan, Saeed Vaseghi, Dimitrios Rentzos, and Ching-Hsiang Ho. Analysis by synthesis of acoustic correlates of british, australian and american accents. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 1, pages I–637. IEEE, 2004.

[90] Michael L Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep neural networks for noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7398–7402. IEEE, 2013.

[91] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4277–4280. IEEE, 2012.

[92] Jing Huang and Brian Kingsbury. Audio-visual deep learning for noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7596–7599. IEEE, 2013.

[93] Andrew L Maas, Quoc V Le, Tyler M O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng. Recurrent neural networks for noise reduction in robust asr. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'12*, 2012.

[94] Anurag Kumar and Dinei Florencio. Speech enhancement in multiple-noise conditions using deep neural networks. *arXiv preprint arXiv:1605.02427*, 2016.

[95] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. Dynamic noise aware training for speech enhancement based on deep neural networks. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'14*, pages 2670–2674, 2014.

[96] Xue Feng, Yaodong Zhang, and James Glass. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 1759–1763. IEEE, 2014.

[97] Takaaki Ishii, Hiroki Komiyama, Takahiro Shinozaki, Yasuo Horiuchi, and Shingo Kuroiwa. Reverberant speech recognition based on denoising autoencoder. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'13*, pages 3512–3516, 2013.

[98] Felix Weninger, Jürgen Geiger, Martin Wöllmer, Björn Schuller, and Gerhard Rigoll. Feature enhancement by deep lstm networks for asr in reverberant multisource environments. *Computer Speech & Language*, 28(4):888–902, 2014.

[99] Sunit Sivasankaran, Aditya Arie Nugraha, Emmanuel Vincent, Juan A Morales-Cordovilla, Siddharth Dalmia, Irina Illina, and Antoine Liutkus. Robust asr using neural network based speech enhancement and feature simulation. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 482–489. IEEE, 2015.

[100] Chao Weng, Dong Yu, Shinji Watanabe, and Biing-Hwang Fred Juang. Recurrent deep neural networks for robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 5532–5536. IEEE, 2014.

[101] Frank Seide, Gang Li, and Dong Yu. Conversational speech transcription using context-dependent deep neural networks. In *Proceedings of International Speech Communication Association Conference, INTERSPEECH'11*, pages 437–440, 2011.

[102] Arun Narayanan and DeLiang Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7092–7096. IEEE, 2013.

[103] Felix Weninger, Shinji Watanabe, Yuuki Tachioka, and Bjorn Schuller. Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4623–4627. IEEE, 2014.

[104] Mengyuan Zhao, Dong Wang, Zhiyong Zhang, and Xuewei Zhang. Music removal by convolutional denoising autoencoder in speech recognition. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*, pages 338–341. IEEE, 2015.

[105] Michael Berouti, Richard Schwartz, and John Makhoul. Enhancement of speech corrupted by acoustic noise. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'79.*, volume 4, pages 208–211. IEEE, 1979.

[106] Sunil Kamath and Philipos Loizou. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. In *ICASSP*, volume 4, pages 44164–44164, 2002.

[107] Yi Hu and Philipos C Loizou. A generalized subspace approach for enhancing speech corrupted by colored noise. *IEEE Transactions on Speech and Audio Processing*, 11(4):334–341, 2003.

[108] Markos Dendrinos, Stelios Bakamidis, and George Carayannis. Speech enhancement from noise: A regenerative approach. *Speech Communication*, 10(1):45–57, 1991.

[109] Yariv Ephraim and Harry L Van Trees. A signal subspace approach for speech enhancement. *IEEE Transactions on speech and audio processing*, 3(4):251–266, 1995.

[110] Yariv Ephraim and David Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6):1109–1121, 1984.

[111] Israel Cohen. Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator. *IEEE Signal processing letters*, 9(4):113–116, 2002.

[112] Language Technologies Institute at Carnegie Mellon University. Cmu arctic databases. `http://www.festvox.org/cmu_arctic/`. Accessed: 2017-04-30.

[113] Nitish Krishnamurthy and John HL Hansen. Babble noise: modeling, analysis, and applications. *IEEE transactions on audio, speech, and language processing*, 17(7):1394–1407, 2009.

# Índice alfabético

Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA

# ACTA DE DISERTACIÓN PÚBLICA

No. 00005
Matrícula: 2143808488

SÍNTESIS DE VOZ BASADA EN MODELOS OCULTOS DE MARKOV Y ALGORITMOS DE APRENDIZAJE PROFUNDO

En la Ciudad de México, se presentaron a las 13:00 horas del día 15 del mes de noviembre del año 2017 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. JOHN CHARLES HENRY GODDARD CLOSE
DR. LUIS MARTIN ROJAS CARDENAS
DR. RENE MACKINNEY ROMERO
DR. PEDRO LARA VELAZQUEZ

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron a la presentación de la Disertación Pública cuya denominación aparece al margen, para la obtención del grado de:

DOCTOR EN CIENCIAS (CIENCIAS Y TECNOLOGIAS DE LA INFORMACION)

DE: MARVIN COTO JIMENEZ

y de acuerdo con el artículo 78 fracción IV del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

Aprobar

Acto continuo, el presidente del jurado comunicó al interesado el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.

MARVIN COTO JIMENEZ
ALUMNO

REVISÓ

LIC. JULIO CESAR DE LARA ISASSI
DIRECTOR DE SISTEMAS ESCOLARES

DIRECTOR DE LA DIVISIÓN DE CBI

DR. JOSE GILBERTO CORDOBA HERRERA

PRESIDENTE

DR. JOHN CHARLES HENRY GODDARD CLOSE

VOCAL

DR. LUIS MARTIN ROJAS CARDENAS

VOCAL

DR. RENE MACKINNEY ROMERO

SECRETARIO

DR. PEDRO LARA VELAZQUEZ