Reconocimiento Automático del Habla y Locución Orientado a la Manipulación de Sistemas

Eduardo Santos Mena

Carlos Alberto Olvera Olvera, José Ismael de la Rosa Vargas

Universidad Autónoma de Zacatecas
Unidad Académica de Ingeniería Eléctrica
Maestría en Ciencias de la Ingeniería
Zacatecas, México

Junio 2017

Reconocimiento Automático del Habla y Locución Orientado a la Manipulación de Sistemas

Ing. Eduardo Santos Mena

Tesis de grado presentada como requisito parcial para optar al título de:
Magister en Ciencias de la Ingeniería

Directores:
Dr. Carlos Alberto Olvera Olvera, Dr. José Ismael de la Rosa Vargas

Línea de Investigación:
Procesamiento Digital de Señales y Mecatrónica
Grupo de Investigación:
Procesamiento Digital de Voz

Universidad Autónoma de Zacatecas
Unidad Académica de Ingeniería Eléctrica
Maestría en Ciencias de la Ingeniería
Zacatecas, Zacatecas
Junio 2017

Dedicatoria

Al muy dulce recuerdo de mis abuelitas y abuelo materno (mi Nena, mi Toña retoña y mi Pino) a quienes he guardado y he salvado por siempre en mi corazón. A mi abuelo paterno (mi Mundo) que nos ha dado a mi padre y a mí el ejemplo de fortaleza, dedicación y humildad para avanzar en la vida de la mejor manera. A ellos mis viejos queridos, que en la vida me han colmado de alegrías y cariñosos y cálidos recuerdos, que me han dado la mejor herencia (mi familia) y el mejor porvenir (mi hogar).

A mi padre y a mi madre que han dado tanto y tanto por mí, y aún más; con quienes no podré saldar cuentas nunca. Mi madre, la mujer de mi vida, me ha enseñado el como se debe llenar el día con las personas que queremos; y a ayudar a los demás hasta que duela como la madre Teresa, y por eso la amo tanto. Mi padre, el ejemplo más grande que un hombre pueda tener, me ha mostrado el como se debe trabajar desde madrugada y día con día para ser imprescindible como decía Bertolt Brecht, aunque no pare de andar el reloj cucú. Dedico esta humilde culminación de una pequeña parte de mi vida a ustedes que son mi Mafalda, mi Garfield, mis Chicago bulls.

A mis hermanos acompañantes siempre de los más grandes momentos de la vida. Aldo tan leal y sincero (mi abogado además), Alan muy noble y muy capaz y Gaby siempre cariñosa y confidente... babosas alimañas.

A Monserrat por ser la acompañante incondicional en los altos y bajos de mi vida desde hace ya bastante tiempo. Por no soltarme de la mano y no dejar de recordarme las cosas importantes de la vida, y los besos. Te amo.

Mateo, mi amor, solo espero que el trabajo que esto representó pueda serte útil como guía para los planes que te estén porvenir. De corazón intento ser el ejemplo para ti como lo fue para mí mi padre y mi abuelo. Ahora eres muy pequeño y no sé si dejes de tener en la memoria cuanto te lleno de besos y abrazos todos los días, por eso es que quisiera que los mejores momentos lleguen a ti al leer esto y así, más que una dedicatoria, quisiera que vieras esto como un recordatorio... ahora mismo hablas dormido en cama.

...

Agradecimientos

Quiero agradecer en primer lugar a mis asesores de tesis.

Agradezco al Dr. Carlos Alberto Olvera Olvera por el apoyo brindado a lo largo de estos años, por sus enseñanzas como docente y como la gran persona que es, quisiera extender mis agradecimientos más hacia usted y lo que ha hecho por mi persona pero las palabras no me serían suficientes.

Agradezco al Dr. José Ismael de Rosa Vargas por el conocimiento trasmitido y por ser una gran guía y ejemplo del modo en que debiera ser un académico dedicado con la investigación para mejorar a nuestra sociedad.

Agradezco al Dr. Arturo Moreno Báez quien apoyó el desarrollo de este y otros trabajos realizados por mí de una forma ferviente y de quién sin duda he aprendido más. Sinceramente lo hago.

Agradezco a mis compañeros Pablo, Gustavo, Manuel, Ana y Angél por los buenos momentos llenos de risas, de discusión, y de halo (ándele por...).

Agradezco especialmente al Dr. Sven Verlinden por la invitación y gran hospitalidad en la estancia de investigación realizada en la West Virginia University. Let's go mountaineers.

Agradezco a la Maestría en Ciencias de la Ingeniería por la oportunidad de realizar mis estudios de maestría y al Consejo Nacional de Ciencia y Tecnología por el apoyo brindado a lo largo del desarrollo de este proyecto.

Resumen

Las características que nacen dentro del órgano tracto vocal encargado de producir el habla humana son ricas en información, tanto de los rasgos físicos del tracto vocal ("único en cada persona"), como del mensaje. En el presente documento se expone una forma de realizar un reconocimiento automático de locutor (RAL) y un reconocimiento automático del habla (RAH) sobre comandos genéricos orientados a manipular un sistema cualquiera al mismo tiempo que el productor del comando es identificado. El reconocimiento se realiza por medio de modelos ocultos de Markov entrenados con los coeficientes cepstrales MFCC's y MDLF's sobre una base de datos de tamaño medio. Se reporta también la manera en que las variaciones en los parámetros para la extracción de la información acústica codificada altera la precisión del reconocimiento, así como el efecto que tiene el modo en el que se definen los modelos estadísticos. Uno de los puntos más notorios logrados, es la tasa de reconocimiento del locutor, la cual alcanza un promedio general superior al 98 %.

Palabras clave: HMM, MDLF, MFCC, Procesamiento de voz, RAL, RAH.

Abstract

The characteristics that are born within the vocal tract organ responsible for producing the human speech are rich in information, both of the physical traits, of the vocal tract "unique in each person") and the message.This document presents how the recognition is performed through of hidden Markov models trained with MFCC's and MDLF's (cepstral coefficients) on a medium size database, it is also reported how the variations on the parameters for the extraction of the encoded acoustic information, modify the accuracy on the speech and speaker recognition, and the effects of the variation in the statistical models. One of the greatest achievements, is the rate accuracy recognition of the speaker reached which is in a general way an average, higher than 98 %.

Keywords: HMM, MDLF, MFCC, Voice processing, Automatic speech and speaker recognition.

Lista de Figuras

Lista de Tablas

Contenido

1. Introducción

El reconocimiento automático del habla (RAH) en conjunción con el reconocimiento automático de locutor (RAL) es un paso importante en el desarrollo de tecnologías de control, inteligencia artificial, seguridad informática y manipulación de dispositivos de forma remota con las que se puede decir que se cuenta con una amplia rama de aplicaciones. En el presente trabajo se crea un sistema capaz de realizar ambos reconocimientos sobre comandos que componen un control genérico que se orienta a la aplicación en cualquier sistema. El procesamiento digital de voz para el reconocimiento del habla, es decir; reconocer automáticamente mediante procesos computacionales letras, palabras u oraciones, tiene como cometido transformar una onda acústica producida por el sistema tracto vocal del ser humano en un arreglo digital. Dicha onda acústica contiene intrínsecamente información del mensaje que el locutor desea transmitir, en esta etapa de la investigación se puede conocer digitalmente el mensaje expuesto por el sujeto productor, más sin embargo de quién proviene dicha muestra acústica no. La siguiente etapa del procesamiento es decidir si el locutor productor es quien dice ser, esto a través de la comparación de las características extraídas del hablante contra una base de datos que contiene por otro lado, las características de las personas a verificar y posteriormente tomar una decisión de aceptación o rechazo. En particular el RAL forma parte de las tecnologías biométricas. La biometría es el estudio de rasgos físicos o de comportamiento de las personas con el fin de identificarles entre un conjunto de individuos. Dentro de los rasgos físicos se encuentra el análisis del iris, de retina, la geometría de la mano, la huella dactilar, ADN, etcétera y dentro del análisis de la información biométrica del comportamiento se encuentran por ejemplo la firma, la forma de andar, la dinámica de tecleo, la forma de uso de interfaces gráficas de usuario (GUI), la voz [4], entre otras; el RAL en particular puede clasificarse como parte de ambas.

En este tipo de tecnologías los usuarios aún están en proceso de aceptación para su uso comercial, en el ejemplo de las de voz la incorporación al mercado es buena, aunque dadas algunas aplicaciones como en el uso de GPS por voz en automóviles existen estudios los cuales aseguran que los usuarios desvían su atención cognitiva. “Hay una inminente crisis de seguridad pública por delante con la futura proliferación de estas tecnologías en los vehículos. Es hora de considerar limitar nuevas y potencialmente peligrosas distracciones mentales incluidas en los vehículos, sobre todo con la percepción errónea del público común que manos libres significa libre de riesgo” Robert L. Darbelnet, presidente de la AAA. Y en la tabla **1-1** se puede observar un comparativo de cuatro tipos de métricas [5].

De lo anterior podemos darnos cuenta que las tecnologías del RAL contra las demás tiene

	Huella	Facial	Iris	Voz
Tipo	Físico	Físico	Físico	Físico-comportamiento
Método	Activo	Pasivo	Activo	Activo
Tasa de error igual	2-3.3 %	4.1 %	4.1-4.98 %	0.1-0.86 %
Tasa de falsa aceptación (FAR)	2.5 %	4 %	6 %	0.75 %
Tasa de falso rechazo (FIR)	0.1 %	10 %	0.001 %	6.75 %
Coste	Medio	Alto	Muy alto	Medio-bajo
Aceptación	Media	Alta	Baja	Alta

Tabla 1-1.: Comparativa de tecnologías biométricas.

un buen desempeño presentando un bajo porcentaje en la tasa de error igual (error en la relación entre falso positivo y falso negativo), un coste medio-bajo y una alta aceptación social. En el tratamiento de señales digitales tanto para el RAH como en el RAL una de las principales consideraciones para su estudio es el comportamiento pseudoaleatorio que presenta la generación de sonidos del sistema productor humano. Se debe analizar una gran cantidad de datos en un periodo corto de tiempo (de 20 a 100 milisegundos) y principalmente se debe tomar en cuenta el ruido tanto eléctrico generado por la circuitería y el tipo de micrófono, así como el ruido acústico generado por sonidos ambientales y de grabación. La señal acústica generada por las diversas partes que contiene el conjunto de órganos productor del habla humana es rica en información, no solo del mensaje en sí, sino también de características únicas del hablante como su intervalo de frecuencias predominantes o incluso como es la forma física de los órganos productores, lo que ofrece la posibilidad de crear un sistema capaz de distinguir un mensaje oral y quién es el productor del mensaje.

Controlar un dispositivo eléctrico, electrónico o electromecánico mediante una orden hablada, además ser capaz de distinguir de qué sujeto proviene el comando, es sin duda un paso hacia la comunicación hombre-máquina en entorno de la inteligencia artificial y visto desde otro punto de vista; un control que es capaz de reaccionar a las ordenes orales de una persona al instante, ofrece la posibilidad de que el usuario realice alguna acción sin entrar en contacto directamente sobre ella, útil en personas con cierto tipo de discapacidades.

El presente trabajo documenta como por medio de la obtención de dichas características se crean modelos estadísticos para locutores y a la vez para los comandos a reconocer, y por ende llevar el proceso de RAH al mismo tiempo que se hace el RAL creando un sistema genérico de fácil migración.

1.1. Antecedentes

A lo largo del desarrollo tecnológico se han presentado diversas técnicas para llevar a cabo el reconocimiento de voz, mismas que se han dejado de usar en algún punto debido a las emergentes que mejoran tanto la tasa de reconocimiento como la velocidad de procesamiento. Existen metodologías que tienen como base la extracción de características acústicas como fuente de alimentación en la creación de modelos matemáticos, y por otra parte hay métodos que comparan directamente la forma de la onda de una muestra contra la forma de onda a reconocer.

El reconocimiento por cuantización vectorial o VQ (Vectorial Quantization) por sus siglas en inglés, es uno de los métodos de reconocimiento que hoy en día se siguen investigando sobre todo en torno al reconocimiento del locutor. En 2005 De Lara [6] usa la cuantización vectorial como método de coincidencia de características evaluado con muestras de habla de la base de datos en idioma español venezolana (SALA) con una frecuencia de muestreo que cae en la banda telefónica. El autor reporta buenos resultados bajo un esquema para reconocimiento independiente de texto.

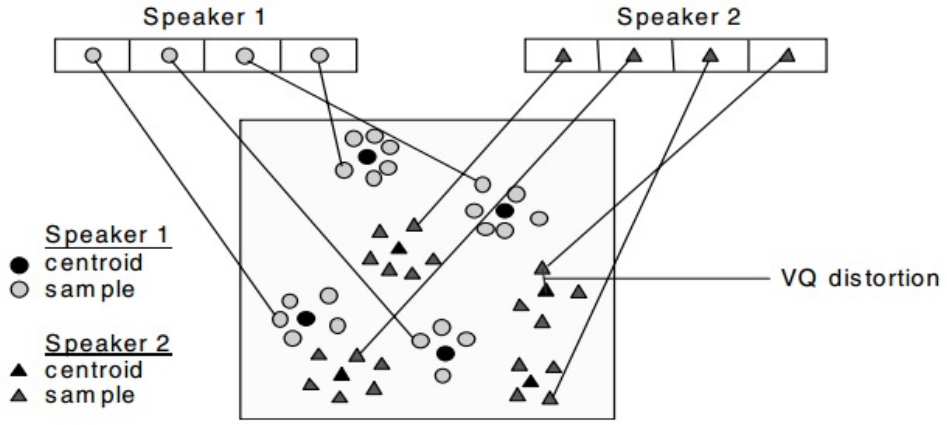

Figura 1-1.: Distribución del *codebeook* para la cuantización vectorial [1].

En la figura **1-1** se muestra un esquema de solamente dos locutores y los coeficientes correspondientes a dos dimensiones, los triángulos están asociados al primer locutor y los círculos al segundo. Cada círculo y triángulo representa a un coeficiente acústico respectivamente, dada la variabilidad en la codificación de la información debido a la dinamicidad del sistema vocal humano los coeficientes tienden a crear agrupaciones de coeficientes o clústers, lo que es una serie de valores cercanos entre sí relacionados en el tiempo con la compresión de la información. Se define entonces un valor centroide para cada clúster y los demás elementos en cada clúster se agregan en la etapa de entrenamiento generando una referencia espacial

entre el centroide y un *umbral de tolerancia*. La distancia desde cualquier vector de coeficientes de prueba al elemento más cercano del clúster asociado al locutor o *codeword* dentro de un grupo de locutores llamado *codebook* se define como la *distorsión VQ*. En la etapa de prueba, se extrae del audio la correspondiente información en su forma de coeficientes y se cuantiza vectorialmente usando el *codebook* y calculando la distorsión VQ, el locutor asociado al *codebook* que tenga la menor *distorsión* VQ es elegido entonces como el productor de las muestras a reconocer y será entonces identificado.

Otro método en el reconocimiento es el alineamiento dinámico temporal o DWT (Dynamic Time Warping) que fue propuesto en 1978 y ha sido principalmente aplicado en el reconocimiento del habla. Es común ver esta técnica en sistemas de reconocimiento como parte de la etapa de preprocesamiento debido al emparejamiento que hace entre un par de señales. El DTW se puede definir como un algoritmo para alinear secuencias no lineales [7] con el fin de encontrar coincidencia de patrones y se considera dentro de la denominada "programación dinámica". En el DTW se busca un mapeo óptimo a través de la evaluación entre una señal de prueba y una señal *plantilla*. El algoritmo tradicional llamado también *algoritmo de fuerza bruta* para realizar el DTW acarrea una gran complejidad computacional que crece exponencialmente a medida que aumenta la longitud de las señales y la frecuencia de muestreo. Y para el caso del tratamiento de voz, la carga de cómputo es grande ya que se generan matrices de n filas igual a la cantidad de muestras en la señal plantilla y m columnas igual a la cantidad de muestras en la señal prueba. En la figura **1-2** se muestra una representación de la distancia resultado del alineamiento de dos vectores con 35 muestras cada uno de una señal.

El aplicar el DTW en la etapa de preprocesamiento en reconocimiento de voz ayuda a disminuir la tasa de error ya que logra hacer que las ondas tengan una mejor relación entre los patrones que ayudan a identificarlas. Sin embargo, la misma distancia euclidiana obtenida en el proceso de emparejamiento se puede interpretar como una medida de igualdad entre las dos señales, y que a medida que la distancia disminuya; las señales alineadas tenderán a la igualdad en su forma y por consecuencia en su frecuencia. Es posible entones definir un sistema de reconocimiento de voz derivado del cálculo de las distancias entre la forma de las ondas en una base de datos contra una forma de onda de evaluación, donde la onda de la base que presente la distancia más baja contra la onda de evaluación sería elegida como muestra reconocida. En la figura **1-3** se observa como se alinean dos señales senoidales.

Los modelos ocultos de Markov o HMM (Hidden Markov Models) son modelos estadísticos derivados de una extensión de las cadenas de Markov y han sido una de las herramientas en el reconocimiento de voz más usadas en las últimas décadas. A su vez uno de los trabajos más citados en este ámbito es el de Rabiner de 1989 [3] donde expone de una manera muy cuidadosa y metodológica los aspectos teóricos y muestra como se usan los HMM en aplicaciones de reconocimiento de voz de modo muy detallado y ejemplificado. Los HMM

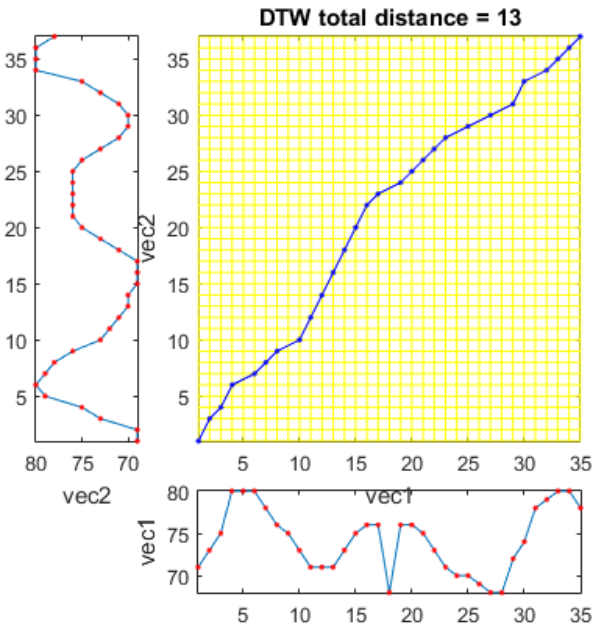

Figura 1-2.: Matriz resultado del DTW de dos vectores de 35 elementos cada uno [2].

son en sí una serie de matrices que contienen series de probabilidades y distribuciones de probabilidades, dichas matrices son creadas a partir de múltiples datos que son usados como entrenamiento. Las principales partes en un modelo oculto de Markov son: el número de estados, la probabilidad de iniciar el proceso en un estado cualquiera, la probabilidad de transitar de un estado a otro, y la densidad de probabilidades de que un estado produzca ciertas salidas. Los modelos típicamente son creados y evaluados por medio de la información acústica comprimida en forma de vectores de coeficientes. En la etapa de entrenamiento, múltiples muestras de audio crean una tendencia en las probabilidades que definen el modelo de forma iterativa. En la etapa de evaluación se calcula la probabilidad de que la muestra haya sido producida por el modelo, dado una colección de modelos; las muestras asociadas al modelo que arroje la probabilidad más alta de generar la muestra de evaluación son identificadas entonces (tanto para el reconocimiento de locutor como para el del habla). En la figura **1-4** se ilustran tres tipos de topologías (modo en el que se conectan los estados de un HMM).

En el reconocimiento de voz la topología de izquierda-derecha es una de las más usadas ya que restringe la transición a estados anteriores aprovechando las propiedades que presenta la señal de voz. Comúnmente se asocian los estados en los modelos con los fonemas, aunque no es necesario que cada estado corresponda a un solo fonema y viceversa, en la metodología de los HMM la relación entre estados y fonemas se crea por medio de los modelos de mezclas Gaussianas o GMM (Gaussian Mixture Models) generando un modelo acústico. Es precisamente en la generación del modelado acústico la parte en la que se ha contribuido

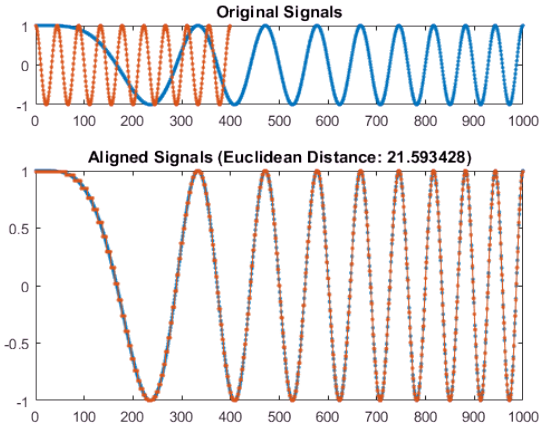

Figura 1-3.: DTW en dos señales senoidales de diferente longitud [2].

en la actualidad al mejoramiento de la tasa de reconocimiento, incluso se prescinde de los GMM sustituyéndolos por redes neuronales y en trabajos más recientes; redes neuronales profundas.

1.2. Estado del la cuestión

Redes Neuronales Profundas para el modelado Acústico en Reconocimiento de Voz

Como ya se ha mencionado una de las metodologías más usadas en el reconocimiento de voz ha sido la de HMM para manejar la variabilidad temporal y los GMM para determinar qué tan bien encajan los estados en la definición acústica para las ventanas de análisis. Una alternativa para evaluar la correspondencia entre estados-ventanas, es el uso de redes neuronales de alimentación hacia adelante que toma varios cuadros de coeficientes como entrada y produce probabilidades a *posteriori* en los estados de los HMM como salida. Las redes neuronales profundas que tienen muchas capas, han demostrado superar a los GMM (dependiendo del entrenamiento) [8].

Modelo híbrido DNN-HMM

Google [9] presentó un esquema de reconocimiento libre del uso de GMM. Las redes neuronales profundas o DNN (Deep Neural Network) se han convertido en el método dominante en el modelado acústico aunque aun así dependen de los GMM para llevar a cabo alineamientos en la fase de entrenamiento y otras. En su trabajo se introduce una DNN en la que no se usan en ninguna etapa los GMM simplificando el sistema de entrenamiento de una DNN. En dicho trabajo se demuestra como las DNN dependientes del contexto pueden ser entrenadas sin usar los GMM y además puede resultar en mejores modelos contra los GMM

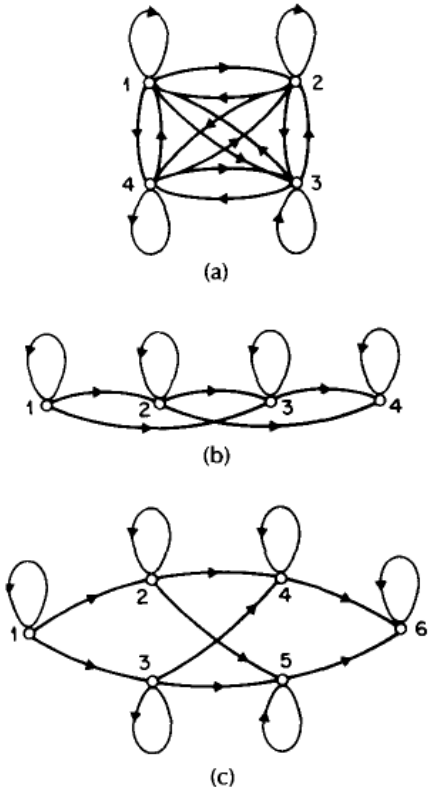

Figura 1-4.: Distintas topologías de un HMM. a) Modelo ergódico de 4 estados. b) Modelo izquierda-derecha de 4 estados. c) Modelo paralelo izquierda-derecha de 6 estados [3].

tradicionales. Es importante recalcar que ese trabajo muestra mejores resultados sin usar los GMM y que el autor aplica el método solo para el reconocimiento del habla. En la tabla **1-2** se muestran los resultados reportando un mejor desempeño cuando la precisión en los cuadros de análisis es mejor.

Red neuronal recurrente para reconocimiento robusto

También, 2014 Mitsibishi [10] propone una estructura de red neuronal recurrente o RDNN (Recurrent Deep Neural Network) para un reconocimiento robusto del habla. El nombre de *recurrente* es derivado del modo en que la red neuronal se interconecta y el modo en que se reestructura en la etapa de aprendizaje, nuevamente; se reporta el uso de una arquitectura híbrida DNN-HMM. Uno de los grandes problemas que siempre se han presentado en los sistemas de reconocimiento de voz es la presencia de ruido ambiental y en el estado de la cuestión se reportan métodos de modelado en adaptación de espacios con buenos resultados. Se ha demostrado además que los sistemas que trabajan con RNND en un modo de total reconexión entre las capas internas que la componen pueden crear un sistema robusto contra el ruido. Los resultados son mostrados en la tabla **1-3**.

Método/coeficientes	Dimensión	WER	FA_{ci}
GMM plpda	39	15.3 %	67.0
DNN plpda	39	15.2 %	67.0
DNN fb	40	15.1 %	66.8
DNN fbda	120	**15.0 %**	**66.8**
DNN ciscore	126	15.1 %	66.9
DNN ciact	1024	15.2 %	67.0

Tabla 1-2.: Tasa de error por palabra o WER (word error rate) para 2000 modelos y precisión por cuadro de análisis FA (Frame Accuracy).

Método	Sin ruido (WER)	Con ruido (WER)	Promedio
GMM	8.28 %	13.83 %	15.3 %
DNN I	3.51 %	8.15 %	15.2 %
DNN II	3.34 %	7.48 %	15.1 %
DNN III	3.24 %	7.44 %	**15.0 %**
RDNN I	3.27 %	**7.30 %**	15.1 %
RDNN II	**3.06 %**	7.26 %	15.2 %

Tabla 1-3.: WER para los alineamientos por cada método.$DNNI \sim III$ hasta 3^{er} capa oculta. $RDNNI \sim II$ hasta la 2^{da} capa recurrente.

DNN para reconocimiento de locutor

En el trabajo que presenta Matejka en 2016 [11] se aplican las DNN tipo cuello de botella en el reconocimiento de locutor. Aplicando GMM en el alineamiento de cuadros de análisis, se usan los coeficientes MFCC. Se muestran también los efectos que resultan en las variaciones de los GMM. Aunque el autor reporta buenos resultados, aplica como fuente de alimentación los coeficientes MFCC cuando se ha mostrado por Mahmood en 2014 [12] que los MDLF pueden tener mejores tasas de reconocimiento de locutor. Los resultados son presentados a continuación en la tabla **1-4**.

1.3. Planteamiento del problema

Durante la última década, el reconocimiento de voz ha sido un área de estudio en aumento debido a la capacidad computacional que se desarrolló en los dispositivos móviles, con lo que sistemas de reconocimiento del habla se comercializaron de forma amplia. El reconocimiento del habla es el área más explotada en el procesamiento de voz, dejando de lado al

Método	Coeficientes	Dimensión	EER
MFCC	MFCC	60	0.019917 %
BN	BN	80	0.020246 %
DNN	MFCC	80	0.012080 %
BN+MFCC	BN+MFCC	140	**0.009381 %**

Tabla 1-4.: Comparación de *error equal rate* (EER) entre coeficientes entrenados por 250 horas .

reconocimiento del locutor, y gran parte de esto se debe a que sistemas comerciales se han enfocado en ello. Esto es debido en gran parte, a que el reconocimiento del locutor requiere que el sujeto a reconocer genere la información suficiente en el corpus para que los modelos estadísticos funcionen correctamente, por lo que cada usuario debería entrenar a su propio dispositivo, siendo un inconveniente en los sistemas comerciales a diferencia del reconocimiento de voz que en el estado de la cuestión corresponde a un sistema de reconocimiento independiente del productor.

Uno de los principales inconvenientes en el desarrollo de sistemas de reconocimiento de voz es estar a la par de los sistemas comerciales como los que presentan Google, Apple y Microsoft sin mencionar otros. Esto se atribuye principalmente a la gran cantidad de métodos (con los que se trabaja a gran profundidad) que estos necesitan como: procesamiento de señales, programación, estadística, redes neuronales, etcétera; a la gran cantidad de información (Big Data) a la que se requiere acceso y a la carga de cómputo que cuesta llegar a la aplicación final. Por lo que aún es necesario comenzar desde una manera más sencilla, y en este trabajo se propone el uso de una metodología de HMM sin llegar hasta un sistema híbrido DNN-HMM como los que se presentan en el estado de la cuestión.

1.4. Justificación

Es fácil observar que la mayor parte de los avances en reconocimiento de voz se han hecho en el reconocimiento del habla, dejando en un rezago a los sistemas de reconocimiento de locutor y al reconocimiento habla/locutor. Por lo en el presente documento se propone la creación de modelos pares para realizar un reconocimiento simultaneo de locutor y habla. La gran mayoría de los sistemas comerciales ofrecen una muy baja tasa de error por palabra WER debido a la cantidad importante de recursos con los que cuentan. Por esa razón es un paso importante presentar un trabajo que siente las bases metodológicas y al mismo tiempo llevarlo a un nivel de calidad aplicable, es decir; que pese a las restricciones que puedan existir siga teniendo una tasa alta de reconocimiento.

Como ya se ha explicado con anterioridad, la complejidad de los algoritmos es alta al igual que la carga de cómputo. Así el uso de un programa de paquetería matemática es una opción válida para un primer acercamiento y al mismo tiempo dejando una ventana abierta para la migración de códigos a lenguajes de programación más veloces como **C** o **C++** que presentan más dificultad en la creación de los códigos necesarios pero que a su vez aumentarían en gran medida el tiempo de creación y evaluación de los modelos estadísticos.

Otro de los puntos importantes es la base de datos o corpus, ya que con el uso de redes neuronales profundas existe una relación entre la cantidad de datos con que son entrenadas y su desempeño, a medida que cuentan con más muestras en su estructuración ofrecen un mejor resultado. Ya que los fonemas son variantes entre idiomas, es difícil agrupar una cantidad de información así de basta de bases de datos libres para el idioma español, con lo que el sistema se plantea como un reconocimiento de unas cuantas palabras aisladas con un corpus de tamaño medio.

1.5. Hipótesis

Es posible la creación de un sistema de reconocimiento dual locutor/habla mediante el uso de pares de modelos creados por medio de un corpus de tamaño medio a una frecuencia de muestreo inferior a la frecuencia estándar de 16 kHz. De forma más específica, en este trabajo se plantea que por medio de un paquete matemático es posible crear los modelos estadísticos necesarios en una etapa de entrenamiento lo cual seguramente tomará un lapso de tiempo significativo, aunque en la etapa de evaluación el tiempo será lo suficientemente corto acercándose a un proceso en tiempo real (y por lo tanto a una aplicación en tiempo real). Como un punto complementario, se plantea que por medio de la correcta selección de los parámetros en la codificación de la información acústica y en la correcta definición de los modelos estadísticos se puede llegar a una buena tasa de reconocimiento dual, aunando las métricas usadas tanto para el RAH como las usadas para el RAL. Lo que por consecuencia se traduce en que con el uso de los HMM entrenados con los coeficientes MFCC y MDLF se puede llevar a cabo un reconocimiento simultáneo con una ligera carga de cómputo relativa.

1.6. Objetivos

El objetivo puntual del presente trabajo es la creación de modelos estadísticos duales HMM para el reconocimiento simultaneo de locutor y habla por medio de un corpus de tamaño medio. Se pretende también establecer una métrica para el reconcentramiento dual la cual supere al 90 % en precisión de reconocimiento (acierto/pruebas) para el reconocimiento de locutor y habla. Así mismo; evaluar la variación en la precisión del sistema cambiando los

valores de las variables en la extracción de la información acústica codificada y en la definición de los modelos estadísticos (número de estados y cantidad de mezclas en los GMM).

Objetivos particulares

- Crear una base de datos o corpus de la categoría *de menos de 50 productores* que tienen un contendido de palabras aisladas o un discurso entero con calidad de grabación de laboratorio según lo describe [13].

- Crear los modelos estadísticos necesarios para una seria de comandos orientados a una aplicación genérica.

- Crear los respectivos códigos para los algoritmos necesarios en una plataforma que facilite la programación y la manipulación de la información necesaria en el corpus sin llegar a una carga de cómputo alta para la etapa de evaluación.

- Evaluar la tasa de reconocimiento para múltiples cambios en las variables de caracterización de la voz y en las variables involucradas en los HMM de acuerdo con lo expuesto en la literatura y comparar los resultados.

1.7. Descripción del resto del documento

En el capítulo 2 se brinda una breve descripción matemática del proceso de extracción de los coeficientes acústicos usados como alimentación de los modelos estadísticos. Al hacer uso de los coeficientes cepstrales para los dos tipos de reconocimiento la metodología de obtención de estos es prácticamente la misma, con lo que se ahorra tiempo de cómputo en el proceso. En dicho capítulo se muestra el efecto en la forma de onda de cada paso justificado matemáticamente y se ilustran mediante el uso de gráficas los coeficientes obtenidos.

En el capítulo 3 se exponen los modelos ocultos de Markov, siendo la parte más importante del reconocimiento. Se observa como los modelos ocultos de Markov son una serie de matrices que contienen las probabilidades con las que se regirá el comportamiento que deberían seguir los vectores de observación, con lo que se pueden crear criterios de coincidencia entre vectores de observación (coeficientes cepstrales) entrenados y los vectores de observación a evaluar.

El capítulo 4 describe de un modo más puntual el proceso del sistema de reconocimiento aplicado y se presentan los algoritmos usados para obtener los modelos estadísticos y como se evalúan. También se describe la base de datos usada, así como el género, la edad y la frecuencia fundamental para cada productor.

Finalmente, en el capítulo 5 se muestran los resultados obtenidos realizando diversos cambios en las variables del proceso haciendo notar el porqué decae o aumenta la precisión del reconocimiento al modificar ciertos parámetros en la etapa de extracción de los vectores acústicos y en la definición de los modelos ocultos de Markov (modelos estadísticos).

2. Base matemática del RAH y RAL

2.1. Aparato fonador y la señal de voz

La onda acústica de presión que es reconocida como la señal de voz humana, es generada por una secuencia de procesos que involucran diferentes órganos del cuerpo, desde los pulmones e incluso el estómago, hasta los dientes. Podemos separar la creación de la voz desde el punto de vista de sistemas temporales en tres etapas básicas: entrada o fuente, respuesta al impulso y salida. Los órganos que comprenden las etapas de entrada y salida del sistema serían:

- Entrada.
  - Pulmones.
  - Bronquios.
  - Estómago.
  - Tráquea.
- Respuesta al impulso y salida.
  - Pliegues vocales.
  - Cavidad faríngea.
  - Cavidad bucal.
  - Cavidad nasal.

El aire que es generado por los pulmones pasa por los bronquios hacia la tráquea generando una columna de aire ascendente que va aumentando en presión por el estado de tensión permanente en el que se encuentran los pliegues o cuerdas vocales en reposo.Una vez que la cantidad de presión es suficiente para vencer la tensión de los pliegues vocales, esta se libera pasando entre ellos y los hace vibrar generando nuevamente baja presión entre la tráquea y los pliegues y así de forma repetitiva a una tasa de aproximadamente cien veces por segundo para los hombres y doscientas veces por segundo para las mujeres [14], este proceso es el responsable de generar la frecuencia de mayor amplitud o frecuencia fundamental F_0, también llamada *pitch*. Las diferentes cavidades por donde transitan las intermitentes columnas de presión son altamente resonantes, lo que agrega componentes de frecuencia a la señal final o sonido que es el diferencial entre la presión de las columnas y el aire de la faringe, obteniendo

la intensidad, de la presión y el tono, de la frecuencia F_0. La figura **2-1** ilustra el conjunto de órganos que trabajan juntos en la producción de voz.

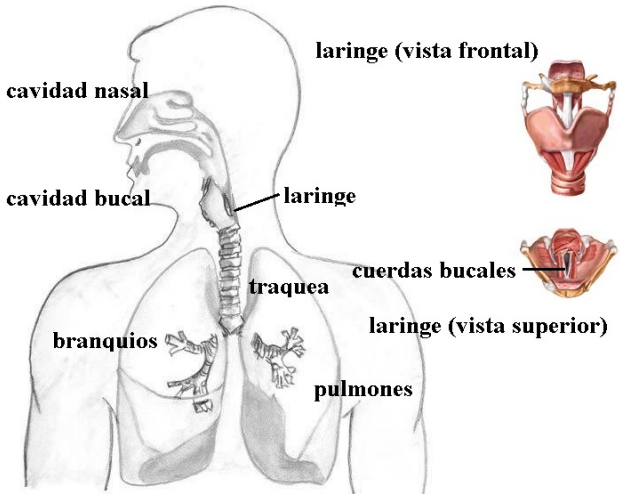

Figura 2-1.: Órganos relacionados en la producción de la voz.

Continuando con la analogía de la producción de voz como un sistema lineal invariante en el tiempo, definimos el aire que se genera en los pulmones o estómago como la fuente del sistema en el dominio temporal como $x(t)$ y al conjunto de los pliegues vocales, cavidades nasales, bucales, paladar, lengua, dientes, y demás como la respuesta al impulso $h(t)$, entonces la convolución entre entrada y respuesta al impulso $y(t) = x(t) * h(t)$ es la señal en tiempo de un fonema, variando entonces $h(t)$ con los fonemas que puedan existir. Resulta conveniente a su vez realizar la interpretación del sistema en el dominio de la frecuencia, por lo que:

$$F\{y(t) = x(t) * h(t)\} = Y(f) = X(f)H(f). \tag{2-1}$$

Donde la ecuación (2-1) hace uso de las propiedades de la transformación de dominios en donde la convolución de dos señales en tiempo equivale a la multiplicación de las señales en la frecuencia lo que resulta en la posibilidad de la separación de los elementos del sistema aplicando un logaritmo de acuerdo con la ecuación (2-2).

$$\log(Y(f)) = \log(X(f)) + \log(H(f)). \tag{2-2}$$

Suponiendo una respuesta al impulso glotal [15] como una función polinomial $h(t) = at^4 + bt^3 + ct^2 + dt$ para $a = -.001, b = .005, c = -.001, d = .005$ se obtienen una aproximación a la velocidad del fluido glotal en la producción de la voz como lo representa la figura **2-2**.

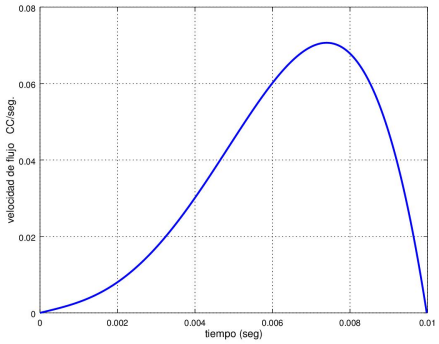

Figura 2-2.: Tasa de flujo volumétrico que representa la respuesta al impulso glotal.

Si se considera como fuente del sistema a un tren de impulsos $x(t) = \delta_T(t)$, la convolución $\delta_T(t) * [at^4 + bt^3 + ct^2 + dt]$ resulta en un tren de impulsos glotales, como se puede observar en la figura **2-3**.

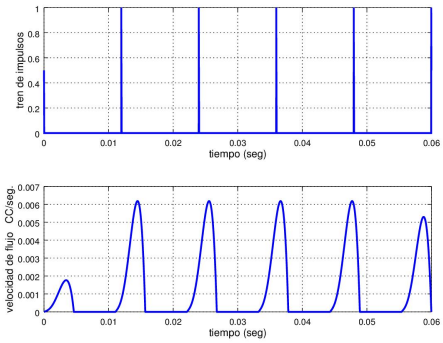

Figura 2-3.: Tren de impulsos a una frecuencia de 100 Hz y Tren de impulsos glotales resultado de la convolución de la fuente y la respuesta al impulso glotal.

Si ahora se analiza esta función resultante en el dominio de la frecuencia, es decir; $Y(f) = F\{h(t) * x(t)\}$, se observa que la frecuencia decae a un ritmo similar a $1/s^2$ como se observa en la figura **2-4**, lo que se puede traducir en un decaimiento de $12dB/octava$. El movimiento de la tráquea, las gesticulaciones, las frecuencias por resonancias y las turbulencias creadas en las cavidades terminan transformando a $y(t)$ en fonemas, los que pueden ser explicados como la unidad mínima del habla (aunque resulta ser un concepto abstracto más que una métrica definida). La conexión entre fonemas crea palabras y estas a su vez las oraciones; las palabras con múltiples conexiones de fonemas comúnmente tienden a ser muy aperiódicas en comparación con los fonemas simples vocalizados que tienden a la periodicidad sin llegar a ella, lo que da forma a la característica señal de voz. En la figura **2-5** se aprecia la forma de onda del fonema /a/. Los fonemas son denotados entre diagonales ej. /s/, /m/, /f/; etc.

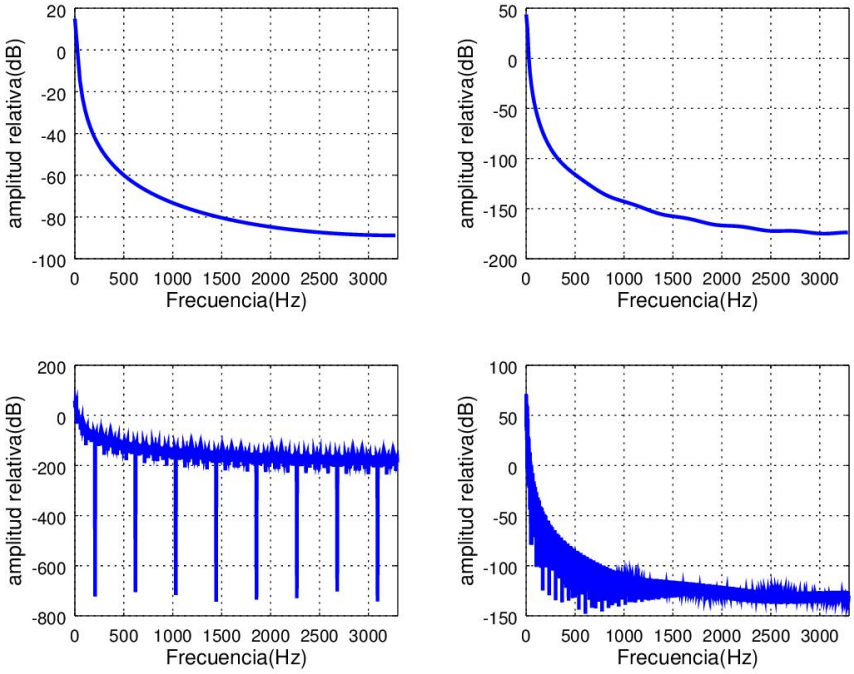

Figura 2-4.: Respuesta en frecuencia en diferentes resoluciones al tren de impulsos glotales. Superior izda.: resolución - 256 puntos, superior dcha.: resolución - 512 puntos, inferior izda.: resolución - 2048 puntos, inferior dcha.: resolución - 4096 puntos.

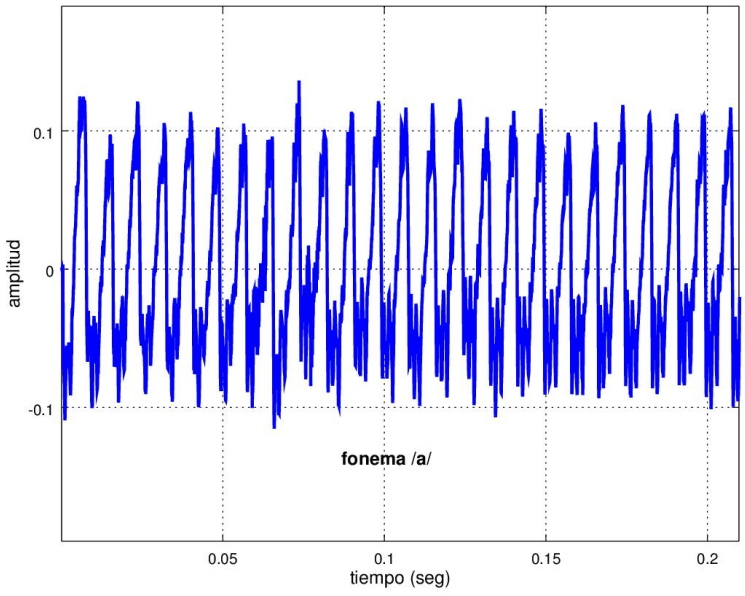

Figura 2-5.: Forma de la onda de presión acústica del fonema /a/.

2.2. Adquisición de la señal

La obtención de la señal de voz se puede llevar a cabo por diferentes tipos de formas, variando desde la frecuencia en la que se adquieren los datos, el tipo de transductor para convertir la onda de presión de las moléculas del aire en electricidad, canales de audio, número de bits para digitalizar la señal, ganancia en los canales, etc. En la actualidad, la gran mayoría de los ordenadores ya cuentan con un micrófono incorporado y una tarjeta de audio para la digitalización con lo que solo basta con definir el valor de las variables necesarias de la adquisición.

De acuerdo con el teorema de Nyquist la frecuencia a la que deben realizarse los muestreos de los datos para poder recrear la señal, es de dos veces la frecuencia máxima de la señal como mínimo, sabiendo que rara vez el habla natural humana supera los 4 kHz, la frecuencia de muestreo f_s es seleccionada en 14.7 kHz generando un efecto de sobremuestreo evitando el fenómeno de *aliasing*. El lapso de tiempo t_s entre muestra y muestra está definido por:

$$t_s = \frac{1}{f_s} = \frac{1}{14700} = 6.8027\text{x}10^{-6}s.$$

Siendo a su vez f_s una fracción de la norma 44.1 kHz (ancho de banda para el muestreo de audio) lo que simplifica procesos de submuestreo a un ancho de banda más adecuado para el rango de frecuencias de la voz. La tabla **2-1** muestra las frecuencias de muestreo estándar y sus respectivas aplicaciones.

Frecuencias estándar (kHz)	Aplicación
8	telefonía
22.05	radio
32	vídeo
44.1	audio (CD)
192.4	audio/vídeo (HD)

Tabla 2-1.: Estándares para frecuencias de muestro.

En términos de tiempo continuo se puede definir al muestreo como la multiplicación de la señal por un tren de impulsos separados por un periodo de muestreo t_s y cada pulso de amplitud unitaria. Así el producto punto de $y(t)$ con un tren de impulsos $\delta_{T=t_s*n}(t)$ define a la ecuación (2-3) de muestreo .

$$y\left[t_s n\right] = y(t) \cdot \delta_{t_s n}(t), \qquad \text{donde} \quad n \in \mathbb{N}. \tag{2-3}$$

Es común encontrar la notación simplificada como $y\left[n\right]$, lo que representa una señal en el dominio del tiempo discreto. La figura **2-6** ilustra en forma gráfica el resultado del teorema

de muestreo aplicado a una sección de 10 ms del fonema /a/ a una tasa de muestreo de 14.7 kHz. Es importante notar que n representa las muestras y que es un número entero positivo que incluye al cero. Para conocer la condición de la señal en el cambio de dominios la ubicación temporal puede conocerse por:

$$t(seg) = t_s(seg)n(muestra) = \frac{n(muestra)}{f_s(Hz)} = \frac{n(muestra)}{14.7kHz}.$$

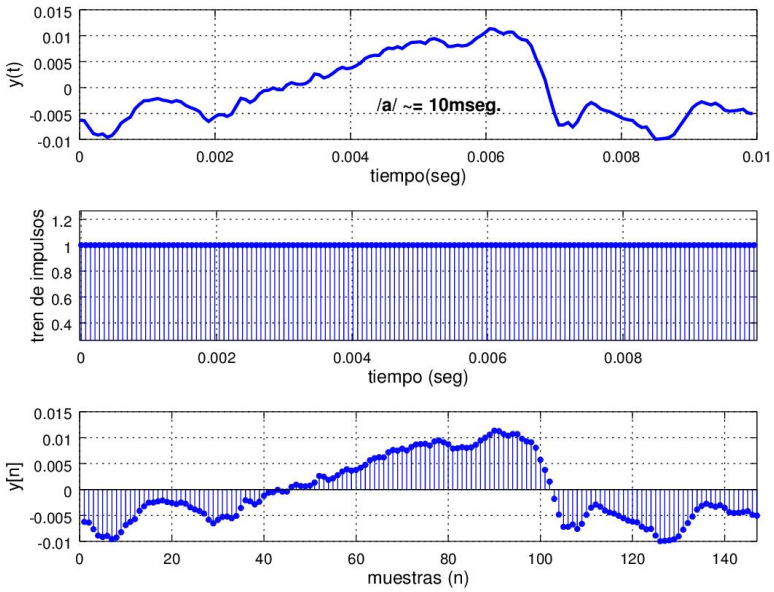

Figura 2-6.: Cambio de dominio temporal continuo-discreto.

2.3. Preprocesaminto

La etapa de preprocesamiento es fundamental para mejorar la tasa de reconocimiento ya que la variabilidad en la producción de la voz y el ruido (ambiental, eléctrico térmico, etc.) generan información fuera de fase e información estocástica sumamente difícil de modelar. El preprocesamiento consta de una serie de filtros digitales y un selector energético. La banda de frecuencia de interés para el procesamiento de voz suele fijarse entre 100 Hz y 3.8 kHz por lo que componentes en la onda que estén fuera de rango se consideran ruido y por lo tanto son atenuados. Sea $y(t)$ la onda acústica transformada en electricidad, podemos entonces definir a la señal contaminada con ruido como:

$$y_c(t) = y(t) + r(t) + C.$$

Donde $r(t)$ es la señal de ruido añadida y C es una constante que representa un nivel de corriente directa que puede ser cero, la operación de transformación de dominio:

$$F\{y_c(t)\} = F\{y(t)\} + F\{r(t)\} + F\{C\} = Y(f) + R(f) + C2\pi\delta(f).$$

Dicta que $R(f)$ existe para los rangos $0 \leq f < 100Hz$ y $3800Hz < f \leq \infty$, en el caso del tercer término(delta de Dirac) existe solo en $f = 0Hz$. Aunque ciertamente $R(f)$ puede estar presente en todo f y las necesidades de otra metodología de filtrado serían pertinentes (como filtros adaptativos). Se aprovecha el hecho de que en la práctica la magnitud de $Y(f)$ suele ser mayor a $R(f)$ para los valores de f en común; con lo que una atenuación $\forall f$ suele mitigar el efecto de $R(f)$ para frecuencias $100Hz \leq f \leq 3800Hz$ lo suficiente. Esta atenuación general suele lograrse por medio de modificaciones en los niveles de *offset* del transductor.

2.3.1. Filtro de voz

El diseño del filtro de voz está basado en los límites de la frecuencia que el habla natural humana suele alcanzar, y definida la frecuencia f_s en 14.7 kHz por el efecto de muestreo las frecuencias superiores a $f_s/2$ son despreciadas. Considerando un filtro con una banda basada en el procesamiento de voz, se pueden definir las frecuencias de 30 Hz y 3.3 kHz como las frecuencias de corte de un filtro pasa banda digital. Se propone un filtro IIR (infinite impulse response) tipo Butterworth (por la respuesta sin lóbulos en la banda de paso) de orden $n = 8$. La ecuación (2-4) describe la forma del filtro digital y 2-5 define la función de transferencia del filtro pasa voz $H_1(Z)$ en el dominio Z(domino de la frecuencia digital) del filtro y la figura **2-7** muestra la respuesta en frecuencia de $H_1(Z)$.

$$H_1(Z) = \frac{b_1 + b_2Z^{-1} + b_3Z^{-2} + \cdots + b_nZ^{-n}}{1 + a_2Z^{-1} + a_3Z^{-2} + \cdots + a_nZ^{-n}}, \tag{2-4}$$

$$\begin{aligned} H_1(Z) = \frac{0.0651 - 0.2605Z^{-2} + 0.3908Z^{-4} - 0.2605Z^{-6} + 0.0651Z^{-8}}{(1 - 4.3824Z^{-1} + 8.0976Z^{-2} - 8.6208Z^{-3} + 6.2147Z^{-4} \cdots} \\ - 3.1791Z^{-5} + 1.0255Z^{-6} - 0.1767Z^{-7} + 0.0213Z^{-8}). \end{aligned} \tag{2-5}$$

Para $Z = e^{i\omega}$ donde $\omega = 2\pi f$ es la frecuencia radial debido al dominio digital e i es la unidad imaginaria.

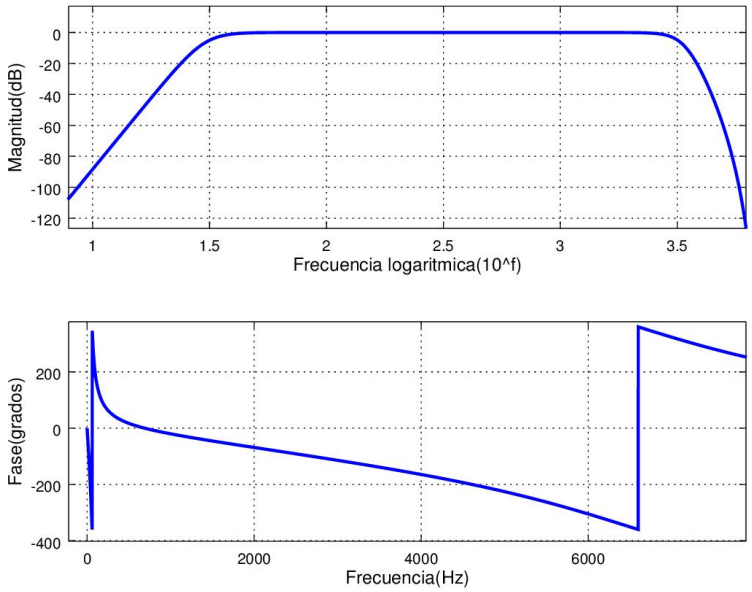

Figura 2-7.: Respuesta en la frecuencia del filtro pasa-voz. Arriba respuesta magnitud-frecuencia logarítmica. Abajo respuesta fase-frecuencia lineal.

2.3.2. Filtro de preénfasis

El filtro de preénfasis es utilizado en amplias ramas del procesamiento de voz y audio, en este caso cumple las funciones de atenuar las frecuencias bajas y realzar las frecuencias altas en la banda de voz con el fin de nivelar la energía espectral en toda la banda de procesamiento, se puede decir que cumple una función de ecualización dado que la densidad espectral de energía (DEE) en la banda de frecuencias bajas $S_b(f)$ es mucho mayor a la densidad espectral de energía en la banda de las frecuencias altas $S_a(f)$. La energía espectral puede calcularse como la integral de la función de la densidad espectral de energía:

$$E(f) = \int_{-\infty}^{\infty} S_b(f) + S_a(f)df = \int_{-\infty}^{\infty} |Y_b(f)|^2 df + \int_{-\infty}^{\infty} |Y_a(f)|^2 df.$$

Donde $Y_b(f)$ es igual a $Y(f)$ para $0Hz \leq f < 2000Hz$ y $Y_a(f)$ es igual a $Y(f)$ para $2000Hz \leq f \leq f_s/2$. La separación del espectro en estas dos bandas resulta útil para ilustrar la diferencia entre la alta energía en las frecuencias bajas y la poca energía en las frecuencias altas relativas, en la figura **2-8** se muestra la densidad espectral de energía para los fonemas /a/, /e/, /i/, /o/, /u/ donde la componente de mayor amplitud se sitúa alrededor de 120 Hz (frecuencia fundamental F_0), enseguida se encuentra una cantidad importante de energía en las frecuencias armónicas 240 Hz y 360 Hz aunque prácticamente son de un décimo de F_0.

El filtro de preénfasis consiste en un filtro digital de primer orden todo polos con la forma:

$$H_2(Z) = 1 - \alpha Z^{-1}, \qquad 0.94 \leq \alpha \leq 0.97$$

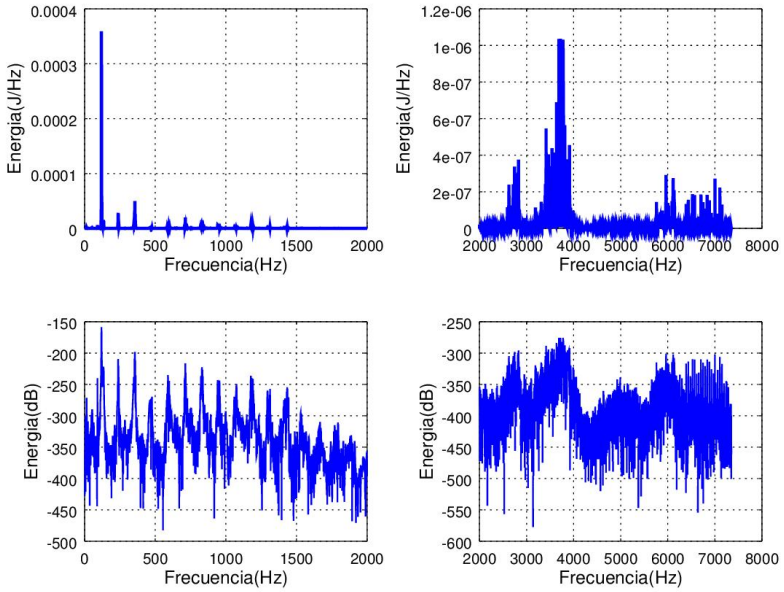

Figura 2-8.: Imagen superior izda. DEE banda baja. Imagen superior dcha. DEE banda alta. Imagen inferior izda. DEE logarítmica banda baja. Imagen inferior dcha. DEE logarítmica banda alta.

Donde α adquiere valores cercanos a uno, comúnmente entre 0.94 y 0.97 y las variaciones del valor α se reflejan en su respuesta en frecuencia del modo en que lo ilustra la figura **2-9**.

2.3.3. Detector de actividad de voz

Debido a las condiciones en las que se producen y se adquieren los datos, existen periodos de tiempo en los que solo está presente ruido ambiental de menor energía que la de la información de interés, provocando que la etapa de prueba de los modelos estadísticos empareje información altamente desfasada entre sectores de silencio, ruido, e información relevante. A su vez es natural que la actividad de interés esté desfasada entre muestras por lo que es propuesto un selector de actividad de voz basado en la energía promedio en ventanas de análisis rectangulares $w_e(t)$ donde la ventana de análisis está definida por:

$$w_e(t) = \begin{cases} 1, & \tau_1 \leq t \leq \tau_2, \\ 0, & \tau_1 > t > \tau_2. \end{cases} \tag{2-6}$$

Siendo τ_1 y τ_2 los límites de existencia de la función ventana. La energía temporal entre los límites τ_1 y τ_2 se puede calcular con la ecuación (2-7).

$$E_{\tau_1,\tau_2}(t) = \int_{\tau_1}^{\tau_2} |y(t)w_e(t)|^2 dt. \tag{2-7}$$

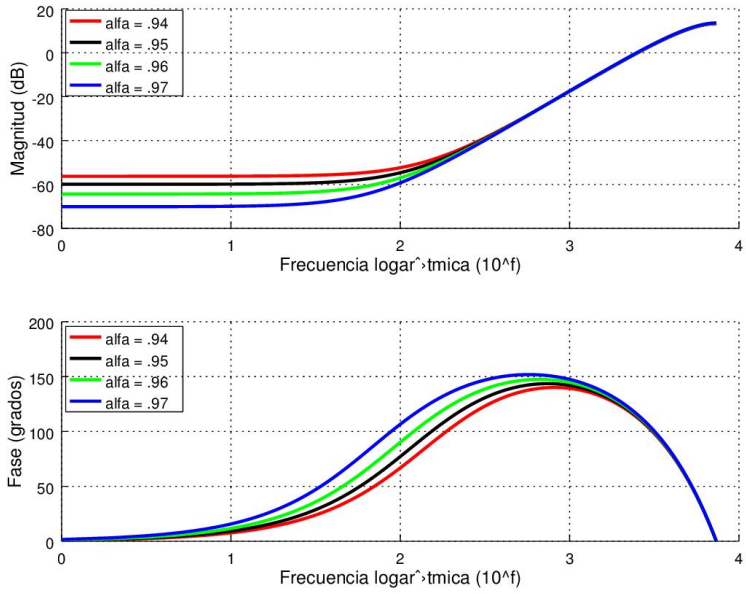

Figura 2-9.: Respuesta en frecuencia del filtro de preénfasis para $\alpha = 0.94$, $\alpha = 0.95$, $\alpha = 0.96$, $\alpha = 0.97$.

Para realizar el cómputo de la energía temporal en el domino discreto, el tiempo de duración de la ventana de análisis en muestras se establece por la multiplicación de la duración de la ventana por la frecuencia de muestreo $N = (\tau_2 - \tau_2)f_s$. Con lo que el vector de la energía temporal discreta por ventana se calcula con la ecuación (2-8)

$$E\left[T\right] = \sum_{n=TN}^{(T+1)N} |y(n)w(n)|^2. \tag{2-8}$$

Donde T es el número de ventana correspondiente. La detección de la actividad de voz se basa en la energía logarítmica del vector de energías $E[T]$ por lo que el nuevo vector de energías logarítmicas se calcula como:

$$E_{\log}[T] = \log(E[T]). \tag{2-9}$$

Se define un umbral $E_u$como separador de ventanas, donde aproximadamente 20 % de la energía logarítmica máxima suele ofrecer buenos resultados en la discriminación de ventanas de rangos de tiempo entre 20 ms y 50 ms. Si entonces $E_{\log}[T] \leq E_u$, la ventana de análisis T es desechada. En la figura **2-10** se muestra el resultado del proceso de la segmentación basada en un umbral de 22 % de la energía logarítmica máxima para $y(n)$ que contiene los fonemas vocalizados /a/, /e/, /i/, /o/, /u/.

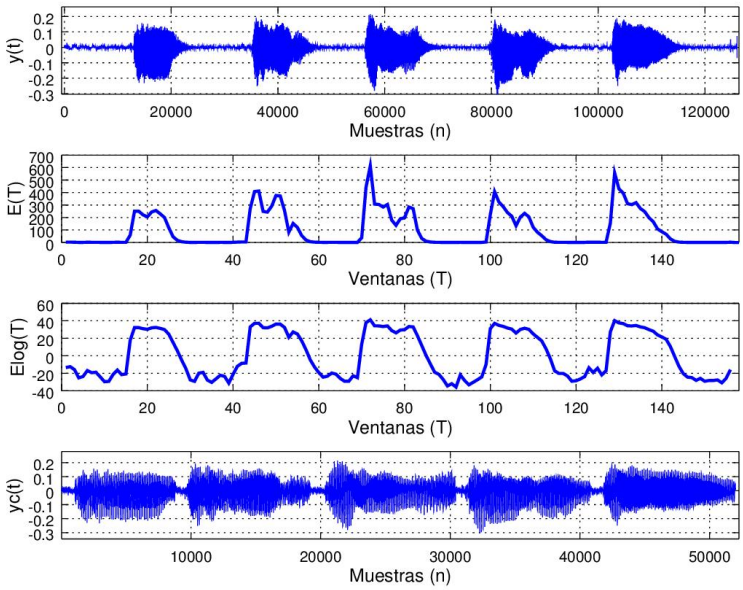

Figura 2-10.: y[t] audio con lapsos de silencio entre fonemas. E[t], energía correspondiente a cada fonema. $E_{\log}[T]$ energía logarítmica. $y_c(n)$ audio resultante del proceso de selector de actividad.

La energía para cada T en $E[T]$ tiende a aumentar directamente con el aumento de la duración de la ventana $w_e(n)$ aunque la resolución en la selección de actividad de voz tiende a bajar, lo que se traduce como la adición de más tramas de silencio en el recorte.

2.4. Extracción de características

El entrenamiento y prueba son etapas que están alimentadas con vectores columna de coeficientes representativos de las muestras $V_h[b]$, y comúnmente son usados coeficientes cepstrales, coeficientes resultantes de un proceso de transformación lineal tiempo-frecuencia y frecuencia-tiempo; de ahí el nombre de *cepstral*, derivado de la palabra *spectral*. Los modelos son creados y probados a partir de los coeficientes cepstrales en escala mel (MFCC) y coeficientes cepstrales de predicción lineal para el reconocimiento del habla y los coeficientes locales en múltiples direcciones y coeficientes derivativos sobre el espectrograma para el reconocimiento del locutor. Los coeficientes o características cepstrales comparten partes de la misma metodología para su extracción, lo que reduce el tiempo de cómputo para la producción de coeficientes pares que a su vez crean pares de modelos, unos para el RAL y los otros para el RAH.

Teniendo el tratamiento de una señal de audio $y(n)$ definido por ventanas de análisis, el resultado del procesamiento produce tantos vectores como número de ventanas y entre 18 a

26 elementos por vector, que corresponden al número de filtros F_{mel} sobre los cuales se agrupa la energía espectral para la compresión de dato. Posteriormente se agrupa la información de todos los filtros en las primeras bandas por medio de la transformada inversa de Fourier, aunque en el procesamiento digital de voz no es relevante la parte imaginaría del proceso de transformación, ya que esta es la información de la fase de señal; por lo que la transformada inversa de Fourier se puede llevar acabo por el método de transformada de coseno.

2.4.1. Tramas y fenómeno de Gibbs

Por la forma de onda de la voz, es impreciso tener un buen resultado al aplicar un cambio de dominio, una de las principales condiciones para aplicar este tipo de transformaciones es que la señal a tratar sea periódica y estacionaria, es decir, que sus características estadísticas no cambien en el tiempo. Puede considerarse a la señal de voz con dichas propiedades en lapsos del orden de decenas de milisegundos y es imprescindible analizarla en tramos en los que dichas condiciones se cumplan. De modo usual las ventanas de análisis varían entre 20 y 45 milisegundos en el procesamiento de voz.

El proceso de entramado se realiza a través de la multiplicación de la señal con una ventana rectangular (2-6) donde las tramas están traslapadas entre sí en un porcentaje típico del 30 % al 50 % de la longitud de la ventana. La señal en la trama $y_T(n)$ es el resultado de la multiplicación de la señal $y(n)$ con una ventana rectangular $w_r(n)$ de la forma:

$$w_r(n) = \begin{cases} 1. & 0 \leq n \leq N-1, \\ 0, & 0 > n > N-1. \end{cases}$$

Con lo que $y_T(n)$ se define formalmente por la ecuación (2-10).

$$y_T(n) = y(n')w_r(n), \qquad n = 0, 1, 2, \ldots, N-1 \tag{2-10}$$

Para $n' = n + (ovl)(T)$ donde T es el número de trama, ovl es el traslape en muestras que se obtiene al multiplicar el tiempo de retroceso por la frecuencia de muestreo $ovl = ovl_t f_s$ y N es el número total de muestras por ventana.

El fenómeno de Gibbs explica cómo se genera información errada por el proceso de integración sobre funciones discontinuas, las ventanas rectangulares generan el efecto de discontinuidad de salto de primera especie a los extremos de la función al faltar a la igualdad de aproximación por izquierda y derecha:

$$\lim_{n \to N^-} w_r(n) \neq \lim_{n \to N^+} w_r(n).$$

Se presenta una discontinuidad de salto finito con un salto:

$$salto = |w_r(N) - w_r(N+1)| = |1 - 0| = 1.$$

Este salto genera sobre información en las proximidades de $w_r(0)$ y $w_r(N)$ en la suma por series de Fourier (base para transformada Fourier) sin importar la cantidad de armónicos que se agreguen a la serie. Para ilustrar al fenómeno sea la serie:

$$S_c(t) = a_0 + \sum_{k=1}^{c} a_k \cos(kt) + b_k \sin(kt).$$

Y los coeficientes calculados para la función ventana rectangular en tiempo continuo $w_c(t)$ con periodo 2π:

$$a_0 = 1/\pi \int_{-\pi}^{\pi} w_c(t)dt = 1/2,$$
$$a_k = 1/\pi \int_{-\pi}^{\pi} w_c(t)\cos(kt)dt = \sin(k\pi)/k\pi,$$
$$b_k = 1/\pi \int_{-\pi}^{\pi} w_c(t)\sin(kt)dt = (1-\cos(k\pi))/k\pi.$$

La serie $S_c(t)$ presenta para todo valor de c un sobretiro de 9 % como se puede observar en la figura **2-11**, por lo que al aumentar en número de coeficientes c se mejora la aproximación $S_c(t) \to w_c(t)$ sin eliminar el error en las proximidades del salto.

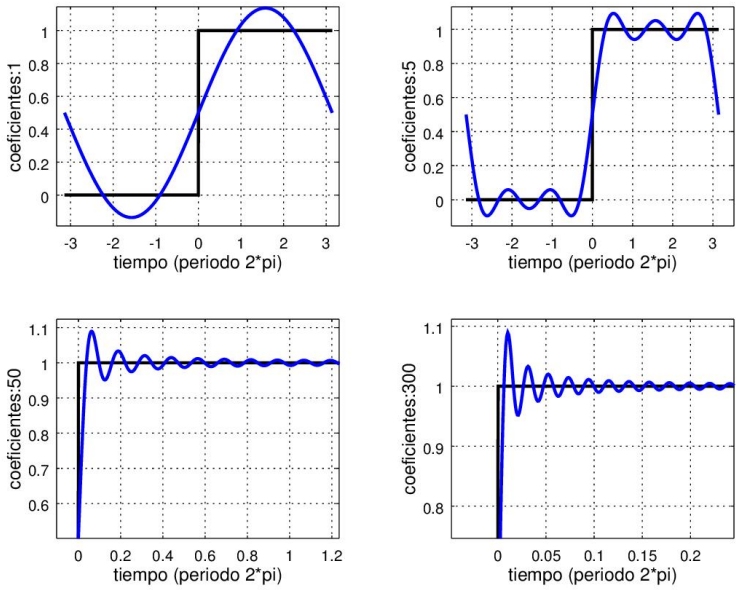

Figura 2-11.: Superior izda.: $S_c(t)$ para c = 1, superior dcha.: $S_c(t)$ para c = 5, inferior izda.: $S_c(t)$ para c = 50, inferior dcha.: $S_c(t)$ para c = 300.

2.4.2. Ventaneo

El aplicar tramas con funciones ventana de forma Gaussiana mitiga el fenómeno de Gibbs y aunque este seccionamiento de datos también genera información errada, el efecto que se genera es el de atenuación en la magnitud espectral en lugar de la adición de información espectral inexistente. Por la forma Gaussiana de la función ventana la atenuación de los datos se refleja a los límites de la función, por lo que al realizar la serie de traslapes las magnitudes reducidas en la ventana de análisis se realzan por la ventana siguiente. Las funciones ventana en el dominio temporal discreto más usadas en el procesamiento de voz son la ventana Hanning, Hamming, y Blackman que se definen por las siguientes ecuaciones.

$$w_{hm}(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad \text{Hamming},$$

$$w_{hn}(n) = 0.5 - 0.4\cos\left(\frac{2\pi n}{N-1}\right), \quad \text{Hanning},$$

$$w_{bm}(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) - 0.8\cos\left(\frac{4\pi n}{N-1}\right), \quad \text{Blackman}.$$

Con forma y respuesta en frecuencia como lo ilustra la figura **2-12** sin cambio en la fase.

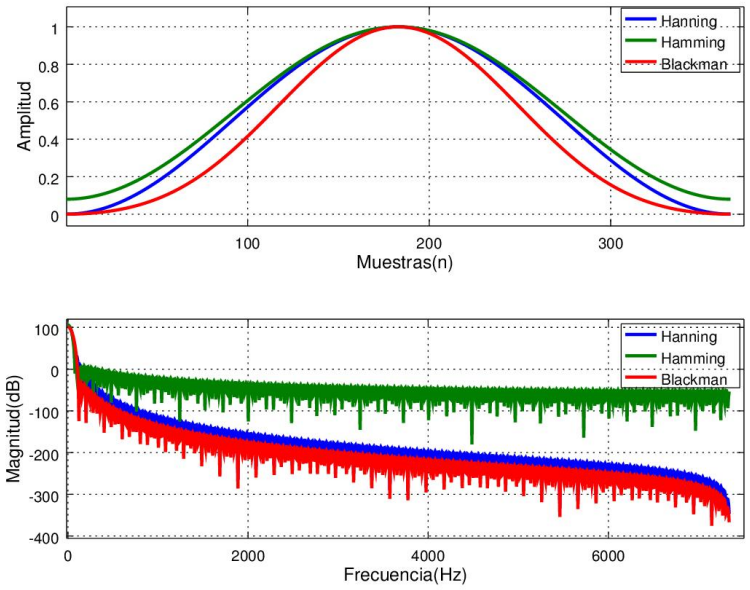

Figura 2-12.: Arriba, forma de las funciones ventana. Abajo, respuesta en frecuencia.

2.4.3. Transformada rápida de Fourier

La transformada rápida de Fourier (FFT) es un algoritmo que optimiza el tiempo de cómputo de la transformada discreta de Fourier (DFT) y no una nueva transformada en sí, aprove-

chando las propiedades de periodicidad y simetría en los fasores $e^{2\pi it}$ de la transformada discreta por lo que elimina cálculos redundantes en operaciones complejas. Siendo la transformada de Fourier (FT) la herramienta básica para realizar el cambio de domino temporal al dominio de frecuencia. El análisis de Fourier se basa en la descomposición de la señal en tiempo $y(t)$ a la serie de ondas senoidales de diferentes frecuencias que la constituyen. La FT en tiempo continuo se define en la ecuación (2-11).

$$Y(f) = \int_{-\infty}^{\infty} y(n)e^{-2\pi it/f}. \tag{2-11}$$

Donde $Y(f)$ es la descomposición compleja de la señal original $y(t)$ en funciones exponenciales constituyentes. La transformada discreta de Fourier por otro lado convierte una señal muestreada en el tiempo en una representación de frecuencia muestreada que arroja una relación entre amplitud y frecuencia. La transformada discreta de Fourier está dada por [16]2-12:

$$Y(k) = \sum_{n=0}^{N-1} y(n)e^{-2\pi ink/N}. \tag{2-12}$$

Para la señal original discreta $y(n)$, y N su longitud en muestras. Así suponiendo una secuencia de datos $y(n)$ de longitud $N = 4$ el cálculo para $Y(k)$ de modo tradicional para la DFT es:

$$\begin{aligned}
Y(0) &= y(0)e^{-2\pi i0*0/4} + y(1)e^{-2\pi i1*0/4} + y(2)e^{-2\pi i2*0/4} + y(3)e^{-2\pi i3*0/4},\\
Y(1) &= y(0)e^{-2\pi i0*1/4} + y(1)e^{-2\pi i1*1/4} + y(2)e^{-2\pi i2*1/4} + y(3)e^{-2\pi i3*1/4},\\
Y(2) &= y(0)e^{-2\pi i0*2/4} + y(1)e^{-2\pi i1*2/4} + y(2)e^{-2\pi i2*2/4} + y(3)e^{-2\pi i3*2/4},\\
Y(3) &= y(0)e^{-2\pi i0*4/4} + y(1)e^{-2\pi i1*4/4} + y(2)e^{-2\pi i2*4/4} + y(4)e^{-2\pi i3*4/4}.
\end{aligned}$$

Se puede observar que para cada valor de k se requieren N multiplicaciones complejas y $N - 1$ sumas complejas, desde otro punto de vista; para toda la secuencia deseada en k puntos, se requieren N^2 multiplicaciones complejas y $N^2 - N$ sumas complejas por lo que el cálculo en bruto requiere una cantidad importante de operaciones, de forma más específica se requieren:

$$O = 4N^2 + (4N^2 - 2N).$$

Traduciéndose en términos de operaciones O una multiplicación compleja en cuatro multiplicaciones reales y una suma compleja en dos sumas reales. Suponiendo una ventana de análisis de 25 ms con una frecuencia de muestreo de $f_s = 14700$, el número de operaciones

por ventana para conocer su espectro sería de $O = 1,076,778$. Para llevar a cabo el algoritmo de la FFT por diezmado en tiempo se comienza con el cambio de variable para el fasor $W_N = e^{-i2\pi/N}$, y haciendo uso de la igualdad de simetría: $W_N^{k+N/2} = -W_N^k$ y la igualdad de periodicidad: $W_k^{k+N} = W_N^k$. Es importante que el número de datos N en la secuencia $y(n)$ sea a base 2 para facilitar el cálculo de la FFT, de modo que si $N \neq 2^{\Delta}$ para $\Delta \in \mathbb{N}$ se rellena con ceros $y(n)$ hasta alcanzar un valor base dos para N, el relleno de con ceros en el vector temporal no altera la información espectral y puede usarse incluso como una herramienta para aumentar la resolución al espectro ya que $Y(k)$ constaría de más puntos para su representación. El siguiente paso es la separación de $y(n)$ en dos partes, la primera contiene los elementos con indexados pares del vector y el segundo a los elementos en los indexados impares, es decir: se divide $y(n)$ en un par de nuevas secuencias y estas a su vez vuelven a dividirse, de ahí el nombre de *diezmado en el tiempo*. Por lo que $Y(k)$ puede expresarse por las sucesiones separadas:

$$\begin{aligned} Y(k) &= \sum_{n=0}^{N-1} y(n) W_N^{kn}, \\ Y(k) &= \sum_{m=0}^{N/2-1} y(2m) W_N^{km} + \sum_{m=0}^{N/2-1} y(2m+1) W_N^{k(2m+1)}. \end{aligned} \tag{2-13}$$

Dado que $W_N^2 = W_{N/2}$, podemos reescribir la expresión (2-13) como:

$$\begin{aligned} X(k) &= \sum_{m=0}^{N/2-1} y_2(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{N/2-1} y_1(m) W_{N/2}^{km}, \\ X(k) &= Y_2(k) + W_N^k Y_1(k). \end{aligned} \tag{2-14}$$

Como $Y_2(k)$ y $Y_1(k)$ son el resultado de la transformación para $y_2(m)$ y $y_1(m)$ respectivamente, la longitud de ambos es $N/2$, además $Y_2(k)$ y $Y_1(k)$ son periódicas con periodo igual a $N/2$, esto es $Y_2(k+N/2) = Y_2(k)$ y $Y_1(k+N/2) = Y_1(k)$. En adición el factor $W_N^{k+N/2} = -W_k$, la ecuación (2-14) puede escribirse como:

$$X(k + N/2) = Y_2(k) - W_N^k Y_1(k), \quad \text{para } k = 0, 1, 2, 3, \ldots, N/2 - 1. \tag{2-15}$$

Con lo que se requieren $2(N/2)^2$ multiplicaciones complejas y $N/2$ sumas complejas para el cómputo de $Y_2(k)$ y $Y_1(k)$. De modo que para la decimación en dos secuencias, la reducción de operaciones para cada valor de k va de N^2 a $N^2/2 + N/2$ multiplicaciones complejas, lo que reduce la carga de cálculo acerca de la mitad. De igual modo, a medida que se van creando nuevas sucesiones, se va reduciendo la cantidad de operaciones necesarias. Llevando la separación de secuencias hasta las secuencias de un punto contra otro (dos secuencias de

un elemento), para $N = 2^{\Delta}$ el diezmado puede ocurrir hasta $\Delta = log_2(N)$, lo que se traduce en un reducción a $(N/2)log_2(N)$ multiplicaciones complejas y $Nlog_2(N)$ sumas complejas. La tabla **2-2** presenta la relación entre el valor N y la *velocidad* de cálculo [16].

Número de Puntos, N	Multiplicaciones Complejas en la DCT, N^2	Multiplicaciones Complejas en la FFT, $N/2log_2(N)$	Reducción de Cálculo, Veces
4	16	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1,024	80	12.8
64	4,096	192	21.3
128	16,384	448	36
256	65,536	1,024	64
512	262,144	2,304	113.8
1024	1,048,576	5,120	204.8

Tabla 2-2.: Comparación de la carga de cálculo entre la DFT y la FFT.

2.4.4. Banco de filtros mel

El banco de filtros mel es una serie de filtros triangulares pasa banda distribuidos uniformemente en escala de frecuencia mel sobre el rango de frecuencias deseado definido por sus frecuencias centrales $l_{fc}(m)$. Se hace uso de este filtrado sobre el espectro de la señal con el fin de mejorar la resolución de las frecuencias en bandas críticas del oído humano, donde cada filtro corresponde a una banda crítica. En un ancho de banda de 4 kHz se encuentran aproximadamente veinte filtros mel [17] y para realizar el cambio de frecuencia lineal a frecuencia en escala mel y viceversa se hace uso de las ecuaciones (2-16) y (2-17).

$$f' = 2595 \log_{10}(1 + \frac{f}{700}). \tag{2-16}$$

$$f = 700(e^{f'/1127} - 1). \tag{2-17}$$

Para conocer las frecuencias centrales en escala lineal primero se definen las frecuencias centrales en escala mel debido a que están linealmente distanciadas (en escala lineal no lo están). Esto depende del rango de frecuencia en el que el banco de filtros se defina existente y la cantidad de filtros en el banco N_M. El rango de frecuencias en el que opera el banco se calcula por la diferencia entre la frecuencia máxima f'_{max} y mínima f'_{min} de este, y las frecuencias centrales en escala mel como:

$$l'_{fc}(m) = m\frac{f'_{max} - f'_{min}}{N_M + 1}.$$

Donde $m = 1, 2, 3, \ldots, N_M$ es el m-ésimo filtro en el banco y se debe incluir a la frecuencia mínima como una frecuencia central inicial (por esa razón se ha de dividir el rango entre $N_M + 1$) Así por ejemplo para un banco de 12 filtros y las frecuencias $f_{min} = 0Hz$ y $f_{max} = 7350Hz$ para $f_{max} = f_s/2$ y $f_s = 14700Hz$, el vector que contiene las frecuencias centrales en escala mel sería:

$$l'_{fc}(m) = [0, 211.3, 422.7, 634, 845, 1.05k, 1.26k, 1.47k, 1.69k, 1.9k, 2.11k, 2.32k, 2.57k, 2.74k]$$

Y el vector con las frecuencias centrales en escala lineal:

$$l_{fc}(m) = [0, 144.6, 319.2, 529.9, 784.1, 1.09k, 1.46k, 1.9k, 2.44k, 3.09k, 3.88k, 4.82k, 5.97k, 7.35k].$$

Donde las frecuencias centrales $l_{fc}(m)$ en el rango de 0 a 1 kHz están prácticamente separadas de modo lineal. Los parámetros que definen el banco de filtros mel son:

- Número de filtros.
- Frecuencia mínima.
- Frecuencia máxima (depende de la frecuencia de muestreo f_s).

La construcción del banco de filtros se define por la ecuación (2-18) [17] y donde toma la forma en que lo ilustra la figura **2-13**:

$$M(m,k) = \begin{cases} 0, & \text{si } l_f(k) < l_{fc}(m-1), \\ \frac{l_f(k) - l_{fc}(m-1)}{l_{fc}(m) - l_{fc}(m-1)}, & \text{si } l_{fc}(m-1) \leq l_f(k) < l_{fc}(m), \\ \frac{l_f(k) - l_{fc}(m+1)}{l_{fc}(m) - l_{fc}(m+1)}, & \text{si } l_{fc}(m) \leq l_f(k) < l_{fc}(m+1), \\ 0, & \text{si } l_f(k) \geq l_{fc}(m+1). \end{cases} \tag{2-18}$$

La multiplicación del banco del filtros por el espectro $Y(k)$ conlleva a la convolución del filtro con la señal temporal $y_v(n)$ generando la agrupación de la densidad espectral de potencia sobre las bandas centrales de los filtros triangulares. El resultado de esta etapa arroja una matriz con una cantidad de columnas igual a la cantidad de ventanas de análisis y una cantidad de filas, igual a la cantidad de filtros en el banco como lo define la ecuación (2-19).

$$E_v(m) = \sum_{j=1}^{m} M(m,k) Y_v(k)^2. \tag{2-19}$$

Donde m es el número de banco, generalmente entre 18 y 26 y $Y_v(k)^2$ es la densidad espectral de potencia (DEP) estimada en la ventana de análisis v, la figura **2-14** muestra el efecto que tiene cada filtro del banco por separado $M(m,k)$ sobre el espectro $Y_v(k)$.

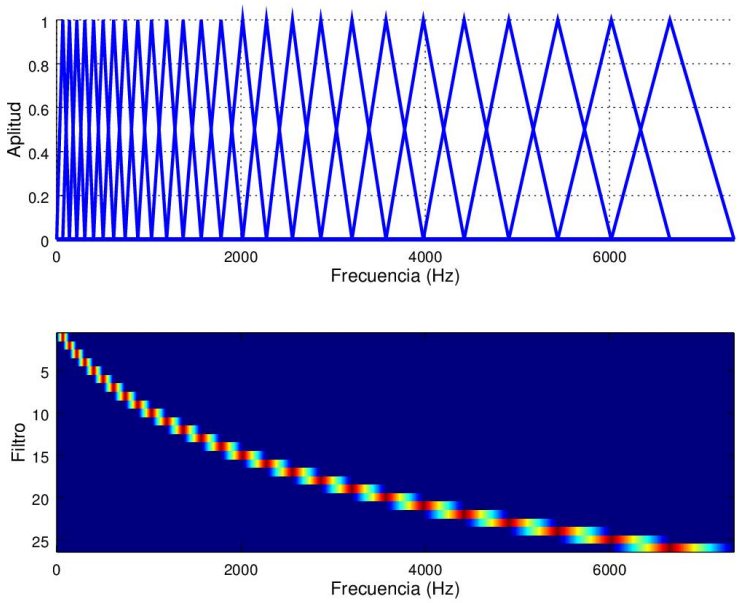

Figura 2-13.: Arriba, forma del banco de filtros de frecuencia lineal de 0 Hz a 7350 Hz. Abajo, agrupación de la energía en cada filtro.

2.4.5. Compresión logarítmica

La compresión logarítmica consiste en calcular el logarítmico del cuadrado de los coeficientes resultantes de 2-19, además de no representar una carga operacional en el cómputo significativo, ya que por propiedades del logaritmo algebraico es posible realizar el cálculo del logaritmo a la p-ésima potencia a través de la multiplicación de un solo logaritmo, esto es: $\log(g^p) = p \cdot \log(g)$. Este proceso de la magnitud y la compresión son realizados también por el oído humano. Los beneficios de esta operación se reflejan en la selección de únicamente de la magnitud ya que la fase es despreciable en el reconocimiento de voz, mientras que la aplicación del logaritmo genera una compresión dinámica, haciendo que el proceso de extracción de características sea menos sensitivo a las variaciones en dinamicidad [13].

Ya que la diferencia en energías agrupadas sobre el banco $M(m,k)$ es basta, especialmente entre los primeros y los últimos filtros del banco, se aplica un logaritmo a todos los elementos de la matriz resultante de la multiplicación del banco de filtros con la densidad espectral de potencia a cada ventana de análisis de frecuencia para una sucesión de datos en tiempo discreto $y(n)$, esto es:

$$E_{\log,v}(m,k) = \log\left(\sum_{j=1}^{N_M} M(m,k) Y_v(k)^2\right).$$

El efecto de la compresión logarítmica se puede observar en la figura **2-15**, donde se aplica

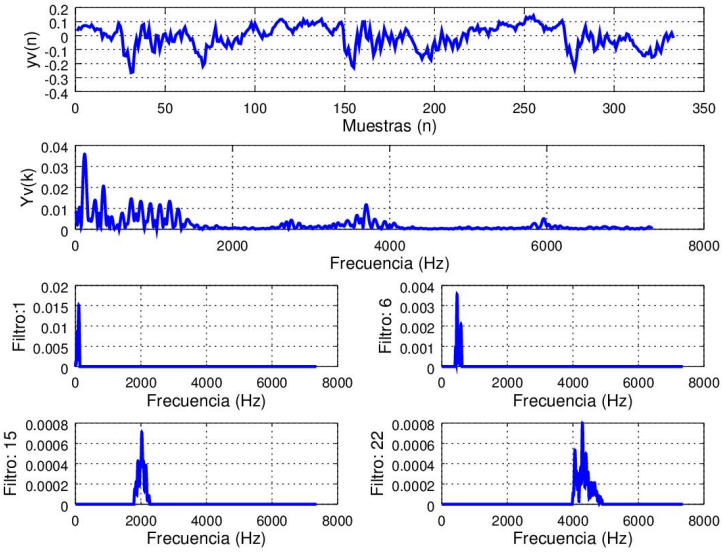

Figura 2-14.: Efecto de los filtros: 1, 6, 15, y 22 sobre la DEP para una ventana de análisis de longitud de 20 ms del fonema /a/.

a la v-ésima ventana de análisis del fonema vocalizado /a/.

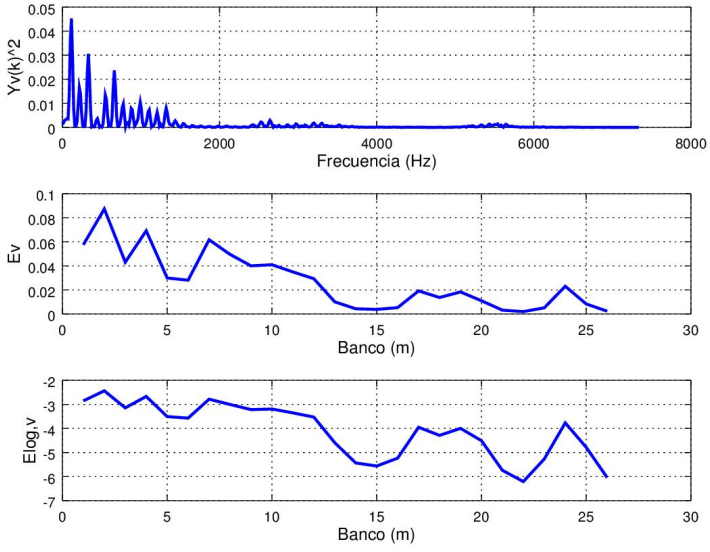

Figura 2-15.: Efecto de la compresión logarítmica para el fonema vocalizado /a/. Arriba densidad espectral de potencia. En medio, suma de la DEP para cada filtro en el banco. Abajo suma de la energía logarítmica en cada filtro del banco para la v-ésima ventana.

2.4.6. Transformada de coseno, análisis cepstral

El nombre de coeficientes cepstrales se deriva de un anagrama de la palabra *espectrales* y es debido a la aplicación de la transformada inversa de Fourier al logaritmo del espectro de la señal $y(n)$, $Y(k)$. Formalmente es la aplicación de la ecuación de síntesis de la transformación de Fourier al logaritmo del espectro, en tiempo continuo se define como:

$$y(t) = 1/2\pi \int_{-\infty}^{\infty} Y(e^{i\omega})e^{i\omega t}d\omega.$$

El complejo $Y(e^{i\omega})$ contiene la información de la fase y la magnitud correspondiente de señal temporal $y(t)$ por lo que para el uso de la ecuación de síntesis en reconocimiento de voz se puede prescindir de la fase como ya se ha mencionado con anterioridad, por lo que en tiempo discreto y contemplando solo $Y(k) = |Y(e^{i\omega})|$ se tiene:

$$y(n) = 1/N \sum_{k=0}^{N-1} Y(k)e^{2i\pi kn/N}.$$

Donde por identidad de Euler el fasor $e^{2i\pi kn/N}$ es el resultado de la suma de las funciones trigonométricas seno y coseno, siendo la función coseno la que se relaciona directamente con la parte real de la identidad, con lo que está ligada con la magnitud y por otra parte, la función seno está ligada con la parte imaginaria de la identidad y por consiguiente con la fase. Para la señal discreta $y(n)$ con fase mínima o despreciada, el análisis *cesptral* real de la parte real por medio de la transformada discreta de coseno (DCT) es [18]:

$$y'(n) = \log\left(\mathrm{DCT}\{|Y(e^{i\omega})|\}\right) = \log\left(\sum_{k=0}^{N-1} Y(k)\cos\left(k\frac{\pi}{N}(n+1/2)\right)\right).$$

Ya que la señal $y(n)$ es rica en armónicos añadidos a medida que las series de impulsos de aire pasan por el sistema tracto vocal, es analizada de mejor modo por los métodos cepstrales que por la correlación o el análisis espectral, el uso del análisis cepstral enfatiza las formantes que genera el tracto vocal, incluso con ruido [13].

Como se ha visto con anterioridad, la señal de voz puede definirse como la convolución en tiempo del sistema tracto vocal (respuesta al impulso) con la producción de aire (fuente). Así pues el análisis cepstral puede considerarse como un filtro *homomórfico* capaz de separar (deconvolucionar) de la señal de voz $y(n)$ la forma del tracto vocal, de la fuente. Esto es [18]:

$$\begin{aligned} y'(n) &= \mathrm{DCT}\{\log\left(\mathrm{DCT}\{h(n) * x(n)\}\right)\}, \\ y'(n) &= \mathrm{DCT}\{\log\left(H(k)\cdot X(k)\right)\}, \\ y'(n) &= \mathrm{DCT}\{\log\left(H(k)\right) + \log\left(H(k)\right)\} = h'(n) + x'(n). \end{aligned}$$

El aplicar la DCT es una opción que reduce las operaciones de cálculo en contra de aplicar la transformada inversa discreta de Fourier (IDFT) aunque para el reconocimiento de voz también es una opción viable aplicar la IDFT discriminando la parte imaginaria. EL proceso cepstral también conlleva a la compresión de información gracias a la DCT, llevando la mayor cantidad de información a los primeros términos del vector resultante. La figura **2-16** muestra los coeficientes cepstrales en frecuencia mel.

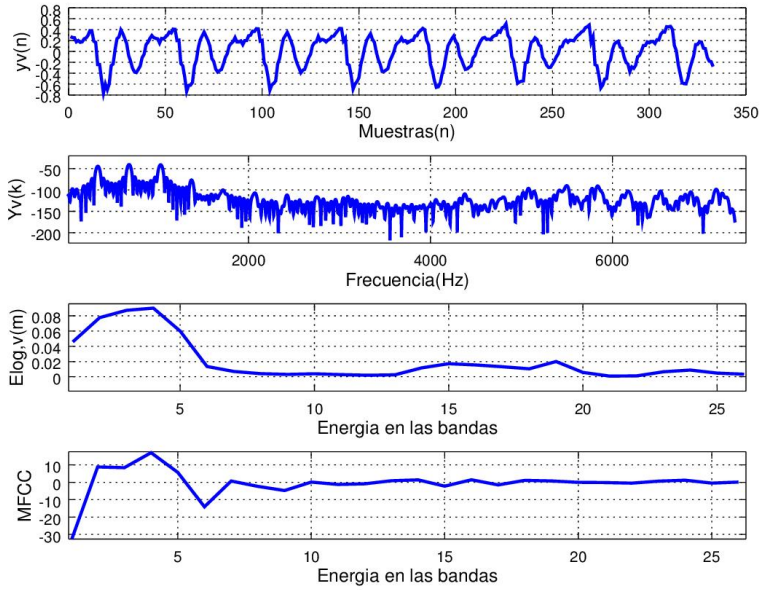

Figura 2-16.: $y_v(n)$, v-ésima ventana de análisis con longitud de 25 ms. $Y_v(K)$ DEP de la v-ésima ventana en dB. $E_{\log,v}(m)$, energía en cada banda del banco de filtros. MFCC, agrupación de la información en $E_{\log,v}(m)$ a los primeros términos del vector.

2.5. Caracterización y coeficientes para el RAH, MFCC

El reconocimiento automático del habla se lleva a cabo mediante la creación y prueba de modelos estadísticos creados a partir de métodos numéricos al igual que para el reconocimiento del locutor. Dichos modelos son alimentados por series de vectores que caracterizan con poca información a la onda mecánica longitudinal que se crea por el proceso del habla humana, aprovechado los patrones que permiten la identificación de los distintos fonemas. Los coeficientes utilizados en el RAH tienden a generar información en común sin importar el locutor que los produzca, aunque importando la cantidad de datos con los se creen los modelos. Si la cantidad de datos es suficiente para encontrar los rasgos propios entre fonemas, entonces los coeficientes habrán creado modelos independientes del locutor, de lo contrario

el reconocimiento contendría modelos insuficientes que incluso pueden presentar una tasa baja de reconocimiento y serán dependientes del locutor que haya creado la base de datos.

Los coeficientes cepstrales en escala mel son el resultado de aplicar la IDFT tomando únicamente la parte real o en otro caso aplicar la DCT a la compresión logarítmica de la energía en cada banda del banco de filtros, es decir; a la suma de la densidad espectral de potencia a lo largo de la existencia de cada filtro triangular sobre cada ventana de análisis, como se comentó a lo largo de la sección anterior. Cada ventana de análisis produce un vector de entre 18 y 26 elementos típicamente (un elemento por cada filtro en el banco). La información más relevante de cada vector se encuentra en los primeros elementos de este, por lo suelen ser despreciados los elementos posteriores al duodécimo o decimotercero conformando así una serie de tantos vectores como ventanas de análisis de tamaño de doce o trece elementos.

Como se puede observar en la figura **2-17** los coeficientes posteriores al onceavo elemento tienen poca magnitud en referencia a los primeros. Supongamos al vector de coeficientes MFCC de trece elementos de la v-ésima ventana como $C_v = [c_1, c_2, \ldots c_{13}]$, y existe la posibilidad de mejorar la tasa de reconocimiento agregando los coeficientes diferenciales y de aceleración, también conocidos como coeficientes deltas y coeficientes delta-delta que se obtienen con el par de ecuaciones (2-20) y (2-21).

$$c_{v,l}^{\Delta} = \frac{\sum_{j=0}^{J} j(C_{l+j} - c_{l-j})}{2\sum_{j=0} j^2}, \tag{2-20}$$

$$c_{v,l}^{\Delta\Delta} = \frac{\sum_{j=0}^{J} j(C_{l+j}^{\Delta} - c_{l-j}^{\Delta})}{2\sum_{j=0} j^2}. \tag{2-21}$$

2.6. Caracterización y coeficientes para el RAL, MDLF

Los MDLF (Multi-Directional Local Features) por sus siglas en inglés, son coeficientes cepstrales que están ligados directamente con los MFCC, ya que el proceso de extracción es prácticamente el mismo, y para obtener los MDLF basta con agregar una etapa en la que se calculan nuevas matrices por medio de regresiones lineales. Por el efecto de las regresiones, los coeficientes MDLF acarrean información de los cambios en el habla, el comportamiento en las consonantes, el poco cambio en las vocales, la transición entre formantes y el *onset-offset* de la señal [12].

Como ya se ha mencionado, los coeficientes MDLF son utilizados por los modelos HMM para llevar a cabo el reconocimiento automático del locutor y se obtienen siguiendo la misma

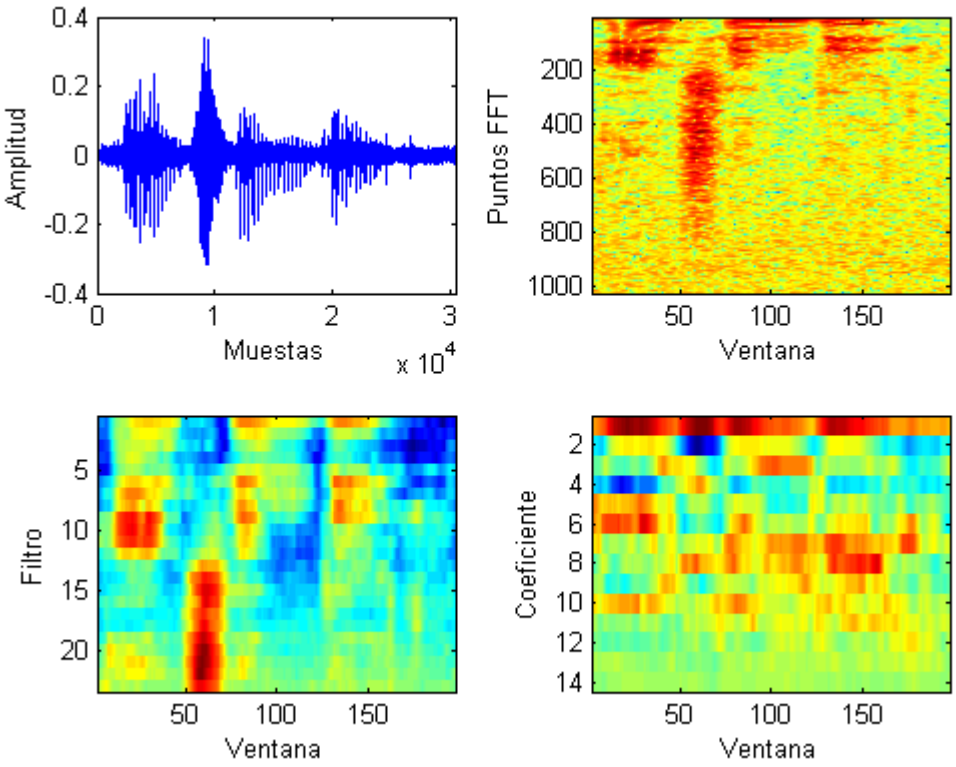

Figura 2-17.: a) Forma de onda de la palabra "Encender". b) Espectrograma con Wl = 100, Ol = 10. c) Energía acumulada en las bandas. d) Coeficientes Cepstrales en escala mel.

metodología de obtención que en los MFCC. Para su extracción, se agrega una etapa en la que se calculan regresiones lineales de 3 puntos en el eje de tiempo, frecuencia, tiempo-frecuencia en 45° y tiempo-frecuencia en 135° a las matrices que resultan en la agrupación de las energías, es decir; se aplican al resultado de la suma de la multiplicación del espectrograma con cada filtro del banco. Después se agrupa el resultado de cada regresión de modo similar a los *deltas* en los MFCC para generar una matriz final de n = número de coeficientes $* 4$ filas. Las regresiones se describen en las ecuaciones (2-22-2-25) y se ilustran en la figura **2-18**

$$d^{t}_{v,l} = \frac{\sum_{e=1}^{3} e\left(E_{t+e,f} - E_{t-e,f}\right)}{2\sum_{e=1}^{3} e^{2}}, \tag{2-22}$$

$$d^{f}_{v,l} = \frac{\sum_{e=1}^{3} e\left(E_{t,f+e} - E_{t,f-e}\right)}{2\sum_{e=1}^{3} e^{2}}, \tag{2-23}$$

$$d^{tf}_{v,l} = \frac{\sum_{e=1}^{3} e\left(E_{t-e,f-e} - E_{t+e,f+e}\right)}{2\sum_{e=1}^{3} e^{2}}, \tag{2-24}$$

$$d_{v,l}^{ft} = \frac{\sum_{e=1}^{3} e\left(E_{t+e,f-e} - E_{t-e,f+e}\right)}{2\sum_{e=1}^{3} e^2}. \tag{2-25}$$

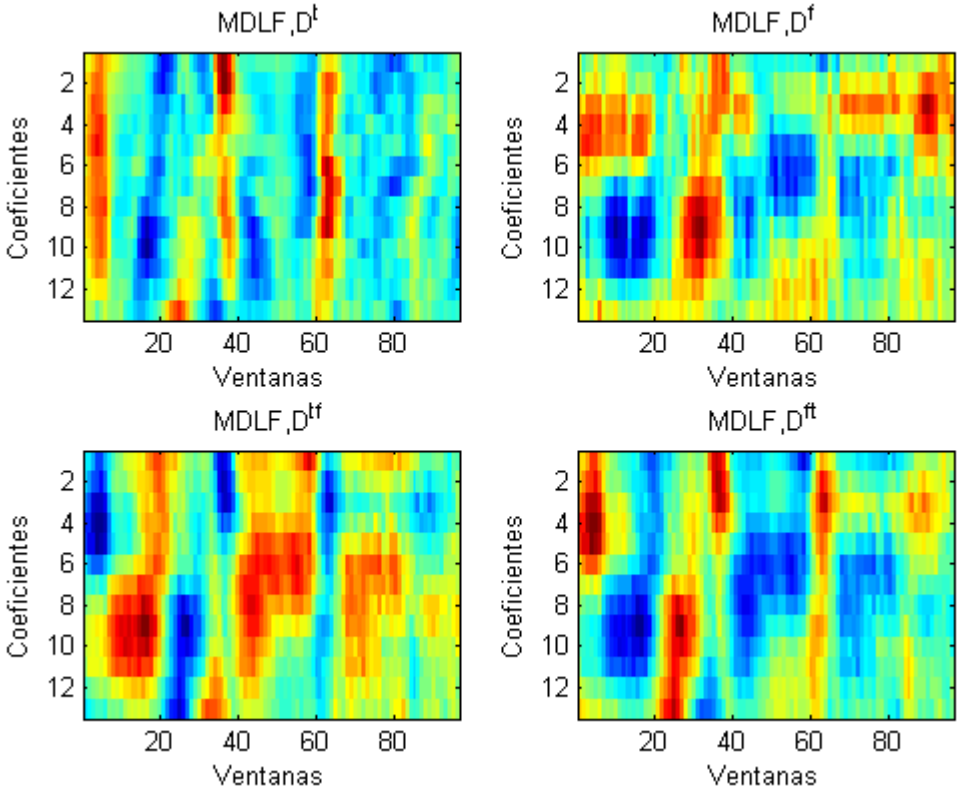

Figura 2-18.: Resultado de las regresiones lineales en tiempo, frecuencia, tiempo-frecuencia y frecuencia-tiempo sobre la energía acumulada en las bandas.

3. HMM como sistema de reconocimiento de voz

El reconocimiento de voz tanto para locutor como para el habla, se basa en el uso de los modelos ocultos de Markov o Hidden Markov Models (HMM) que son alimentados por coeficientes representativos de la forma de onda que se produce en el aparato tracto-vocal. Los modelos que son un conjunto de matrices de probabilidades son iniciados con valores que favorecen el entrenamiento de acuerdo con la misma naturaleza de la señal de voz. Una vez iniciados los modelos, estos son entrenados por algoritmos de máxima expectación (ME) a través de observaciones con lo que se modifican los valores en las matrices de probabilidades. Los modelos ya entrenados son probados a partir de observaciones entrantes, el modelo que describa mejor las observaciones se define como el correcto, y las observaciones asociadas a dichos modelos como el sonido pronunciado.

3.1. Modelos Ocultos de Markov

Los modelos ocultos de Markov son una extensión de las cadenas de Markov, donde los *estados* son desconocidos y se definen por las observaciones. Se puede decir que los HMM son modelos estadísticos creados por métodos numéricos apoyados de modelos de mezclas gaussianas o GMM (Gaussian Mixture Models) por sus siglas en inglés, para crear un modelado acústico. Un HMM define estados de transición que se rigen por una serie de probabilidades, y cada estado es asociado a una distribución de probabilidades de emitir observaciones. Aunado a la estructura antes mencionada, se debe definir un vector de probabilidades de iniciar en el estado cualquiera.

Sea un modelo oculto de Markov λ definido por los estados S, la matriz de probabilidad de transitar estados es entonces una matriz cuadrada A de NxN elementos, para $N =$ número de estados:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,j} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,j} \\ \vdots & & \ddots & \vdots \\ a_{i,1} & a_{s,2} & \cdots & a_{i,j} \end{bmatrix}, \quad 1 \leq i, j \leq N.$$

Las distribuciones de probabilidad de emitir símbolos se pueden definir entonces como un vector B de tantos elementos como especiación de símbolos posible M = número de símbolos:

$$B = \begin{bmatrix} b_1(k) & b_2(k) & \dots & b_M(k) \end{bmatrix}, \quad 1 \leq k \leq M.$$

Y el vector de probabilidades π de iniciar en el estado i como:

$$\pi = \begin{bmatrix} \pi_1 & \pi_2 & \dots & \pi_j \end{bmatrix}, \quad 1 \leq j \leq N.$$

Dados los símbolos $V = [v_1, v_2, \dots, v_M]$, de modo formal podemos definir las probabilidades de transición como: $a_{i,j} = P(s_{t+1} = j | s_t = j)$. La distribución de probabilidades $b_j(k) = P(o_t = v_k | s_t = j)$, y la probabilidad de iniciar en el estado i: $\pi_i = P(s_i = i)$. Por lo que un modelo se puede definir formalmente como el conjunto:

$$\lambda = (A, B, \pi). \tag{3-1}$$

3.1.1. Topología izquierda-derecha y uso de los HMM

Las topologías en los HMM establecen qué transiciones son permitas a través de los estados, una topología ergódica permite que se puede transitar de un estado dado a cualquier otro, mientras que una topología de izquierda a derecha no permita transiciones de un estado dado a un estado anterior e incluso no permite el avance hacia estados que superen limites definidos como se ilustra en la figura 3.1.1.

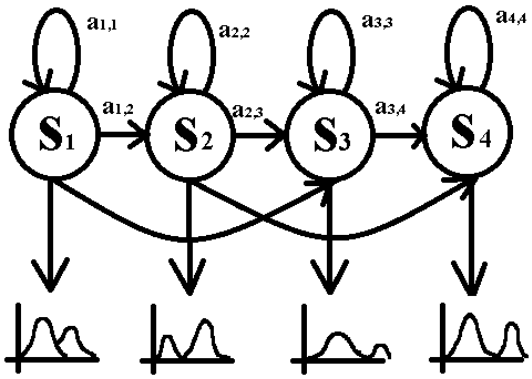

Figura 3-1.: Modelo de izquierda a derecha de cuatro estados.

Con lo que la matriz de transición de estados tendrá la forma:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0 \\ 0 & a_{2,2} & a_{2,3} & a_{2,4} \\ 0 & 0 & a_{3,3} & a_{3,4} \\ 0 & 0 & 0 & a_{4,4} \end{bmatrix}, \qquad \sum_{j=1}^{N} a_{i,j} = 1.$$

Definidos los valores apropiados para λ, se puede usar el modelo para generar una secuencia de observaciones $O = [o_1, o_2, \ldots, o_T]$ donde cada observación de O corresponde a un símbolo. El modo de usar el modelo como generador de observaciones es de acuerdo con el algoritmo **3.1** [3]:

Algoritmo 3.1: Creación de observaciones

1.-**Selección del estado de acuerdo a la distribución** π
2.-**Se define el estado actual** $t = 1$
3.-**Se selecciona la observación** O **de acuerdo a la distribución** $b(k)$
4.-**Avanza al estado** s_j **de acuerdo con la probabilidad de transición de estados** A
5.-$t = t + 1$**, regresa a** 3.-**. Si** $t > N$**, termina**

El procedimiento del algoritmo **3.1** puede utilizarse a su vez como generador de observaciones y como evaluador de observaciones, en otras palabras; dados los vectores de observación entrantes O, se evalúa qué tan bien el modelo λ describe el comportamiento de O, basado en la probabilidad de emitir símbolos en estado actual dado en modelo, formalmente $P(O|\lambda)$, y comparando los símbolos que genera λ contra los entrantes en O.

3.1.2. Problemas a resolver con los HMM

Los problemas que se enfrentan cuando se usan los HMM son básicamente problemas de eficiencia. Los algoritmos tradicionales para:

1) Calcular probabilidad de la secuencia O dado λ.

2) Dados λ y O ¿Cómo calculamos la secuencia de transición de estados que mejor describa el comportamiento de O?

3) ¿Cómo modificar A, B y π para maximizar la probabilidad $P(O|\lambda)$?

El problema 1 es el *problema de evaluación*, dados los vectores de observación O y el modelo λ, se debe calcular la probabilidad de que O haya sido producido por λ. Viéndose desde otro enfoque, se puede decir; que es un problema de evaluar que tan bien encajan los parámetros de un modelo dadas las observaciones. Si por ejemplo se crean los modelos para una base de reconocimiento de diez palabras se tendrían entonces diez modelos, con lo que una secuencia de vectores característicos extraídos del audio a reconocer, sería comparada con cada uno de los modelos, seleccionando al modelo que mejor encaje con dichas observaciones.

El problema 2 es referente a la transición de estados (que como ya se ha mencionado son ocultos). La solución buscada es conocer la secuencia de transición de estados, lo que en parte entra en conflicto con la aseveración de que "no existe una secuencia de estados correcta", sino simplemente una transición de estados "óptima", con lo que se debe definir un criterio de optimización para resolver este problema de la mejor forma posible orientado a la estructuración del sistema de reconocimiento en particular.

Por último, la solución al problema 3 busca modificar los parámetros del modelo, es decir; los valores en la matriz A y en los vectores B y π usando observaciones de varias ondas acústicas de las palabras o fonemas a reconocer para mejorar la caracterización de λ sobre O. Esta etapa también es llamada entrenamiento, donde las variaciones de O entre repeticiones dan un margen de *error* que mejora la tasa de reconocimiento.

Los algoritmos que se implementan para dar solución a los problemas básicos de los HMM son: algoritmo de *forward-backward* para el problema 1, el algoritmo de Viterbi para el problema 2 y el algoritmo de Baum-Welch para el problema 3.

3.1.3. Algoritmo forward-backward

Para calcular la probabilidad de la secuencia de observaciones $O = O_1, O_2, \ldots, O_T$ dado el modelo $P(O|\lambda)$ de modo tradicional, se deben calcular todas las secuencias de estados posibles con una longitud de estados, igual a la cantidad de observaciones T, considerando entonces la secuencia [3]:

$S = s_1, s_2, \ldots, s_T$.

Para la probabilidad de observar una secuencia de estados en particular se define por la multiplicatoria de la probabilidad de iniciar en el estado por las probabilidades de transitar los estados de la secuencia:

$P(S|\lambda) = \pi_1 a_{1,2} a_{2,3}, \ldots, a_{T-1,T}$.

Con lo que la probabilidad de observar una secuencia O dada la transición de estados es simplemente el producto de:

$P(O, S|\lambda) = P(O|S, \lambda)P(S, \lambda)$.

Para la probabilidad de tener una observación dada la secuencia y el modelo:

$P(O|S, \lambda) = b_{s_1}(O_1) b_{s_2}(O_2), \ldots, b_{s_T}(O_T)$.

Entonces la probabilidad de tener una secuencia O dado λ se obtiene por la suma de las probabilidades conjuntas sobre todas las secuencias de estados posibles:

$$P(O|\lambda) = \sum_{\forall S} P(S|\lambda)P(O|S, \lambda) = \sum_{\forall S} \pi_{s_1} a_{1,1} b_{s_1}(O_1) a_{1,2} b_{s_2}(O_2) \cdots_{T-1,T} b_{s_T}(O_T). \quad (3\text{-}2)$$

Con lo que el cálculo de (3-2) requiere una cantidad de operaciones para todas las combinaciones posibles de transición de estados por la cantidad de observaciones, esto es una cantidad de operaciones de $Op = 2TN^T$, con lo que para una secuencia de 100 observaciones (observaciones en un segundo de datos con una ventana de 20 ms con 10 ms de traslape) con cuatro estados, la cantidad de operaciones sería entonces de: $\approx 3.213\text{x}10^62$.

Considerando que las probabilidades $a_{i,j}$, π_i y $b_i(k)$ son conocidas, el algoritmo de **forward-backward** define una variable $\alpha_t(i)$ como:

$\alpha_t(i) = P(O_1O_2 \cdots O_t, s_t = s_i|\lambda)$.

La probabilidad de que se generen las observaciones $O_1, O_2, \ldots, O_t$ (t observaciones) con el estado s_i en tiempo t dado λ. Se puede resolver entonces α como [3]:

1) Inicio:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N,$$

2) Inducción:

$$\alpha_{t+1}(j) = \left[\sum_{j=1}^{N} \alpha(j)a_{i,j}\right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N. \tag{3-3}$$

3) Terminación:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{3-4}$$

3.1.4. Algoritmo de Viterbi

Como se ha mencionado con anterioridad, el problema 2 al que da solución el algoritmo de Viterbi, radica en que todas las transiciones de estados pueden ser *correctas* bajo la restricción de la topología. Con lo que se pretende la secuencia de estados *óptima*, que explique la mejor secuencia $S = [s_1, s_2, \ldots, s_T]$, para las observaciones $O = [O_1, O_2, \ldots, O_T]$. Entonces se define la probabilidad $\gamma_t(i)$:

$$\gamma_t(i) = \max_{s_1,s_2,\ldots,s_t-1} P[s_1, s_2, \ldots, s_t = i, O_1, O_2, \ldots, O_t|\lambda]. \tag{3-5}$$

Con lo que $\gamma_t(i)$ es el mejor *score* a lo largo de un único camino en el tiempo t, es decir; lo que ocurre para las primeras t observaciones y termina en el estado S_i, por lo que se tiene:

$$\gamma_{t+1}(j) = (\max_i \gamma_t(i))a_{i,j}b_j(O_{t+1}). \tag{3-6}$$

Para recuperar la secuencia de datos, se necesita rastrear el argumento que maximice a la ecuación (3-6) para cada tiempo t y j. Generando un arreglo ψ_t(j) de modo que los pasos son:

1) Inicio:

$$\gamma_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N,$$
$$\psi_i(i) = 0.$$

2) Recursión:

$$\gamma_t(i) = \max_{1 \leq i \leq N} [\gamma_{t-1}(i) a_{i,j}] b_j(O_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N,$$
$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\gamma_{t-1}(i) a_{i,j}], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N.$$

3) Fin:

$$P^* = \max_{1 \leq i \leq N} [\gamma_T(i)],$$
$$s_T^* = \max_{1 \leq i \leq N} [\gamma_T(i)].$$

4) Secuencia de estados (rastreo hacia atrás):

$$s_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1.$$

Es importante recalcar que el algoritmo de Viterbi es similar al algoritmo *forward-backward* sin tomar en cuenta el paso 4. Sin embargo, se usa la maximización sobre el estado anterior en lugar del procedimiento de suma en las ecuaciones (3-3) y (3-4). Es importante también aclarar que la estructura de *enrejado* conlleva a un cálculo eficiente.

3.1.5. Algoritmo Baum-Welch

El tercer problema busca modificar los parámetros por medio de algoritmos de máxima expectación para mejorar la probabilidad de observar una secuencia dado un modelo. Dada una secuencia finita de observaciones como entrenamiento, y aunque no hay un modo óptimo para estimar los parámetros del modelo, podemos escoger valores en $\lambda = A, B, \pi$ de modo que la probabilidad de generar una secuencia $P(O|\lambda)$ se maximice localmente, se pueden usar métodos iterativos como búsquedas a partir del gradiente o como el método de Baum-Welch. Para el uso del algoritmo Baum-Welch se define la probabilidad de estar en el estado s_j en el tiempo t y la probabilidad de estar en el estado s_j en el tiempo $t+1$, dados λ y O:

$$\xi_t(i,j) = P(s_t = s_i, s_{t+1} = s_j | O, \lambda).$$

De acuerdo con la con la definición de la variable auxiliar de avance del algoritmo de forward-backward:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \lambda).$$

Y usando una variable de retroceso:

$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T | q_t = S_i, \lambda)$.

Podemos definir $\xi_t(i,j)$ en términos de $\alpha_t(i)$ y de $\beta_t(j)$ como:

$$\xi(i,j) = \frac{\alpha_t(i) a_{i,j} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i) a_{i,j} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{i,j} b_j(O_{t+1}) \beta_{t+1}(j)}.$$

Donde el numerador corresponde a la probabilidad de que el estado oculto actual y el estado oculto siguiente sean igual al cambio de estados S_i, S_j dadas las observaciones y el modelo. Se puede crear una relación entre $xi_t(i)$ con la probabilidad de estar en el estado S_i en el tiempo t dadas las observaciones y el modelo:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j).$$

Si se suma $\gamma_t(i)$ sobre el tiempo t, se obtiene una cantidad que se puede interpretar como el número de veces que se ha transitado por el estado S_i y la sumatoria sobre el tiempo t de $\xi_t(i)$ se puede interpretar como el número de transiciones de S_i a S_j. Usando las ecuaciones anteriores [3] se puede establecer un método reestimativo de los parámetros por las fórmulas:

$\bar{\pi}_i = \gamma_1(i)$,

$$\bar{a_{i,j}} = \frac{\sum_{t=1}^{T-1} \xi_i(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)},$$

$$\bar{b_j}(k) = \frac{\sum_{t=1, O_t = V_k}^{T} \gamma_t(j)}{\gamma_t(j)}.$$

Donde se sigue cumpliendo la condición de probabilidad absoluta sobre los elementos de la j-ésima columna de la matriz $\bar{A}$, para todo $\bar{b_j}(k)$ y sobre toda la distribución inicial $\bar{\pi}$.

$$\sum_{j=1}^{N} \bar{a_{i,j}} = 1 \quad 1 \leq i \leq N,$$

$$\sum_{k=1}^{M} \bar{b_j}(k) = 1 \quad 1 \leq j \leq N.$$

Técnicas de multiplicaciones Lagrangianas son usadas para encontrar los valores de los parámetros que maximice la probabilidad de generar una secuencia de observaciones dado el modelo $P = P(O|\lambda)$, la condición de optimización de P por la modificación de $\pi_i, a_{i,j}, b_j(k)$ formalmente se define bajo:

$$\pi_i = \frac{\pi_i \frac{\partial P}{\partial \pi_i}}{\sum_{k=1}^{N} \pi_k \frac{\partial P}{\partial \pi_k}},$$

$$a_{i,j} = \frac{a_{i,j}\frac{\partial P}{\partial a_{i,j}}}{\sum_{k=1}^{N} a_{i,k}\frac{\partial P}{\partial a_{i,k}}},$$

$$b_j(k) = \frac{b_j(k)\frac{\partial P}{\partial b_j(k)}}{\sum_{l=1}^{M} b_j(l)\frac{\partial P}{\partial b_j(l)}}.$$

Con lo que se puede inferir que el problema 3 es sobre la naturaleza de la optimización de los parámetros, donde técnicas de búsquedas por gradiente para maximizar P convergen a máximos locales arrojando resultados comparables a los procedimientos de reestimación.

4. Sistema de RAL y RAH propuesto

El esquema de reconocimiento del presente trabajo se basa en la creación de pares de modelos Λ_H y λ_L, la primer búsqueda dónde se maximice la probabilidad máx $P(O|\lambda_H)$ arroja el resultado del reconocimiento del habla y por otro lado máx $P(O|\lambda_L)$ asigna que locutor produjo la muestra. Se creó un corpus (base de datos) con el contenido como lo muestra la tabla **4-1**, donde se usa el 70 % de la información para el entrenamiento de los HMM y el 30 % restante para la prueba de los modelos ya creados y entrenados.El sistema propuesto trabaja bajo el esquema de verificación de locutores y reconocimiento de palabras aisladas. Se reconocen diez palabras y diez locutores por medio de una única muestra que es obtenida con un transductor micrófono a un canal con una tasa de muestreo $T_s = 6.8\text{x}10^{-5} seg$.

Sujeto	Edad	Género	F_0 aprox.(Hz)
1	25	masculino	218.27
2	23	femenino	442.66
3	25	femenino	212.11
4	18	masculino	140.33
5	21	masculino	150.32
6	24	femenino	156.147
7	27	masculino	160.3
8	20	femenino	86.58
9	26	masculino	231.18
10	26	masculino	145.83

Tabla 4-1.: Descripción del corpus empleado.

4.1. Corpus o base de datos

La base de datos es creada a partir de 10 locutores que producen 10 palabras, en este caso; 5 de esas 10 palabras corresponden a comandos genéricos orientados a las necesidades básicas en cualquier tipo de implementación, los cuales son:

- Encender.
- Apagar.

- Información.
- Sensor.
- Actuador.

Las otras 5 palabras restantes son los dígitos del 1 al 5, con lo que se pueden obtener combinaciones de tres comandos maestros (**Encender, Apagar, Información**) por 2 tipos de dispositivos (**Sensor, Actuador**) por hasta 5 dispositivos (dígitos **1-5**). Con una estructura: **comando maestro, tipo de dispositivo, no. de dispositivo** se tiene una posibilidad de 30 comandos disponibles para la manipulación de sistemas.

El corpus o base de datos al ser de menos de 50 productores con la cantidad de repeticiones dada, se puede considerar **pequeña** frente a otras como TIMIT, TI digits, ATISOt, etc. Así las bases cuentan con las condiciones de adquisición y la cantidad suficiente de grabaciones para alcanzar cierta precisión en los parámetros de los modelos [13, pp.: 74-84]. Una descripción con las partes más destacadas del corpus se puede observar en la tabla **4-2**. La

Especificación	Descripción
Ambiente	Estudio, SNR= -3.8755dB
Locutores	10
Comandos	10
Vocabulario	Natural
Fs	44.1 kHz
Canales	Monofónico
Transductor	Micrófono omnidireccional
Horas	0.9
Información	590Mb
Unidades	
Base	Palabras aisladas

Tabla 4-2.: Aspectos destacados del corpus.

base datos es una única grabación que es segmentada y submuestreada posteriormente, cada locutor produce una palabra repetidamente y luego pasa a la siguiente palabra (todos los locutores producen las mismas palabras con las mismas repeticiones). Todos los productores del corpus pertenecen a una misma región geográfica, con lo que se puede asumir que comparten un mismo acento lingüístico y generan las muestras del modo más natural posible.

4.2. Creación de los HMM empleados

La primer etapa en la creación de los modelos λ_H y λ_L consta en extraer la información acústica *compactada* para cada muestra de los comandos a reconocer. Una vez obtenidos los múltiples vectores (coeficientes MFCC para λ_H y MDLF para λ_L) se crea un modelo base con ellos. Los MFCC son matrices almacenadas como datos tipos *celdas* y no como arreglos en tres dimensiones debido a la dinamicidad entre audios muestra (cada muestra tiene un tamaño diferente) y aunque existen métodos como el alineamiento dinámico temporal que fuerzan a las muestras a coincidir en longitud, los algoritmos suelen ser lentos conforme aumenta la frecuencia de muestreo y la longitud temporal entre los estados de los HMM. Es importante remarcar que el uso del alineamiento dinámico temporal puede mejorar la tasa de reconocimiento, aunque en aplicaciones en tiempo real es necesario contemplar un algoritmo eficiente que no repercuta en tiempo de procesamiento.

El primer algoritmo aplicado (algoritmo **4.1**) en la creación de los modelos λ_H es para la exacción de las características MFCC. Dicho algoritmo es utilizado en múltiples etapas del sistema de reconocimiento del habla. Básicamente es utilizado en las etapas de: creación de λ_H, entrenamiento de λ_H y prueba del conjunto λ_H. Por lo que es conveniente presentarlo como una parte independiente en el proceso. A continuación se expone el algoritmo antes mencionado.

Algoritmo 4.1: Extracción de los $MFCC$

1.-**Definición de los parámetros**
2.-**Definición de la dirección del corpus**
3.-**Algoritmo de selección por actividad de voz, if** $E_{\log,v} > 0.2E_{\log,max}$
4.-**Aplicación del Filtro pasa voz** $y' = y(n) * H_1(Z)$
5.-**Aplicación del Filtro de preénfasis** $y''(n) = y'(n) * H_2(Z)$
6.-**Segmentación de** y'' en lapsos de $f_s V_l$ para cada muestra
7.-**Aplicar** $Y_v(k) = |F\{y''_v(n)\}|^2 \quad k = 1, 2, ..., 2^x$
8.-**Agrupar la energía en las bandas** $E_v(m) = \sum M(k,m) Y_v(k)^2 \quad 9 = 1, 2, .., M \quad 18 \leq M \leq 26$
9.-**Operar** $MFCC_v = DCT\{E_v(m)\}$
10.-**Despreciar los valores menos significativos** $MFCC_v(m) \quad m = 1, 2, ..., 12$
11.-**Almacenar los coeficientes**

Algoritmo 4.2: Extracción de los $MDLF$

1.-**Definición de los parámetros**
2.-**Definición de la dirección del corpus**
3.-**Algoritmo de selección por actividad de voz, if** $E_{\log,v} > 0.2E_{\log,max}$
4.-**Aplicación del filtro pasa voz** $y' = y(n) * H_1(Z)$
5.-**Aplicación del filtro de preénfasis** $y''(n) = y'(n) * H_2(Z)$
6.-**Segmentación de** y'' en lapsos de $f_s V_l$ para cada muestra
7.-**Aplicar** $Y_v(k) = |F\{y''_v(n)\}|^2 \quad k = 1, 2, ..., 2^x$
8.-**Agrupar la energía en las bandas** $E_v(m) = \sum M(k,m)Y_v(k)^2 \quad m = 1, 2, .., M \quad 18 \leq M \leq 26$
9.-**Aplicar regresión lineal** $E_{v_R} = \Sigma_{r=1}^{3} E(0°, 45°, 135°, 180°)$
10.-**Operar** $MDLF_v = DCT\{E_{v_R}(m)\}$
11.-**Despreciar los valores menos significativos** $MDLF_v(m) \quad m = 1, 2, ..., 12$
11.-**Almacenar los coeficientes**

4.2.1. Valores de los parámetros para los MFCC y MDLF

La variación en los parámetros con los que son obtenidos los coeficientes repercute directamente en el desempeño del sistema de reconocimiento, y como se puede observar en la sección de resultados; la tasa de reconocimiento en palabras aisladas para una base de datos pequeña se mejora significativamente aumentando la duración en las ventanas de análisis, aunque esto implica que pierda sensibilidad y presente ambigüedad ante las palabras con sonidos similares como:"/a/ /c/ /t/ /u/ /a/ /d/ /o/ /r/" y "/a/ /p/ /a/ /g/ /a/ /r/". Es basta la cantidad de variables que influyen en el proceso de extracción de características aunque realmente son pocas las que tienen un impacto relevante en la tasa de reconocimiento en sistemas similares al propuesto en este documento. Dejando de lado la frecuencia de muestreo y la etapa de preprocesamiento, algunos de los parámetros que tienen más impacto la variación de la precisión del reconocimiento son: el número de filtros en el banco, la banda del banco de filtros, la longitud temporal de las ventanas de análisis, el traslape y la discriminación del coeficiente cero (coeficiente con mayor información de la media de la señal).

4.2.2. Iniciación de λ_H

Los HMM son creados por medio de la herramienta **HMM toolkit** © de Kevin Murphey y distribuida por **MIT license, Open Source Iniciative** [19] usada bajo **GNU Octave v.40** © [20] en un sistema operativo **Linux** basado en debian. La creación para cada modelo se basa en el algoritmo de EM el cual se alimenta por múltiples vectores de observaciones de coeficientes MFCC y MDLF para cada muestra de audio asociada al modelo presuntamente resultante. Posteriormente se evalúan los vectores muestra contra los modelos haciendo uso del algoritmo de Viterbi.

Se ha mencionado en secciones pasadas que el corpus de entrenamiento es el 70 % del corpus total, de modo que son usadas las primeras siete repeticiones por comando para cada orador. Un elemento importante a considerar es la regla de la transición de estados que se aplica en reconocimiento de voz regida por la topología de izquierda a derecha, por lo que una matriz inicial de transición de estados A_0 se puede plantear como:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0 \\ 0 & a_{2,2} & a_{2,3} & a_{2,4} \\ 0 & 0 & a_{3,3} & a_{3,4} \\ 0 & 0 & 0 & a_{4,4} \end{bmatrix} \qquad \sum_{j=1}^{N} a_{i,j} = 1$$

La probabilidad de iniciar en el estado i es a su vez útil, considerando que en este tipo de aplicaciones es posible establecer la regla de iniciar las transiciones de los estados a partir del estado 1, esto es: $\pi_0 = [1, 0, 0, 0]$ siguiendo con la topología del ejemplo anterior donde el número de estados es igual a 4. Una vez definidos estos arreglos iniciales podemos crear un modelo $\lambda^c_{H_0}$ prototipo para el comando c haciendo uso del algoritmo **4.2** el cual se muestra a continuación.

Algoritmo 4.3.1: Creación de λ^c_H

1.-**Seleccionar las muestras de la base para el comando** c **y la definición** λ_H
2.-**for** $e = 1\,\% : 70\,\%$ **de** c
$c(e) \rightarrow$ **algoritmo 4.1** $= f_{mfcc}(e)$
almacenar $f_{mfcc}(e)$ **en una estructura tipo celda** $F_H\{e\} = f_{mfcc}(e)$
end
3.-**Inicializar los parámetros** μ_0, σ_0, mix_0 **dados** λ_H**,**A_0**,**π_0 y F_{mfcc}
4.-**Comenzar el algoritmo de Baum-Welch**
5.-**Almacenar el modelo como las matrices** $A, B, \mu, \sigma, M_{mix}$ **para** λ^c_H

Algoritmo 4.3.2: Creación de $\lambda^s_{L_0}$

1.-**Seleccionar las muestras de la base para el sujeto** s
2.-**for** $e = 1\,\% : 70\,\%$ **de** s
$s(e) \rightarrow$ **algoritmo 4.2** $= f_{mdlf}(e)$
almacenar $f_{mdlf}(e)$ **en una celda** $F_{mdlf}\{e\} = f_{mdlf}(e)$
end
3.-**Iniciar los parámetros** μ_0, σ_0 **dados** A_0**,** π_0 y F_{mdlf}
4.-**Comenzar el algoritmo de Baum-Welch**
5.-**Almacenar el modelo como las matrices** $A, B, \mu, \sigma, M_{mix}$ **para** λ^s_L

Los valores iniciales μ_0 y σ_0 obtenidos con el algoritmo **4.2** gracias a la definición de A_0 y π_0 y a la secuencia de observaciones $MFCC$, son utilizados en algoritmo de maximización de expectativas o EM por sus siglas en inglés (expectation maximization) para crear los modelos finales λ_H y λ_C.

4.2.3. Estimación de los parámetros de λ^c_H y λ^s_L

Una vez generados los modelos base (prototipo) $\lambda^c_{H_0}$ y $\lambda^s_{L_0}$ dadas las observaciones y definida la topología de los modelos, se continua a estimar los parámetros A, π, B, μ, σ que son en si la serie de matrices y vectores que definen λ^c_H y λ^s_L respectivamente; haciendo uso del algoritmo de Baum-Welch (algoritmo de EM), los elementos son modificados iterativamente hasta que convergen cerca de los valores óptimos. Es posible evaluar una curva de *aprendizaje* para definir la cantidad de iteraciones mínima necesaria I_{max} para alcanzar un nivel de convergencia Cv_{th}, típicamente se usa un valor *threshold* de $1\text{x}10^{-4}$. Con el algoritmo **4.4** es posible obtener la maximización de expectativas por el procedimiento Baum-Welch.

Algoritmo 4.4: Estimación de $A, B, \mu, \sigma, M_{mix}$ **(Baum-Welch)**

1.-**Definir los datos** $A_0, B_0, \mu_0, \sigma_0$ **y la cantidad de iteraciones y la matriz de covarianza de acuerdo al modelo izquierda-derecha**
2.-**Se establece las banderas:**
Probabilidad pasada $Ll_p = -\infty$
Probabilidad actual $Ll_a = 0$
Convergencia $Cv = 0$
Tolerancia de convergencia $Cv_{th} = 1\text{x}10^{-4}$
Iteración $i = I$

3.-**while(**$I \leq I_{max}$)&($Cv_{th} \leq Cv$)
Calcula las *pdf* de acuerdo a los CCs (coeficientes)
Implementa el algoritmo de *fordward-backward*
Suma las probabilidades logarítmicas
Actualiza los valores
Estima la convergencia
4.-**Retorna** $A, B, \pi, \mu, \sigma, M_{mix}$ si hay convergencia

---

4.3. Evaluación de λ_H^c y λ_L^s

.
En esencia esta sección corresponde a una etapa de reconocimiento como tal, evaluando el porcentaje de precisión del conjunto de modelos λ_H^c y λ_L^s previamente creados. Se establece entonces si los modelos se desempeñan a una tasa de reconocimiento mayor al 90 % y de no ser así, sería necesario ajustar los parámetros desde la etapa de preprocesamiento hasta la de creación de λ. La evaluación se lleva a cabo por medio del algoritmo de Viterbi aplicado a los modelos contra muestras del corpus. El proceso consiste en llamar al 30 % de las muestras de la base de datos que no fueron usadas en el algoritmo de creación de los modelos prototipo ni en el algoritmo de maximización de la esperanza y aplicar el algoritmo de Viterbi entre cada muestra contra cada par de modelos λ_H^c y λ_L^s. El par de modelos que arrojen la mayor probabilidad logarítmica (probabilidad resultado del preprocesamiento de los modelos [al cambiar multiplicaciones de probabilidades por suma de probabilidades logarítmicas]). En el algoritmo **4.5** se ilustra el proceso de evaluación donde los pasos 4 y 5 corresponden al algoritmo de Viterbi.

Algoritmo 4.5: Evaluación de λ_H^c y λ_L^c

1.-**Seleccionar las muestras de la base para el sujeto** s**y comando** c
2.-**for** $e = 71\,\% : 100\,\%$ **de** s **y** c
$s(e) \rightarrow$ **algoritmo 4.1** $= f_{mfcc}(e)$
$c(e) \rightarrow$ **algoritmo 4.2** $= f_{mdlf}(e)$
$F_{mfcc}\{e\} = f_{mfcc}(e)$, $F_{mdlf}\{e\} = f_{mdlf}(e)$
end
3.-**Evalúa la pdf de las mezclas gaussianas**
4.-**Aplica el algoritmo *fordward-backward***
5.-**Selecciona el modelo con la mayor prob. logarítmica**

---

5. Resultados y conclusiones

La combinación de los valores en la variación de resultados debido al cambio entre los parámetros de extracción de características y en la definición de los modelos estadísticos da cabida a una amplia variedad de resultados en cuestión a la precisión porcentual del reconocimiento por palabra y locutor. Donde las variaciones con mejores resultados son mostradas a continuación mediante tablas que reflejan el resultado comparativo entre cinco configuraciones en la extracción de las características y cinco variaciones en las definiciones de los modelos estadísticos. La definición de los parámetros en la extracción de los coeficientes se muestra en la tabla **5-1** tanto para los MFCC como para los MFLF ya que como se ha expuesto en capítulos anteriores; el método para su obtención varía solo en la etapa donde se agregan las regresiones para los MDLF y no se agregan los deltas(Δ) ni los deltas-deltas($\Delta - \Delta$) ya que su longitud es fija en 48 elementos (4 regresiones de 12 coeficientes [eliminando de los cálculos el valor promedio o coeficiente $CC(0)$]).

Valores de los parámetros	Wl (ms)	Ol (ms)	NFB	BB (Hz)	α	CC
Configuración 1	20	10	18	0 - 7350	.97	13
Configuración 2	25	10	20	30 - 5000	31/32	$13 + \Delta + \Delta\Delta$
Configuración 3	45	20	23	0 - 4500	31/32	$13 + \Delta + \Delta\Delta$
Configuración 4	100	10	23	100 - 4500	31/32	13
Configuración 5	150	20	23	100 - 4500	31/32	$13 + \Delta + \Delta\Delta$

Tabla 5-1.: Conjunto de variaciones en la extracción de los coeficientes.

Donde acorde con la notación de las variables usadas en los códigos de programación se define como: longitud de ventana de análisis Wl (Window length), traslape Ol (Overlap), número de filtros en el banco NFB, banda del banco de filtros BB, coeficiente de preénfasis α y cantidad de elementos en los vectores de coeficientes CC.

Tres de los cambios más significativos para el aumento en la precisión del reconocimiento, tanto para el RAL como para el RAH; recae en la longitud de ventana de análisis, traslape y el número en el banco de filtros donde se obtiene un incremento de dos a tres puntos porcentuales. En la figura **5-1** se refleja cómo afecta al reconocimiento la variación en el número de filtros, alcanzando su máximo para 23 filtros al igual que lo reporta Han et al. en

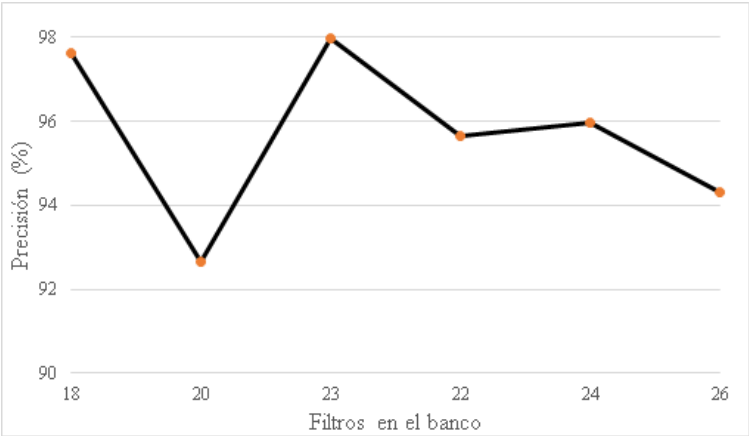

Figura 5-1.: Precisión de reconocimiento usando diferente número de filtros triangulares en el banco.

su publicación del 2006, presentado un modo eficiente en la extracción de los MFCC [21].

Las combinaciones en la definición de los HMM se escogieron de acuerdo con lo que presenta la literatura ([3, 22, 21]) y a los mejores resultados obtenidos en la experimentación. Las tablas **5-2** y **5-3** exponen la cantidad de estados usados y las mezclas gaussianas por estado evaluadas con las cinco configuraciones expresadas en la tabla **5-1**; mostrando un resultado en la precisión del reconocimiento general para RAL y para RAH.

HMM		Precisión general (%)				
Estados (Q)	Mezclas (mix)	Config. 1	Config. 2	Config. 3	Config. 4	Config. 5
1	2	95.95	96.64	**97.32**	97.31	92.63
2	3	**99.66**	98.99	99.326	98.99	97.983
3	1	**99.66**	90.31	94.31	94.313	95.66
5	3	99.66	100	**100**	98.66	99.66
8	2	99.66	100	**100**	98.66	97.99

Tabla 5-2.: Precisión general por modelo λ_H para todas las configuraciones definidas.

Para $\lambda_{L,1} : Q = 1, \text{mix} = 2$,$\lambda_{L,2} : Q = 2, \text{mix} = 3$,$\lambda_{L,3} : Q = 3, \text{mix} = 1$,$\lambda_{L,4} : Q = 5, \text{mix} = 3$,$\lambda_{L,5} : Q = 8, \text{mix} = 2$. Como se puede apreciar en la tabla anterior la configuración tres, muestra una mayor tasa de reconocimiento, seguida de la configuración uno para el caso de RAH en todos los modelos ($\lambda_{H,1} \cdot \lambda_{H,5}$). En el caso del RAL de nuevo la configuración tres es la que tiene un mayor índice de reconocimiento, seguido de la configuración cinco de igual manera para las cinco definiciones de λ_L.

HMM		Precisión general (%)				
Estados (Q)	Mezclas	Congfig. 1	Config. 2	Config. 3	Config. 4	Config. 5
1	2	97.98	99.32	97.98	99.32	**99.66**
2	3	99.32	99.33	**100**	99.32	99.66
3	1	94.98	88.32	**99.66**	98.32	96.64
5	3	100	100	**100**	99.66	100
8	2	100	100	**100**	99.66	100

Tabla 5-3.: Precisión general por modelo λ_L para todas las configuraciones definidas.

Los comandos a reconocer por su contenido fonético presentan un cierto error que es susceptible a la variación de longitud de ventana más que a la variación en la definición de los modelos. Como se puede observar en la tabla **5-4** los comandos "apagar" y "actuador" presentan la mayor cantidad de error acumulado porcentual, aunque en longitudes largas en las ventanas de análisis (100 ms - 150 ms) con poco traslape (10 ms - 20 ms) el error cae de modo notorio en comparación a la longitud intermedia en las ventanas de 45 ms. Es importante notar como el error porcentual para los casos expuestos va aumentando desde configuración "tradicional" de 25 ms hasta una de 45 ms y cómo vuelve a disminuir con una pendiente pronunciada hasta ventanas de análisis "largas" con poco traslape.

Comando	Error(%)					EA(%)
	$\lambda_{H,1}$	$\lambda_{H,2}$	$\lambda_{H,3}$	$\lambda_{H,4}$	$\lambda_{H,5}$	
Encender	1.34	1.34	4.68	0	0	7.36
Apagar	4.02	**2.008**	**12.68**	0.68	**2.02**	**21.408**
Sensor	2.04	0	2	0	0	4.04
Actuador	**10.04**	**2.014**	**14.014**	0	0	**26.068**
Información	3.34	0	0	0	0	3.34
Uno	2.72	1.36	8.72	**2.68**	2	17.48
Dos	6.06	0.68	5.36	0	0.68	12.78
Tres	**7.36**	2.7	2.7	0	**2.68**	15.44
Cuatro	2.02	0	1.34	0.68	0	4.04
Cinco	1.36	0	0	0	0	1.36

Tabla 5-4.: Error porcentual promedio y error porcentual acumulado promedio para cada configuración dado cada modelo λ_H.

En la tabla **5-5** se muestra una de las partes más importantes del presente trabajo, puesto que se exponen una serie de resultados de hasta el cien por ciento en la precisión del reconocimiento para los sujetos nueve y diez y un error acumulado promedio porcentual bajo y hasta de cero para las configuraciones de ventanas largas (como se expone en las tablas

específicas expuestas en el apéndice), con los modelos $\lambda_{L,3}$, $\lambda_{L,4}$ y $\lambda_{L,5}$.

Sujeto	Error (%)					EA(%)
	$\lambda_{L,1}$	$\lambda_{L,2}$	$\lambda_{L,3}$	$\lambda_{L,4}$	$\lambda_{L,5}$	
1	2.02	0	0	0	0	2.02
2	**3.36**	0	0	0	0	3.36
3	1.34	**2.7**	**0.68**	**0.68**	**0.68**	**6.08**
4	0	0	0	0	0	0
5	2.72	1.36	0	0	0	**4.08**
6	0	0	0	0	0	0
7	1.36	0	0	0	0	1.36
8	0.68	0.68	0	0	0	1.36
9	0	0	0	0	0	0
10	0	0	0	0	0	0

Tabla 5-5.: Error porcentual promedio y error porcentual acumulado promedio para cada configuración dado cada modelo λ_L.

En la figura **5-2** se muestra como el aumento en la longitud de las ventanas de análisis manteniendo un traslape corto ayuda a definir las frecuencias fundamentales al agrupar las frecuencias que se dispersan alrededor de estas. Esto se atribuye a la reducción del efecto de la incertidumbre. Al aumentar la longitud en las ventanas de análisis se conoce más del comportamiento de la frecuencia y menos del comportamiento de la señal en el tiempo, ya que al aumentar el tiempo de análisis es posible dar un margen de estudio de las frecuencias más altas.

Como se puede corroborar en los resultados obtenidos, es posible observar como el sistema de reconocimiento superó con creces a la expectativa planteada en los objetivos donde se esperaba un resultado cercano al **90 %**, cuando el promedio de la precisión para todas las configuraciones y definiciones de los modelos fue de **98,24 %**, y en el caso del RAH por separado; se consiguió un reconocimiento de hasta el **100 %** para las configuraciones tres y cuatro con los modelos $\lambda_{H,4}$ y $\lambda_{H,5}$ y en el caso del RAL se consiguió un **100 %** para las configuraciones dos y tres con los modelos $\lambda_{L,4}$ y $\lambda_{L,5}$. El reconocimiento dual máximo fue a su vez del **100 %** para la configuración tres para las definiciones de modelos cuatro y cinco, con lo que se puede concluir que los mejores modelos fueron los que tienen entre cinco y ocho estados con las configuraciones convencionales para el reconocimiento automático de voz.

Una de las conclusiones más notorias, es que al tener una longitud de análisis más amplia, se pueden estudiar mejor las frecuencias que oscilan fuera de un rango temporal insuficiente,

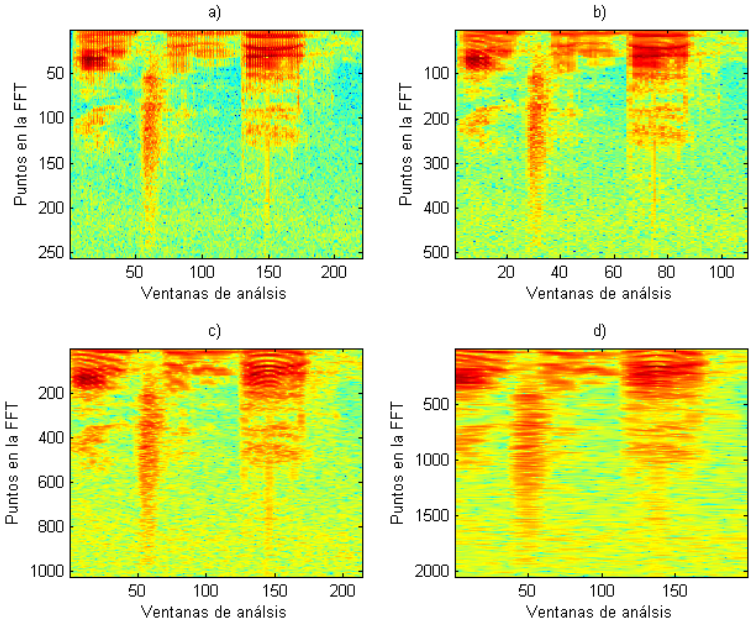

Figura 5-2.: a) Espectrograma con un Wl = 25 ms. b) Wl = 45 ms. c) Wl = 100 ms. d) Wl = 250 ms.

aunque al sobrescribir la información temporal por medio del traslape se sesga la incertidumbre, y que para estas configuraciones, funcionan mejor los modelos con pocas mezclas por estado. El contraefecto es que al realizar el enventanado dadas las formas de las funciones ventana, la información es atenuada por la parte ascendente; ya que las ventanas tienen una forma gaussiana, generando información distorsionada en magnitud de forma no lineal sobre todo el análisis.

Uno de los factores más perjudiciales al momento de realizar el reconocimiento es el ruido ambiental o de fondo. Ya que el sonido en el entorno es extremadamente variable sobre toda la banda de análisis de voz, con lo que una de las propuestas para los trabajos futuros es la aplicación de filtros adaptativos en conjunto con filtros estáticos. Otra propuesta para mitigar el efecto del ruido, es crear una base de datos contaminada con un ruido ambiental controlado con lo que se supondría que los modelos ocultos de Markov serían más robustos al ruido.

Se esperaría que el próximo avance en el reconocimiento dual se dé por medio de la aplicación de redes neuronales profundas, con lo que en dicho caso, habría que usar una base de datos mucho mayor a la usada en la presente investigación. Eso aunado a una exigencia de cómputo mucho mayor con lo que sería necesario el uso de GPU's. Finalmente, dado el resultado que se obtuvo en el RAL, se sugiere que se oriente una investigación entorno a la rama biométrica más a fondo.

A. Tablas específicas

A.1. Resultados de la precisión de reconocimiento para λ_H

Las tablas que son presentadas a continuación describen la precisión del reconocimiento por comando de un modo más desglosado para cada configuración (ver tabla **5-1**) y para cada definición de modelo λ_H.

Comando	Precisión porcentual$\lambda_{H,1}(\,\%)$				
	Config. 1	Config. 2	Config. 3	Config. 4	Config. 5
Encender	93.3	100	100	100	100
Apagar	96.6	100	90	100	93.3
Sensor	96.6	96.6	100	100	96.6
Actuador	93.3	93.3	100	86.6	76.6
Información	100	93.3	100	100	90
Uno	96.6	96.6	96.6	100	96.6
Dos	96.6	93.3	96.6	96.6	86.6
Tres	96.6	93.3	90	93.3	90
Cuatro	93.3	100	100	96.6	100
Cinco	96.6	100	100	100	96.6

Tabla A-1.: Precisión de reconocimiento porcentual para $\lambda_{H,1}(Q = 1, mix = 2)$.

Comando	Precisión porcentual$\lambda_{H,2}(\%)$				
	Config. 1	Config. 2	Config. 3	Config. 4	Config. 5
Encender	100	100	100	93.3	100
Apagar	100	93.3	96.66	100	100
Sensor	100	100	100	100	100
Actuador	100	100	100	96.6	93.33
Información	100	100	100	100	100
Uno	96.6	100	100	100	96.6
Dos	100	100	100	100	96.6
Tres	100	96.6	96.6	100	93.3
Cuatro	100	100	100	100	100
Cinco	100	100	100	100	100

Tabla A-2.: Precisión de reconocimiento porcentual para $\lambda_{H,2}(Q = 2, mix = 3)$.

Comando	Precisión porcentual$\lambda_{H,3}(\%)$				
	Config. 1	Config. 2	Config. 3	Config. 4	Config. 5
Encender	100	76.6	100	100	100
Apagar	100	83.3	80	80	93.3
Sensor	100	100	100	100	90
Actuador	100	63.3	83.3	83.33	100
Información	100	100	100	100	100
Uno	96.6	86.6	96.6	96.6	80
Dos	100	100	86.6	86.6	100
Tres	100	93.3	96.6	96.6	100
Cuatro	100	100	100	100	93.3
Cinco	100	100	100	100	100

Tabla A-3.: Precisión de reconocimiento porcentual para $\lambda_{H,3}(Q = 3, mix = 1)$.

Comando	Precisión porcentual$\lambda_{H,4}(\%)$				
	Config. 1	Config. 2	Config. 3	Config. 4	Config. 5
Encender	100	100	100	100	100
Apagar	96.6	100	100	100	100
Sensor	100	100	100	100	100
Actuador	100	100	100	100	100
Información	100	100	100	100	100
Uno	100	100	100	90	96.6
Dos	100	100	100	100	100
Tres	100	100	100	100	100
Cuatro	100	100	100	96.6	100
Cinco	100	100	100	100	100

Tabla A-4.: Precisión de reconocimiento porcentual para $\lambda_{H,4}(Q = 5, mix = 3)$.

Comando	Precisión porcentual$\lambda_{H,5}(\%)$				
	Config. 1	Config. 2	Config. 3	Config. 4	Config. 5
Encender	100	100	100	100	100
Apagar	96.6	100	100	93.3	100
Sensor	100	100	100	100	100
Actuador	100	100	100	100	100
Información	100	100	100	100	100
Uno	100	100	100	100	90
Dos	100	100	100	100	96.6
Tres	100	100	100	93.3	93.3
Cuatro	100	100	100	100	100
Cinco	100	100	100	100	100

Tabla A-5.: Precisión de reconocimiento porcentual para $\lambda_{H,5}(Q = 8, mix = 2)$.

A.2. Resultados de la precisión de reconocimiento para λ_L

Las tablas que se presentan a continuación contienen la precisión del RAL por cada sujeto usando las configuraciones en la extracción de características y las mismas definiciones usadas en las tables anteriores de los modelos ocultos de Markov.

Sujeto	Precisión porcentual $\lambda_{L,1}$ (%)				
	Config.1	Config.2	Config.3	Config.4	Config.5
1	93,3	100	100	96,6	100
2	93,3	96,6	93,3	100	100
3	100	100	93,3	100	100
4	100	100	100	100	100
5	100	96,6	96,6	96,6	96,6
6	100	100	100	100	100
7	96,6	100	96,6	100	100
8	96,6	100	100	100	100
9	100	100	100	100	100
10	100	100	100	100	100

Tabla A-6.: Precisión de reconocimiento porcentual para $\lambda_{H,1}(Q = 2, mix = 1)$

Sujeto	Precisión porcentual $\lambda_{L,2}$ (%)				
	Config.1	Config.2	Config.3	Config.4	Config.5
1	93,3	100	100	96,6	100
2	93,3	96,6	93,3	100	100
3	100	100	93,3	100	100
4	100	100	100	100	100
5	100	96,6	96,6	96,6	96,6
6	100	100	100	100	100
7	96,6	100	96,6	100	100
8	96,6	100	100	100	100
9	100	100	100	100	100
10	100	100	100	100	100

Tabla A-7.: Precisión de reconocimiento porcentual para $\lambda_{H,1}(Q = 2, mix = 3)$.

Sujeto	Precisión porcentual $\lambda_{L,3}$ (%)				
	Config.1	Config.2	Config.3	Config.4	Config.5
1	100	100	100	100	100
2	100	100	100	100	100
3	96.6	93.3	100	96.6	100
4	100	100	100	100	100
5	100	100	100	96.6	96.6
6	100	100	100	100	100
7	100	100	100	100	100
8	96.6	100	100	100	100
9	100	100	100	100	100
10	100	100	100	100	100

Tabla A-8.: Precisión de reconocimiento porcentual para $\lambda_{H,1}(Q = 3, mix = 1)$.

Sujeto	Precisión porcentual $\lambda_{L,4}$ (%)				
	Config.1	Config.2	Config.3	Config.4	Config.5
1	100	100	100	100	100
2	100	100	100	100	100
3	100	100	100	96.6	100
4	100	100	100	100	100
5	100	100	100	100	100
6	100	100	100	100	100
7	100	100	100	100	100
8	100	100	100	100	100
9	100	100	100	100	100
10	100	100	100	100	100

Tabla A-9.: Precisión de reconocimiento porcentual para $\lambda_{H,4}(Q = 5, mix = 3)$.

Sujeto	Precisión porcentual $\lambda_{L,5}$ (%)				
	Config.1	Config.2	Config.3	Config.4	Config.5
1	100	100	100	100	100
2	100	100	100	100	100
3	100	100	100	96.6	100
4	100	100	100	100	100
5	100	100	100	100	100
6	100	100	100	100	100
7	100	100	100	100	100
8	100	100	100	100	100
9	100	100	100	100	100
10	100	100	100	100	100

Tabla A-10.: Precisión de reconocimiento porcentual para $\lambda_{H,4}(Q = 8, mix = 2)$.

B. Códigos de programación

Definición de las configuraciones en la extracción de los coeficientes

```
clc, clear all, close all;
%% Definición de config. 1
conf.Wl = 20;     % Windos length
conf.Ol = 10;     % Overlap
conf.NFB = 18;    % Número de banco de filtros
conf.BBN = [0,7250]; % Banda del banco de filtros
conf.Window = @hanning;   % Función Ventana
conf.Alpha = 0.97;    % Coeficiente de pre-énfasis
conf.Nc = 14;  % Cantidad de coeficientes a entrenar
conf.Deltas = false;  % Incluir Deltas
save Config1.mat conf
%% Definición de config. 2
conf.Wl = 25;
conf.Ol = 10;
conf.NFB = 20;
conf.BBN = [30,5000];
conf.Window = @hanning;
conf.Alpha = 31/32;
conf.Nc = 14;
conf.Deltas = true;
save Config2.mat conf
%% Definición de config. 3
conf.Wl = 45;
conf.Ol = 20;
conf.NFB = 23;
conf.BBN = [0,4500];
conf.Window = @hanning;
conf.Alpha = 31/32;
conf.Nc = 14;
conf.Deltas = true;
save Config3.mat conf
%% Definicción de config. 4
conf.Wl = 100;
conf.Ol = 10;
conf.NFB = 23;
conf.BBN = [100,4500];
conf.Window = @hanning;
```

```
conf.Alpha = 0.97;
conf.Nc = 14;
conf.Deltas = false;
save Config4.mat conf
%% Definicción de config. 5
conf.Wl = 150;
conf.Ol = 20;
conf.NFB = 23;
conf.BBN = [100,4500];
conf.Window = @hanning;
conf.Alpha = 31/32;
conf.Nc = 14;
conf.Deltas = true;
```

Extracción de los coeficientes MDLF y MFCC

```
% Función para la extracción de los MFCC y los MDLF
% Li = límite inferior
% Ls = límite superior
% Config = 'configX.mat'   configración de los parámetros
% Features = 'MFCC' o 'MDLF'
function [MFCCs,Deltas] = extractFeatures(Li,Ls,Config,Features)
%% Directorios
Dir = 'C:\Tesis\corpus\fuente\fuente_14700\corpus_relleno\';
locutor = {'P_';'A_G_';'A_C_';'Aldo_';'Eduardo_';...
  'Fatima_';'Francisco_';'Naomi_';'Angel_';'Luis_'};
comando = {'encender_';'apagar_';'sensor_';'actuador_';'informacion_';...
  'uno_';'dos_';'tres_';'cuatro_';'cinco_'};
%% Configruación de los Coeficientes usados
load (Config)
Wl = conf.Wl;Ol = conf.Ol; NFB = conf.NFB; BBN = conf.BBN;
Window = conf.Window; Alpha = conf.Alpha; Nc = conf.Nc;
Deltas = conf.Deltas;x = 0;

if strcmp(Features,'MFCC')
%% Extracción de MFCC
for i  = 1:size(locutor,1)  % desde 1 hasta todos los comandos
    for j = 1:size(comando,1)    % desde 1 hasta todos los locutores
        for k = Li:Ls          % desde limite inferior hasta limite sup
        x = x+1;
        archivo = strcat(Dir,locutor{i},comando{j},int2str(k),'.wav');
        [audio_es,fs] = wavread(archivo);
        audio_mon = audio_es(:,1);
        audio_norm = audio_mon/max(audio_mon); % normalización del audio
        audio = actividad(audio_norm,fs,30);    % selector de actividad
        MFCC = mfcc(audio,fs,Wl,Ol,Alpha,Window,BBN,NFB,Nc,Nc);
        if Deltas; Fuente{x} = [MFCC;diff(MFCC)/2;diff(diff(MFCC))/2];...
        else Fuente{x} = MFCC; end;
```

```
        end
    end
    MFCCs.(locutor{i}) = Fuente;
    Fuente ={};
    x = 0;
end
else
%% Extracción de MDLF
for i  = 1:size(locutor,1)  % desde 1 hasta todos los comandos
    for j = 1:size(comando,1)    % desde 1 hasta todos los locutores
        for k = Li:Ls          % desde limite inferior hasta limite sup
        x = x+1;
        archivo = strcat(Dir,locutor{i},comando{j},int2str(k),'.wav');
        [audio_es,fs] = wavread(archivo);
        audio_mon = audio_es(:,1);
        audio_norm = audio_mon/max(audio_mon); % normalización del audio
        audio = actividad(audio_norm,fs,30);    % selector de actividad
        MDLF = mdlf(audio,fs,Wl,Ol,Alpha,Window,BBN,NFB,Nc,Nc);
        Fuente{x} = MDLF;
        end
    end
    MDLFs.(locutor{i}) = Fuente;
    Fuente ={};
    x = 0;
    Deltas = 0;
    save MDLFs.mat MDLFs
end
end
```

Creación de λ_H

```
clc, close all, clear all;
warning off  %#ok<*WNOFF>

%% Indexado
comando = {'encender_';'apagar_';'sensor_';'actuador_';'informacion_';...
  'uno_';'dos_';'tres_';'cuatro_';'cinco_'};

%% Extracción de los MFCC
Li = 1; Ls = 7;
[MFCCs,Deltas] = extractFeatures(Li,Ls,'config6.mat','MFCC');

%% Definición de los HMM
Q = 2;  % número de estados
O = size(MFCCs.encender_{1},1);  % número de coeficientes por ventana
mix = 2;  % cantidad de mezclas Gaussianas por estado
for i = 1: size(comando,1)
    Fuente = MFCCs.(comando {i});
```

```
    [LL, prior2, transmat2, mu2, Sigma2, mixmat2] = ...
                                    Entrenamiento(Fuente,Q,O,mix);
    eval(['save modelos/comandos_7/' comando{i} ...
            ' LL prior2 transmat2 mu2 Sigma2 mixmat2']);
end
```

Creación de λ_L

```
clc, close all, clear all;
warning off  %#ok<*WNOFF>

%% Indexado
 locutor = {'P_';'A_G_';'A_C_';'Aldo_';'Eduardo_';...
  'Fatima_';'Francisco_';'Naomi_';'Angel_';'Luis_'};
%% Extracción de los MDLF
Li = 1; Ls = 7;
 [MDLFs,Deltas] = extractFeatures(Li,Ls,'config3.mat','MDLF');
%% Definición de los HMM
Q = 8; % número de estados
O = size(MDLFs.P_{1},1); % número de coeficientes por ventana
mix = 2; % cantidad de mezclas Gaussianas por estado
for i = 1: size(locutor,1)
    Fuente = MDLFs.(locutor {i});
    [LL, prior2, transmat2, mu2, Sigma2, mixmat2] = ...
                                    Entrenamiento(Fuente,Q,O,mix);
    eval(['save modelos/locutores_md/' locutor{i} ...
            ' LL prior2 transmat2 mu2 Sigma2 mixmat2']);
end
```

Entrenamiento, subfunción en la creación de λ_H y λ_L

```
% Creación de los modelos por la topología izquierda–derecha
% Q = número de estados
% O = número de observaciones
% mix = número de Mezclas
function [LL, prior2, transmat2, mu2, Sigma2, mixmat2] = ...
                                                Entrenamiento(Coef,Q,O,mix)
%addpath(genpath('HMMall'));  % directorios de HMM toolkit (K. Murphey)
cov_type = 'full'; %the covariance type that is chosen as ?ull?for gaussians.
prior1 = zeros(Q,1);
prior1(1) = 1;
transmat1 = mk_stochastic(triu(rand(Q,Q)));
temp = cell2mat(Coef);
[mu1, Sigma1] = mixgauss_init(Q*mix, temp, cov_type);
mu1 = reshape(mu1, [O Q mix]);
Sigma1 = reshape(Sigma1, [O O Q mix]);
mixmat1 = mk_stochastic(rand(Q,mix));
```

```
[LL, prior2, transmat2, mu2, Sigma2, mixmat2] =....
mhmm_em(Coef, prior1, transmat1, mu1, Sigma1, mixmat1, 'max_iter', 100);
end
```

Para las subfunciones dentro de la serie códigos de Kevin Murphey [19].

Evaluación de λ_H

```
clc, close all, clear all
warning off
%% Directorio
comando = {'encender_';'apagar_';'sensor_';'actuador_';'informacion_';...
  'uno_';'dos_';'tres_';'cuatro_';'cinco_'};
%% Extracción de los MFCC
Li = 8; Ls = 10;
[MFCCs,Deltas] = extractFeatures(Li,Ls,'config4.mat','MFCC');
%% Contadores
acierto = 0; error = 0; ViterRes = zeros(1,10);
x = (Ls-Li+1)*size(comando,1);
%% Evaluación
for l = 1:size(comando,1)
    for m = 1:length(MFCCs.(comando{l}))
        for n = 1:size(comando,1)
          eval(['load modelos/comandos_7/' comando{n}]);
          ViterRes(n) = mhmm_logprob(MFCCs.(comando{l})(m),prior2, transmat2,...
                              mu2, Sigma2, mixmat2);
        end
        [tmp,Recog] = max(ViterRes);
        if Recog == l; acierto = acierto + 1; end;
    end
porcentaje = (acierto / x)*100;
fprintf('***--Porcentaje de reconocimiento= %d --***\n',porcentaje);
acierto = 0;
end
```

Evaluación de λ_L

```
clc, close all, clear all
warning off
%% Directorio
Dir = 'C:\Tesis\corpus\fuente\fuente_14700\corpus_relleno\';
locutor = {'P_';'A_G_';'A_C_';'Aldo_';'Eduardo_';...
  'Fatima_';'Francisco_';'Naomi_';'Angel_';'Luis_'};

%% Extracción de los MFCC
Li = 8; Ls = 10;
[MDLFs,Deltas] = extractFeatures(Li,Ls,'config3.mat','MDLF');
```

```
%% Contadores
acierto = 0; error = 0; ViterRes = zeros(1,10);
x = (Ls-Li+1)*size(locutor,1);
%% Evaluación
for l = 1:size(locutor,1)
    for m = 1:length(MDLFs.(locutor{l}))
        for n = 1:size(locutor,1)
          eval(['load modelos/locutores_md/' locutor{n}]);
          ViterRes(n) = mhmm_logprob(MDLFs.(locutor{l})(m),prior2, transmat2,...
                              mu2, Sigma2, mixmat2);
        end
        [tmp,Recog] = max(ViterRes);
        if Recog == l; acierto = acierto + 1; end;
    end
porcentaje = (acierto / x)*100;
fprintf('***--Porcentaje de reconocimiento=%d --***\n',porcentaje);
acierto = 0;
end
```

Detector de actividad de voz

```
% función de detección de
% sign =  waveform de la seal
% V = longitud de analisis de la ventana en mili-segundos

function sign_corte = actividad(signal,fm,v)
%-------------------------------------------------------
V = v/1000;
Vn = ceil(V*fm);
longitud = length(signal);
energia = zeros(1,ceil(longitud/Vn));
limite = length(energia);
sign_corte = [];
%-------------------------------------------------------
j =1;
while (j+1<limite)
    energia(j) = sum(abs(signal(j*Vn:(j+1)*Vn))).^2;
    j = j+1;
end
logenergia = log(energia);
umbral = max(logenergia)*20/100;

for k =0:limite-1
    if logenergia(k+1)>=umbral
        sign_corte = [sign_corte;signal((k*Vn)+1:(k+1)*Vn)];
    end
end
```

Extracción de los MDLF

```
% MDLF multi-directional features extraction.
function [ Dt Df Dtf Dft ] = mdlf( speech, fs, Tw, Ts, alpha, window, R, M, N, L )
    % Ensure correct number of inputs
    if( nargin~= 10 ), help mfcc; return; end;
    if( max(abs(speech))<=1 ), speech = speech * 2^15; end;
    Nw = round( 1E-3*Tw*fs );     % frame duration (samples)
    Ns = round( 1E-3*Ts*fs );     % frame shift (samples)
    coef_reg = 28;                % sum(1^2 + 2^2 + 3^2)
    nfft = 2^nextpow2( Nw );      % length of FFT analysis
    K = nfft/2+1;                 % length of the unique part of the FFT
% Greetings to my friends Pacho el azul, Juanito, Castor, Mike, Marco, Olaf,....
    hz2mel = @( hz )( 1127*log(1+hz/700) );     % Hertz to mel warping function
    mel2hz = @( mel )( 700*exp(mel/1127)-700 ); % mel to Hertz warping function
    dctm = @( N, M )( sqrt(2.0/M) * cos( repmat([0:N-1].',1,M) ...
                                        .* repmat(pi*([1:M]-0.5)/M,N,1) ) );
    ceplifter = @( N, L )( 1+0.5*L*sin(pi*[0:N-1]/L) );
    speech = filter( [1 -alpha], 1, speech ); % fvtool( [1 -alpha], 1 );
    speech = voicefilter(speech);
    frames = vec2frames( speech, Nw, Ns, 'cols', window, false );
    MAG = abs( fft(frames,nfft,1) );
    H = trifbank( M, K, R, fs, hz2mel, mel2hz ); % size of H is M x K
    FBE = H * MAG(1:K,:); % FBE( FBE<1.0 ) = 1.0; % apply mel floor
    DCT = dctm( 12, N -1);
    % log compression
    LogFBE = log(FBE);
    LogFBE = LogFBE(2:M,:);
    [freq time] = size(LogFBE);
    %-------------------------------------------------------------------------
    % LR through time
    for f = 1:freq
      for t = 4:time-3
        Dt(f,t-3) = ((LogFBE(f,t+1) - LogFBE(f,t-1)) ...
        + 2*(LogFBE(f,t+2) - LogFBE(f,t-2))+...
        3*(LogFBE(f,t+3) - LogFBE(f,t-3)))/coef_reg;
      end
    end
    %-------------------------------------------------------------------------
    % LR through frequency
    for t = 1:time
      for f = 4:freq-3
        Df(f-3,t) = ((LogFBE(f+1,t) - LogFBE(f-1,t))...
        + 2*(LogFBE(f+2,t) - LogFBE(f-2,t))+...
        3*(LogFBE(f+3,t) - LogFBE(f-3,t)))/coef_reg;
      end
    end
```

```
    %-------------------------------------------------------------------------
    % LR through time-frequency
    for t = 4:time-3
      for f = 4:freq-3
        Dtf(f-3,t-3) = ((LogFBE(f-1,t-1) - LogFBE(f+1,t+1))...
        + 2*(LogFBE(f-2,t-2)...
        - LogFBE(f+2,t+2)) + 3*(LogFBE(f-3,t-3) - LogFBE(f+3,t+3)))/coef_reg;
      end
    end
    %-------------------------------------------------------------------------
    % LR through frequency-time
    for f = 4:freq-3
      for t = 4:time-3
        Dft(f-3,t-3) = ((LogFBE(f+1,t+1) - LogFBE(f-1,t-1))....
        + 2*(LogFBE(f+2,t+2)...
        - LogFBE(f-2,t-2))+ 3*(LogFBE(f+3,t+3) - LogFBE(f-3,t-3)))/coef_reg;
      end
    end
   %-------------------------------------------------------------------------
    Df = Df(1:N-1,1:size(Dt,2)); Dt = Dt(1:size(Df,1),:);
    Dtf = Dtf(1:N-1,:);Dft = Dft(1:N-1,:);
    Dt = DCT*Dt; Df =DCT*Df; Dtf = DCT*Dtf; Dft = DCT*Dft;
    CC = [Df;Dt;Dtf;Dft];
end
```

Extracción de los MFCC

```
function [ CC, FBE, MAG ] = mfcc( speech, fs, Tw, Ts, alpha, window, R, M, N, L )

    if( nargin~= 10 ), help mfcc; return; end;
    if( max(abs(speech))<=1 ), speech = speech * 2^15; end;
    Nw = round( 1E-3*Tw*fs );     % frame duration (samples)
    Ns = round( 1E-3*Ts*fs );     % frame shift (samples)
    nfft = 2^nextpow2( Nw );      % length of FFT analysis
    K = nfft/2+1;                 % length of the unique part of the FFT
    hz2mel = @( hz )( 1127*log(1+hz/700) );     % Hertz to mel warping function
    mel2hz = @( mel )( 700*exp(mel/1127)-700 ); % mel to Hertz warping function
    dctm = @( N, M )( sqrt(2.0/M) * cos( repmat([0:N-1].',1,M) ...
                                      .* repmat(pi*([1:M]-0.5)/M,N,1) ) );
    ceplifter = @( N, L )( 1+0.5*L*sin(pi*[0:N-1]/L) );
    speech = filter( [1 -alpha], 1, speech ); % fvtool( [1 -alpha], 1 );
    speech = voicefilter(speech);
    frames = vec2frames( speech, Nw, Ns, 'cols', window, false );
    MAG = abs( fft(frames,nfft,1) );
    H = trifbank( M, K, R, fs, hz2mel, mel2hz ); % size of H is M x K
    FBE = H * MAG(1:K,:); % FBE( FBE<1.0 ) = 1.0; % apply mel floor
    DCT = dctm( N, M );
    CC =  DCT * log( FBE );
```

```
lifter = ceplifter( N, L );
CC = diag( lifter ) * CC;
CC = CC(2:14,:);
```

los códigos para la extracción de los MFFC y los MDLF se basan en el código de Kamil Wojcicki [23].

Bibliografía

[1] F. K. Soong, A. E. Rosenberg, B.-H. Juang, and L. R. Rabiner, "Report: A vector quantization approach to speaker recognition," *AT&T technical journal*, vol. 66, no. 2, pp. 14–26, 1987. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/j.1538-7305.1987.tb00198.x/abstract

[2] "Distance between signals using dynamic time warping - MATLAB dtw." [Online]. Available: https://www.mathworks.com/help/signal/ref/dtw.html

[3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/18626/

[4] K. O. Bailey, J. S. Okolica, and G. L. Peterson, "User identification and authentication using multi-modal behavioral biometrics," *Computers & Security*, vol. 43, pp. 77–89, Jun. 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167404814000340

[5] C.-M. Universidad Politécnica de Madrid, "Boletín de vigilancia tecnológica Biometría," Sep. 2016. [Online]. Available: http://docplayer.es/17020329-Boletin-de-vigilancia-tecnologica-biometria.html

[6] J. de Lara, "A method of automatic speaker recognition using cepstral features and vectorial quantization," *Progress in Pattern Recognition, Image Analysis and Applications*, pp. 146–153, 2005. [Online]. Available: http://www.springerlink.com/index/P3382W74138J627X.pdf

[7] C. Fang, "From dynamic time warping (DTW) to hidden markov model (HMM)," *University of Cincinnati*, vol. 3, p. 19, 2009. [Online]. Available: http://www.academia.edu/download/5803810/fromdtwtohmm_chunshengfang.pdf

[8] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and others, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6296526/

[9] A. Senior, G. Heigold, M. Bacchiani, and H. Liao, "GMM-free DNN acoustic model training," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 5602–5606. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6854675/

[10] C. Weng, D. Yu, S. Watanabe, and B.-H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 5532–5536. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6854661/

[11] P. Matejka, O. Glembek, O. Novotny, O. Plchot, F. Grézl, L. Burget, and J. H. Cernocky, "Analysis of DNN approaches to speaker identification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on.* IEEE, 2016, pp. 5100–5104. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7472649/

[12] A. Mahmood, M. Alsulaiman, and G. Muhammad, "Automatic Speaker Recognition Using Multi-Directional Local Features (MDLF)," *Arabian Journal for Science & Engineering (Springer Science & Business Media B.V.)*, vol. 39, no. 5, pp. 3799–3811, May 2014. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=95753230&lang=es&site=ehos
live

[13] B. Claudio and L. Ricotti, "Speech recognition theory and C++ implementation," *John WILEY&Sons, Ltd*, p. 132, 1999.

[14] E. Formisano, F. De Martino, M. Bonte, and R. Goebel, ""Who"Is Saying"What¿ Brain-Based Decoding of Human Voice and Speech," *Science*, vol. 322, no. 5903, pp. 970–973, 2008. [Online]. Available: http://science.sciencemag.org/content/322/5903/970.short

[15] M. Rothenberg, "The glottal volume velocity waveform during loose and tight glottal adjustments," in *Proceedings VII International Congress on Phonetic Sciences, Montreal*, 1971.

[16] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing.* Pearson Prentice Hall, 2007, google-Books-ID: twtGPwAACAAJ.

[17] S. K. Kopparapu and K. K. Bhuvanagiri, "Recognition of subsampled speech using a modified Mel filter bank," *Computers & Electrical Engineering*, vol. 39, no. 2, pp. 655–662, Feb. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790612001863

[18] A. V. Oppenheim, J. E. Bondaryk, D. T. Cobra, M. M. Covell, M. Feder, E. Weinstein, J. S. Lim, D. W. Griffin, D. J. Harasty, J. C. Hardwick, and others, "Digital signal processing," Research Laboratory of Electronics (RLE) at the

Massachusetts Institute of Technology (MIT), Tech. Rep., 1987. [Online]. Available: https://dspace.mit.edu/handle/1721.1/57007

[19] M. Kevin, "HMM toolkit," Massachusetts EE.UU., 2005. [Online]. Available: https://www.cs.ubc.ca/ murphyk/Software/HMM.zip

[20] J. W. Eaton, D. Bateman, and S. Hauberg, *Gnu octave*. Network thoery London, 1997. [Online]. Available: http://folk.ntnu.no/joern/itgk/octave.pdf

[21] W. Han, C.-F. Chan, C.-S. Choy, and K.-P. Pun, "An efficient MFCC extraction method in speech recognition," in *2006 IEEE International Symposium on Circuits and Systems*, May 2006, pp. 4 pp.–.

[22] M. Nofal, E. Abdel-Reheem, and H. E. Henawy, "Arabic/English automatic spoken language identification," in *1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 1999). Conference Proceedings (Cat. No.99CH36368)*, 1999, pp. 400–403.

[23] "HTK MFCC MATLAB - File Exchange - MATLAB Central." [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab