UNIVERSIDAD AUTÓNOMA DE ZACATECAS
"FRANCISCO GARCÍA SALINAS"
Posgrado del Área de Ingeniería

COMPARACIÓN DE TÉCNICAS DE PARAMETRIZACIÓN ESPECTRAL PARA RECONOCIMIENTO DE VOZ EN IDIOMA ESPAÑOL

Tesis de Maestría

presentada como requisito parcial para obtener el grado de

MAESTRO EN CIENCIAS DE LA INGENIERÍA

Presentada por:

Manuel Alejandro Soto Murillo

Directores de tesis:

Dr. José Ismael de la Rosa Vargas y Dr. Arturo Moreno Baez

UNIDAD ACADÉMICA DE INGENIERÍA ELÉCTRICA

Zacatecas, Zac., 9 de marzo de 2018

APROBACIÓN DE TEMA DE TESIS DE MAESTRÍA

C. Manuel Alejandro Soto Murillo

PRESENTE

De acuerdo a su atento oficio de fecha 23 de enero de 2018, en el cual solicita se le señale el tema a desarrollar para su trabajo de Tesis del Programa de Maestría en Ciencias de la Ingeniería, le manifiesto lo siguiente.

Se aprueba su solicitud, designando como directores de tesis a los profesores Dr. José Ismael de la Rosa Vargas y Dr. Arturo Moreno Baez, mismos que acordaron en fijar a usted el tema titulado:

COMPARACIÓN DE TÉCNICAS DE PARAMETRIZACIÓN ESPECTRAL PARA RECONOCIMIENTO DE VOZ EN IDIOMA ESPAÑOL

en su oportunidad, antes de la impresión de la versión final del documento.

Atentamente

Zacatecas, Zac., 23 de febrero de 2018

_____

Dr. Jorge de la Torre y Ramos

Director de la Unidad Académica de Posgrado en Ingeniería

AUTORIZACIÓN DE IMPRESIÓN DE TESIS DE MAESTRÍA

C. Manuel Alejandro Soto Murillo
PRESENTE

La Dirección de la Unidad Académica de Posgrado en Ingeniería le notifica a usted que la Comisión Revisora de su documento de Tesis de Maestría, integrada por los profesores Dr. José Ismael de la Rosa Vargas, Dr. Arturo Moreno Baez, Dr. Hamurabi Gamboa Rosales, Dr. Efrén González Ramírez y M. en C. Aldonso Becerra Sanchéz, ha concluido la revisión del mismo y ha dado la aprobación para su respectiva presentación.

Por lo anterior, se le autoriza la impresión definitiva de su documento de Tesis de Maestría a fin de dar trámite a la sustentación de su Examen de Grado, a presentarse el 9 de marzo de 2018.

Atentamente
Zacatecas, Zac., 23 de septiembre de 2018

_____

Dr. Jorge de la Torre y Ramos
Director de la Unidad Académica de Posgrado en Ingeniería

APROBACIÓN DE EXAMEN DE GRADO

Se aprueba por unanimidad el Examen de Grado de Manuel Alejandro Soto Murillo presentado el 9 de marzo de 2018 para obtener el Grado de

MAESTRO EN CIENCIAS DE LA INGENIERÍA

Jurado:

Presidente: Dr. José Ismael de la Rosa Vargas _____

Primer vocal: Dr. Arturo Moreno Baez _____

Segundo vocal: Dr. Hamurabi Gamboa Rosales _____

Tercer vocal: Dr. Efrén González Ramírez _____

Cuarto vocal: M. en C. Aldonso Becerra Sanchéz _____

RESUMEN

El reconocimiento de voz es un área de investigación del procesamiento digital de señales con un amplio campo de aplicaciones en diversos sistemas y dispositivos electrónicos, en los que la interacción humano-máquina es deseable o indispensable mediante comandos de voz. La correcta caracterización de la señal de voz y la elección del método adecuado que modele los coeficientes obtenidos en la etapa de extracción de características es esencial para obtener una tasa de reconocimiento significativa. En el presente trabajo, se realizó una comparación de dos técnicas clásicas de parametrización en la etapa de caracterización de la señal de voz; Codificación Predictiva Lineal (LPC) y Coeficientes Cepstrales de Frecuencias Mel (MFCC). Se realizaron diferentes pruebas de estas técnicas con el fin de encontrar la configuración que brinde la mayor tasa de reconocimiento y el menor consumo de recursos (tiempo y cálculo). Se usaron dos frecuencias de muestreo (8 y 16kHz) y se varió el número de coeficientes (8-12 para 8kHz y 16-24 para 16kHz) que caracterizaron a la señal de voz. En la etapa de modelado se hizo uso de la técnica Modelos Ocultos de Markov (HMM). En los resultados se resalta que la técnica de extracción MFCC presentó una tasa de reconocimiento superior que la técnica LPC para la misma frecuencia de muestreo y con el mismo número de coeficientes.

ABSTRACT

Voice recognition is a research area of digital signal processing with a wide range of applications, it is applied in various systems and electronic devices in which human machine interaction is desirable or indispensable by using voice commands. The correct characterization of the voice signal and the choice of the appropriate method that models the coefficients obtained in the feature extraction stage is essential to obtain a significant recognition rate. This document presents a comparison of two classical parametrization techniques, which was carried out in the voice signal characterization stage; Linear predictive coding (LPC) and Mel-frequency cepstral coefficients (MFCC). Different tests of these techniques were made to find the one that provides the highest recognition rate and the lowest consumption of resources (time and calculation). Two sampling frequencies were used (8 and 16kHz) and the number of coefficients that characterized the voice signal were veried (8-12 for 8kHz y 16-24 for 16kHz). In the modeling stage, was used the Hidden Markov model (HMM) technique. In the results, it is highlighted that the MFCC extraction technique presented a higher recognition rate than the LPC technique with the same sampling frequencies and with the same number of coefficients.

Dedicada a mis padres, mi hermana y Karen

Agradecimientos

Agradezco a la Universidad Autónoma de Zacatecas y en especial a la Unidad Académica de Ingeniería Eléctrica, que me han formado profesionalmente como ingeniero y ahora como maestro en ciencias de la ingeniería. Gratifico al Consejo Nacional de Ciencia y Tecnología por otorgarme los recursos correspondientes para mis estudios de maestría.

Agradezco al Dr. Carlos Alberto Olvera Olvera por darme la oportunidad de pertenecer a este programa de maestría, por su apoyo a lo largo de mis estudios de maestría y por su desempeño como director.

Agradezco al Dr. José Ismael de la Rosa Vargas por proponerme este tema de tesis tan interesante, por su apoyo y ayuda a lo largo del desarrollo de la investigación, por proporcionarme las herramientas necesarias y el conocimiento invaluable que me brindó para culminar este trabajo de tesis con éxito. Un académico que inspira a generar y compartir conocimiento.

Agradezco al Dr. Arturo Moreno Baez por brindarme un espacio de trabajo, por su ayuda en la codificación de los algoritmos, por darle seguimiento al desarrollo del proyecto y por su invaluable capacidad de transmitir ese ímpetu de enseñar e investigar, demostrando que no hay trabajo imposible de realizar. Un docente que inspira a cumplir las metas propuestas.

Lo doy gracias a las personas que colaboraron en esta investigación para conformar el corpus de voz; Alma, Ángel, Gaby, Gustavo, Ivon, Juan Manuel, Karen, Lulú y Pablo. También agradezco especialmente a mis compañeros y amigos de cubículo; Ángel, Eduardo, Gaby, Gustavo y Pablo, por su gran apoyo a lo largo de mis estudios de maestría y sus consejos académicos y personales.

Contenido General

Pag.

Lista de figuras

Figura Pag.

Figura Pag.

Lista de tablas

Capítulo 1

Introducción

La comunicación humana ha ido evolucionado a lo largo de la historia, desde las pinturas rupestres y las señales de humo en la antigüedad hasta las tecnologías de la información y comunicación (telecomunicaciones) de hoy en día. Todas las variantes de comunicación han sido utilizadas con el fin de transmitir información, haciendo de ésta una actividad cotidiana. La comunicación oral es una de las capacidades básicas y más esenciales que poseen los seres humanos, al igual que su sistema auditivo. Por esa razón, se puede decir que el habla no sólo es el método más importante a través del cual las personas pueden transmitir información, sino que además sus características muestran el estado emocional de la persona.

El desarrollo científico y tecnológico ha permitido que el ser humano se comunique e interactúe de manera muy simple, sin importar la distancia. Las invenciones de dispositivos electrónicos como; el teléfono, los celulares, computadoras, tabletas electrónicas, entre otros, nos han permitido establecer conversaciones o realizar video llamadas con otras personas alrededor del mundo. Esto con la ayuda de la telefonía fija, las comunicaciones móviles o satelitales y el internet, que permiten la trasmisión de la información entre el emisor y el receptor.

Las necesidades del ser humano para simplificar sus actividades laborales, académicas o sociales son cada vez más exigentes y se han ido satisfaciendo con la creación de nuevas tecnologías, máquinas y dispositivos electrónicos, permitiendo realizar nuestras actividades cotidianas con mayor facilidad. Sin embargo, aún se tiene la necesidad de interactuar con dichas

máquinas y dispositivos de una manera más sencilla, sin tener que utilizar nuestras manos o pies. Se ha optado por hacer uso de la voz, para poder dar indicaciones a dichas máquinas y dispositivos electrónicos, o comunicarse con ellos tal y como la comunicación oral entre humanos.

Cuando la comunicación es entre humano y máquina se presentan varios inconvenientes y factores a tomar en cuenta. Así surgen áreas de estudio como; Procesamiento Digital de Señales (PDS), Procesamiento Digital de Voz (PDV), Reconocimiento Automático de Voz (RAV), Reconocimiento Automático del Habla (RAH) o ASR (por sus siglas en ingles de Automatic Speech Recognition) y el Reconocimiento Automático de Locutor (RAL), cuyo objetivo general es lograr la comunicación humano-máquina.

Los sistemas de reconocimiento del habla son una de las tecnologías con mayor implementación en todo tipo de dispositivos electrónicos que se usan en la actualidad. Dicha tecnología no es un campo nuevo de investigación, ya que sus orígenes datan de poco más de seis décadas atrás. Sin embargo, gracias a las investigaciones realizadas a lo largo de estos 65 años por diversas universidades, laboratorios, compañías e investigadores dedicados al procesamiento de voz, se han obtenido sistemas capaces de reconocer la voz humana con una tasa de reconocimiento relativamente alta a nivel laboratorio o en ambientes libres de ruido y reverberaciones.

En los entornos no ideales, en los cuales son implementados la mayoría de los sistemas de reconocimiento de voz, las tasas de reconocimiento aún no han alcanzado el 100% de efectividad, debido a que los sistemas de reconocimiento de voz implementados en; humanoides, domótica, cajeros automáticos, traductores, búsqueda web, sistemas de seguridad, automóviles, manos libres, búsqueda de música por comandos de voz, información audible del vehículo, lectura o dictado de mensajes de texto y en sistemas de navegación, etc., regularmente son afectadas en su funcionamiento debido al ruido acústico de fondo en el que se encuentran inmersas. Este ruido impide identificar con claridad; el inicio y el final de cada palabra, las palabras y el ruido, afectando la tasa de reconocimiento de voz.

1.1 Planteamineto del problema

Con el fin de mejorar la tasa de reconocimiento de voz y generar más áreas de estudio, se continúan realizando nuevas investigaciones en el campo del procesamiento de voz. Se debe de presentar mayor énfasis en cada una de las etapas de los sistemas de reconocimiento de voz. Este trabajo está enfocado a la etapa de extracción de características o parametrización, en la cual se comparan dos métodos de parametrización espectral; la Codificación Predictiva Lineal (LPC) y los Coeficientes Cepstarales de Frecuencias Mel (MFCC) con diferente número de parámetros y frecuencia de muestreo. Con el fin de determinar cuál de las técnicas y bajo qué condiciones presenta una tasa de reconocimiento mayor, y así hacer una mejora significativa en los sistemas de reconocimiento de voz.

1.1.1 Problema

El problema fundamental que se presenta en la parametrización es la elección de un modelo adecuado de la señal de voz. El cual estima la envolvente espectral de la señal que caracteriza las diferentes realizaciones acústicas de forma temporal. El modelo elegido para la señal debe ser capaz de estimar una envolvente útil para el sistema de reconocimiento de voz. Esta envolvente debe ser, además, susceptible de ser representada con un número reducido de parámetros, cuyo cálculo debe exigir el mínimo gasto computacional posible.

En la actualidad los sistemas de reconocimiento de voz utilizan las técnicas clásicas de parametrización espectral como LPC o MFCC para extraer parámetros característicos de la señal voz previamente preprocesada. En la mayoría de los casos se obtienen de 8-12 parámetros para el análisis de las señales de voz adquiridas a una frecuencia de muestreo de 8kHz. En este trabajo se realizará la extracción de características de señales de voz con frecuencias de muestreo de 8kHz y 16kHZ, se obtendrán 8-12 y 16-24 parámetros respectivamente. Finalmente se compararán ambas técnicas de parametrización con las diferentes configuraciones mencionadas.

1.1.2 Objetivos

Codificar y comparar las técnicas de parametrización espectral LPC y MFCC para reconocimiento de voz en idioma español y determinar el número de parámetros y frecuencia de muestreo que presente la mayor tasa de reconocimiento de voz.

Objetivos Particulares

- Obtener la tasa de reconocimiento de voz con 8-12 parámetros con las técnicas LPC y MFCC a una frecuencia de muestreo de 8kHz.
- Obtener la tasa de reconocimiento de voz con 16-24 parámetros con las técnicas LPC y MFCC a una frecuencia de muestreo de 16kHz.
- Modelar los coeficientes obtenidos mediante Modelos Ocultos de Markov (HMM).
- Determinar cuál de las técnicas presenta la mayor tasa de reconocimiento de voz en el idioma español y bajo que condiciones.

1.1.3 Justificación

En la actualidad los sistemas de reconocimiento de voz se encuentran implementados en un sin número de dispositivos electrónicos como; teléfonos celulares, computadoras, sistemas de navegación de autos, sistemas de seguridad, casas y oficinas inteligentes. Sin embargo, estos dispositivos se encuentran en ambientes ruidosos y por consecuencia la tasa de reconocimiento es afectada. Por esta razón, al mejorar la etapa de extracción de características mediante los métodos de parametrización MFCC y LPC los sistemas de reconocimiento de voz aumentan su tasa de reconocimiento sustancialmente y permiten una comunicación oral humano-máquina sin importar el entorno que les rodea, esto coadyuvara a la solución de los problemas antes mencionados.

1.1.4 Hipótesis

Una correcta elección de la técnica de parametrización de la señal de voz y el número de coeficientes es esencial para obtener una alta tasa de reconocimiento del habla.

Al aumentar la frecuencia de muestreo a 16kHz y el número de parámetros en las técnicas de parametrización espectral LPC y MFCC se coadyuvará a una mejora significativa en la tasa de reconocimiento de voz en sistemas de reconocimiento de voz del idioma español.

1.2 Historia del Análisis de Voz

Los primeros estudios relativos al análisis de voz datan de 1668, cuando el religioso y naturalista inglés John Wilkins publicó un libro en el que se ilustraban dibujos de las posiciones del tracto vocal para diferentes caracteres alfabéticos (fonemas). Wilkins propuso un alfabeto fonético que consistía de 8 vocales y 26 consonantes, en el cual los símbolos representaban las posiciones de la boca al pronunciar los distintos sonidos en inglés [3].

1.2.1 Síntesis de Voz

Los primeros dispositivos para producir voz humana artificial fueron mecánicos y surgieron a finales del siglo XVIII, como el sintetizador de voz mecánico creado por el científico alemán Christian Gottlieb Kratzenstein en 1779, quien construyo 5 silbatos (de geometría distinta) que modelaban el tracto vocal del ser humano y eran capaces de reproducir los sonidos de las vocales en inglés (a, e, i, o, u) [4]. En 1791 fue publicado el trabajo del húngaro Wolfgang Ritter von Kempelen, quien creó la "Máquina de voz Acústico-Mecánica" que funcionaba mediante un fuelle que simulaba los pulmones, un silbato que simulaba las cuerdas vocales y un tubo de cuero como si fuese el tracto vocal, dicha máquina era capaz de producir sonidos aislados y algunas combinaciones sonoras [5].

En 1846 J. Faber construyó la máquina Euphonia con un avance significativo sobre la máquina de Kempelen, ya que era posible variar el tono fundamental y esto permitía producir

voz normal, en susurro, entonar las preguntas, etc.

En 1850, el físico ingles C. Wheatstone hizo una mejora a la máquina de Kempelen, añadiendo un resonador de piel. Alexander Graham Bell y su padre A. M. Bell construyeron la máquina sucesora a la de Wheatstone, tratando de construir un dispositivo lo más semejante posible al aparato fonador humano usando materiales como goma, algodón, alambre, etc. El dispositivo conseguía producir sonidos vocálicos y nasales e incluso frases.

Thomas Alva Edison inventó el fonógrafo en 1876, el cual era un sistema de grabación mecánico elaborado con cilindros de cartón ranurados y recubiertos con una capa de estaño, una bobina acústica cerrada por un lado con un diafragma de pergamino, todo sobre una caja. Las ondas sonoras producidas por la voz eran transformadas en vibraciones mecánicas mediante un transductor acústico-mecánico, estas vibraciones movían un estilete colocado sobre el diafragma, el cual presionaba la hoja de estaño produciendo protuberancias y huecos debido a la diferente presión ejercida por las vibraciones del aire provocadas por la voz. Para escuchar lo grabado, se colocaba el estilete de nuevo al comienzo del cilindro, y la superficie rugosa provocaba variaciones de presión en la caja que eran amplificadas por la bocina [2]. Posteriormente, en 1881 Alexander Graham Bell, Chichester Bell y Charles Sumner Tainter inventaron el gramófono, un dispositivo de grabación de sonidos compuesto de un cilindro giratorio con un revestimiento de cera en el cual por medio de una aguja se perforaban orificios inferiores y superiores dependiendo de la presión ejercida, los cuales correspondían a un sonido entrante (voz).

El primer dispositivo completamente eléctrico fue desarrollado por J.Q. Stewart en 1922. Era un sintetizador constituido por en un interruptor para simular las cuerdas vocales y una serie de circuitos para simular las resonancias de las cuerdas vocales. Otro interruptor cortaba o permitía el paso de la corriente, de la misma manera que lo hacen las cuerdas vocales al provocar una pulsación en la corriente de aire.

C. Paget en 1923 descubrió observando su propia voz que había dos componentes frecuenciales en todos los sonidos vocálicos. A partir de esto construyó un sistema formado por un conjunto de resonadores acústicos parecidos a los tubos de Kratzenstein. Estos resonadores estaban hechos de arcilla y goma e individualmente podían producir todos los sonidos vocálicos y consonánticos.

En la feria mundial de Nueva York de 1939 se presentó un sintetizador electrónico llamado Voder (del inglés Voice Demostrator) que producía discursos continuos controlados mediante un teclado. Era necesario aproximadamente un año de entrenamiento. Un operador manipulaba 14 llaves con la mano que controlaban la estructura resonante del tracto vocal y un pedal de pie que permitía lograr un tono variable. El Voder tenía dos fuentes de sonido: un oscilador que generaba un zumbido periódico y análogo, y un ruido aleatorio para los sonidos sordos. La sección de resonadores contenía 10 filtros pasa-banda que abarcaban todo el rango de frecuencias de la señal. El control de ganancia de cada uno de los 10 filtros podía ser ajustado individualmente mediante llaves. Los sonidos oclusivos se podían producir por tres llaves extras que generaban pulsos transitorios [2]. Una vez que se conocía su funcionamiento, se podía producir un discurso continuo inteligible con facilidad.

Contemporáneamente con el Voder apareció el Vocoder. Era un compresor de banda ancha para la telefonía. Este instrumento partía de un análisis del habla real. La señal de voz se codificaba de forma que podía ser transmitida eficientemente y decodificada posteriormente al otro lado de la línea telefónica. Un banco de filtros separaba las diferentes bandas de frecuencia de la señal. Para poder caracterizar la señal acústica original se calculaba una serie de parámetros. Se detectaba si había sonido sordo o sonoro (las cuerdas vocales sólo vibran para los sonidos sonoros, como las vocales). Se medía también el tono fundamental de vibración de las cuerdas vocales, la amplitud de la señal en cada banda, etc. Estos valores eran multiplexados y transmitidos a través de la línea telefónica. El aparato receptor realizaba la operación inversa mediante un demultiplexor y se reconstruía la señal. Era mejor transmitir los parámetros en vez de señal completa, ya que éstos variaban más lentamente y sólo era necesario transmitirlos

cada 20 ms. Pero el equipo necesario era caro para la época y por eso el Vocoder sólo se usó para aplicaciones muy especializadas. El Vocoder fue importante ya que su sección de recepción es análoga a la que poseen muchos sintetizadores modernos y la sección de transmisión es similar a la etapa de análisis espectral de la mayoría de los reconocedores actuales.

Durante la segunda guerra mundial los Laboratorios Bell diseñaron un espectrógrafo y lo nombraron Sona-graph. Este espectrógrafo de sonidos producía un dibujo de la distribución de energía en el dominio del tiempo (eje horizontal) y de la frecuencia (eje vertical) llamado espectrograma. La energía de los sonidos se mostraba en el grado de ennegrecimiento sobre el papel. En periodos de silencio no se dibuja nada sobre el papel mientras que los sonidos de mediana intensidad aparecen con tonos de grises y los de alta intensidad eran tonos más oscuros. Los primeros espectrogramas aparecieron en 1947 en el libro titulado "Visible Speech" de R. K. Potter, G. A. Kopp y H. G. Green.

Surge un nuevo sintetizador de voz hecho por H.K.Dunnen 1950. Este dispositivo eléctrico modelaba el tracto vocal y presentaba mejores resultados con respecto a los proporcionados por el Voder. Este sistema estaba basado en una fuente de energía eléctrica que simulaba las cuerdas vocales, un modelo de líneas de transmisión (bobinas y capacitores) que representaban al tracto vocal y mediante filtros pasa-baja se generaba retardo tal como ocurre con la onda sonora a través de la cavidad bucal. El tono del sonido sólo podía cambiarse mediante ajustes manuales. Sin embargo, G. Rosen introdujo en 1958 el primer sintetizador controlable dinámicamente. Estaba basado en la misma idea que el de Dunn, usando capacitores y bobinas. El sintetizador era controlado por un dispositivo con retardo que seleccionaba una secuencia de configuraciones del tracto vocal. Las transiciones entre una configuración y otra eran suavizadas electrónicamente.

Actualmente los sistemas de síntesis de voz emplean técnicas digitales y están disponibles como aplicaciones en dispositivos electrónicos y también como productos. El sonido que producían los primeros sintetizadores era muy artificial, pero la calidad ha aumentado notablemente en los últimos años. Ahora los sintetizadores son capaces de producir entonación como en la voz humana, por lo que la voz ya no es monótona. Los vocabularios entre cada sintetizador varían, desde los más simples hasta los que tienen cientos de palabras. Los sintetizadores fonéticos que generan palabras a partir de una cadena de fonemas (la unidad más básica) tienen en principio un vocabulario ilimitado. No obstante, este método implica un gasto computacional mayor y un aumento en el tiempo de procesamiento.

1.2.2 Reconocimiento de Voz

A diferencia de los avances que presentaron los sintetizadores de voz a lo largo de la historia, pasando de ser mecánicos a electrónicos, los avances de los sistemas de reconocimiento de voz se fueron logrando con el desarrollo de las computadoras, siendo las décadas de 1970 y 1980 las de mayores avances para los sistemas de reconocimiento del habla. Sin embargo, a principios del siglo XX se crearon algunos sistemas electrónicos.

J. B. Flowers diseño una máquina para transcribir voz en 1916. La transmisión de mensajes entre submarinos se realizaba mediante un código alfabético especial y se registraban como una línea ondulada sobre papel, para posteriormente ser descifrados. La máquina de Flowers era capaz de convertir los sonidos de las letras en ondas de la misma naturaleza que las que se transmitían entre submarinos. Esta máquina consistía de dos electroimanes y capacitores ajustados para cubrir todo el rango de frecuencias.

Década de 1950

Los primeros intentos de diseñar sistemas de reconocimiento del habla se hicieron en 1950, cuando se comenzaron a investigar las ideas fundamentales de la fonética acústica. Sin embargo, el procesamiento de señales y las computadoras no eran aún herramientas útiles para la

creación de estos sistemas.

A partir de la invención del espectrógrafo y por consecuencia del espectrograma, se dio la posibilidad de crear sistemas de reconocimiento del habla. Fue hasta 1952 que K. H. Davis, R. Buddulph y S. Balashek de los Laboratorios Bell construyeron el primer dispositivo de reconocimiento del habla llamado Audrey, capaz de distinguir los números del uno al diez (en inglés) pronunciados de forma aislada y por un solo locutor. Este dispositivo era completamente electrónico y hacia una comparación entre los parámetros del dígito pronunciado y los patrones correspondientes a cada uno de los diez posibles dígitos [7].

Al pronunciar un número en el micrófono del sistema Audrey, la señal de voz entrante era separada por dos filtros, un filtro pasa-alta para componentes de frecuencia superiores a 900Hz y un filtro pasa-baja para las frecuencias inferiores. Estas señales eran aplicadas a los canales X e Y de un osciloscopio y se dibujaba una curva de dos dimensiones en la pantalla del osciloscopio. Estas curvas eran diferentes para cada uno de los 10 dígitos. Para comparar las curvas automáticamente se dividía el dibujo de dos dimensiones dentro de cuadros y se medía el trozo de curva que ocupaba cada cuadro. Estos cuadros estaban definidos por un circuito de válvulas que activaba uno de los 28 relevadores que lo componían. Cada relevador correspondía a uno de los cuadros en los que se había dividido la pantalla del osciloscopio. En función de los relevadores activados se encendía uno de los indicadores correspondientes al dígito reconocido. Las medidas de los patrones se hicieron con 100 repeticiones de cada uno de los 10 dígitos y se obtenían tasas de reconocimientos comprendidos entre un 97% y 99%.

H.F. Olson y H. Bellar de los Laboratorios RCA, desarrollaron en 1956 la máquina de escribir fonética. La señal de voz era adquirida por un micrófono y después amplificada y comprimida para ajustar la señal a un valor medio, esto hacia que las señales entrantes de los locutores que hablan en tono alto y los de tono bajo fueran lo más parecidas posible. Posteriormente la señal resultante era procesada por un banco de 8 filtros que calculaban la envolvente. Esta información se enviaba a un banco de relevadores mediante un interruptor que actuaba

cada 40 ms desde el momento enque la pronunciación comenzaba. Se obtenía un dibujo característico para los distintos sonidos, dado por las diferentes configuraciones de relevadores. Los datos proporcionados por los relevadores eran decodificados en sílabas y después en letras individuales para actuar sobre las teclas de una máquina de escribir. El sistema fue probado con frases constituidas por 10 sílabas en varias permutaciones diferentes. Si la entonación de la frase era cuidada, resultaba un porcentaje de acierto de un 98%. Continuaron sus experimentos y llegaron a analizar palabras aisladas traduciéndolas a diferentes idiomas. El sistema reconocía palabras inglesas y francesas y las podía traducir a inglés, francés, alemán y castellano [8].

En 1959, en la Universidad College de Inglaterra, Denis Fry y Peter Denes construyeron un sistema reconocedor de fonemas para reconocer cuatro vocales y nueve consonantes [9]. Mediante la incorporación de la información estadística relativa a las secuencias de los fonemas permitidos en inglés, aumentaron la precisión global del reconocimiento de fonemas de las palabras formadas por dos o más fonemas. Este trabajo marcó el primer uso de la sintaxis estadística (a nivel fonético) en el reconocimiento de voz automático [6]. En el mismo año, J. W. Forgie y C. D. Forgie de los laboratorios MIT Lincoln diseñaron un sistema que era capaz de reconocer 10 vocales introducidas en un formato /b/ - vocal -/t/ de forma independiente del locutor [10].

Década de 1960

Para la década de 1960, J. Suzuki y K. Nakata de los laboratorios Radio Research Lab construyeron un reconocedor de vocales en japonés [11]. Sakay y Doshita en la Universidad de Kyoto construyeron un reconocedor de fonemas en 1962, usando un segmentado de voz y un análisis de cruces por ceros de diferentes regiones del enunciado de entrada [12]. En el mismo año IBM presentó su máquina "Shoebox" que podía entender 16 palabras del idioma inglés. En 1963 los laboratorios NEC construyeron un reconocedor de dígitos [13].

Uno de los problemas del reconocimiento del habla es la uniformidad de las escalas de tiempo en los eventos de voz, es decir, cuando se alarga o se acorta una misma palabra al pronunciarla. Sin embargo, T.B. Martin (fundador de Threshold Technology) y los laboratorios RCA desarrollaron un conjunto de métodos de normalización temporal, basados en la capacidad de detectar los inicios y fin de las palabras, lo que redujo significativamente la variabilidad del reconocimiento [14]. A la par, el ucraniano Taras Vintsyuk propuso usar métodos de programación dinámicos para el alineamiento en el tiempo de palabras, surgiendo la Distorsión Dinámica Temporal (DTW, Dynamic Time Warping) [15]. Sakoe y Chiba de los laboratorios NEC también comenzaron a usar ténicas de programación dinámicas para resolver problema de la falta de uniformidad [16]. Numerosas técnicas de programación dinámica, incluyendo el algoritmo de Viterbi, han sido indispensables en el reconocimiento automático del habla.

P. Deves y M. V. Mathews construyeran un reconocedor de dígitos con ayuda de la computadora IBM 704 y con la técnica DTW que permite comparar los parámetros de las palabras iguales pronunciadas a diferente velocidad. Esto aumentó la tasa de reconocimiento. El sistema fue probado con cinco locutores masculinos, obteniendo un error del 6% y de 12% cuando la normalización no era aplicada [6]. Este fue el impulso a realizar una gran cantidad de trabajos de reconocimiento de palabras aisladas y mono locutor.

Raj Reddy de la Universidad Carnegie Mellon en 1967 desarrolló un sistema con dos computadoras interconectadas (una IMB 7090 y una PDP1), con el cual llevó a cabo una investigación pionera en el campo del reconocimiento de habla continua mediante el seguimiento dinámico de los fonemas. El sistema de Reddy fue capaz de identificar fonemas vocálicos y consonánticos pero aún presentaba problemas lingüísticos como fonéticos y semánticos.

Década de 1970

En la década de 1970 surgen las técnicas de parametrización de la señal de voz. Con las técnicas de programación dinámica los laboratorios Bell mostraron cómo las ideas de Codificación Predictiva Lineal (LPC) podrían implementarse en los sistemas de reconocimiento de

voz a través del uso de una medida de distancia apropiada, basada en los parámetros espectrales LPC [17]. Los Laboratorios IBM empezaron a desarrollar sistemas de reconocimiento de voz de amplio vocabulario [18], mientras que los laboratorios AT&T Bell sistemas de reconocimiento de voz independientes del locutor [19].

En 1970 el profesor emirato Gunnar Fant del Real Instituto de Tecnología de Estocolmo planteó un modelo de reconocimiento de voz, en el que se proponían 5 pasos fundamentales; extracción de características, detección de segmentos, transcripción fonética, identificación de palabras e interpretación semántica. En ese mismo año, surge el sistema de reconocimiento HEARSAY (del inglés Hear=oir y SAY=decir) que trabajaba en paralelo con tres reconocedores; uno acústico, uno sintáctico y otro semántico. La frase que se deseaba reconocer era pre-procesada para extraer sus características, el resultado era procesado por los tres módulos y finalmente un sistema mediador que almacenaba toda la información controlaba el sistema y daba el resultado.

El 1971 se presentó el mayor proyecto de la historia del reconocimiento automático del habla, llamado ARPA-SUR (Advanced Research Projects Agency-Speech Understanding Research) [20]. Este proyecto fue dirigido por A. Newell y proponían reconocer discursos continuos, adaptación a nuevos locutores, un diccionario de 10,000 palabras, el error de comprensión debía ser menor de 10% y una respuesta en tiempo real. El objetivo no se cumplió, sin embargo, se obtuvo un mejor conocimiento de las propiedades del habla y de las limitaciones que presentaban los sistemas de reconocimiento automático del habla.

A mediados de la década de 1970 aparecen en el reconocimiento automático del habla los modelos estructurales estocásticos y en específico los Modelos Ocultos de Markov (HMM Hidden Markov Model). Es un proceso doblemente estocástico que tiene un proceso estocástico subyacente que no es observable (de ahí el término oculto), pero se puede observar a través de otro proceso estocástico que produce una secuencia de observaciones. Aunque el HMM era bien conocido y comprendido dentro de algunos laboratorios (principalmente IBM, Instituto

de Análisis de Defensa (IDA) y Dragon Systems), no fue hasta la publicación generalizada de los métodos y la teoría de los HMM a mediados de 1980 que se convirtió en la técnica ampliamente aplicada en prácticamente todos los laboratorios de investigación de reconocimiento de voz en el mundo. Siendo el sistema DRAGON el primero en utilizar los modelos Markovianos.

La Universidad de Carnegie Mellon creó el sistema HARPY en 1976, modelaba las características lingüísticas (fonológicas, léxicas, sintácticas y semánticas) en una red de estados finitos que se obtenía a partir de sub-redes que modelaban las distintas palabras. Este sistema asumía una estructura en red particularizada al problema en específico e introducía una importante modificación al algoritmo de reconocimiento de Viterbi; la búsqueda en haz (Beam Search). Su tasa de errores semánticos fue de aproximadamente el 5%, con un vocabulario de 1011 palabras, una sintaxis artificial de baja perplejidad. El sistema HARPY fue el primero en reducir el tiempo de cálculo y determinar eficientemente la cadena coincidente a la palabra más cercana [21].

A la par del sistema HARPY surge el sistema HEARSAY-II que introdujo la arquitectura Blackboard (pizarrón). Tenía un vocabulario de 1000 palabras y una interpretación correcta del 90% de las frases de prueba. Estaba conformado por un conjunto de técnicas generales de inteligencia artificial y un conocimiento específico de acústica y lingüística, necesarios para llevar a cabo el proceso de reconocimiento de voz. Eran procesos paralelos, independientes, cooperativos y asíncronos que se comunicaban entre sí a través de una estructura de datos global (el pizarrón). Este sistema consiguió tasas de error del 9% a nivel semántico, y del 26% al nivel sintáctico-semántico, con un consumo de recursos de 3 a 4 veces superior que el HARPY. No obstante, su mayor logro ha sido las aportaciones a la arquitectura de sistemas complejos inteligentes, llamados normalmente sistemas basados en el conocimiento.

El grupo de tratamiento del habla del Centro de Investigación IBM J. Thomas propuso un sistema de reconocimiento de discurso continuo. Conformado por un procesador acústico que transcribía la señal de entrada a una cadena de símbolos y un decodificador lingüístico que

traducía la cadena fonética a una cadena de palabras [22] [23]. El decodificador lingüístico constaba de un Modelo de Markov, compuesto de los submodelos correspondientes a las palabras y los fonemas, además de un algoritmo de decodificación (algoritmo de Viterbi), cuyo objetivo era encontrar la cadena de palabras que con mayor probabilidad podía ser producida por el generador de Markov y que fuera además compatible con la cadena observada. Los resultados obtenidos con este sistema se acercaban e incluso superaban a los de HARPY para tareas semejantes, con la ventaja de que gran parte del trabajo de construcción de las fuentes de conocimiento estaba totalmente automatizado.

Para finales de la década de 1970 el japonés Sadaoki Furui propone un método para el reconocimiento del hablante, el cual utilizó una combinación de coeficientes cepstrales instantáneos y sus coeficientes polinomiales de primer y segundo orden, ahora llamados coeficientes Δ y coeficientes $\Delta\ \Delta$ cepstral, como características espectrales fundamentales para el reconocimiento de voz [24]. Este método es ampliamente utilizado en casi todos los sistemas de reconocimiento de voz.

Década de 1980

Para la década de 1980 aún era remanente el problema de crear un sistema robusto capaz de reconocer una cadena de palabras conectadas con una voz fluida. Para tratar de resolver este problema se desarrollaron e implementaron una gran variedad de algoritmos basados en la coincidencia de un patrón concatenado de las palabras individuales.

La investigación de reconocimiento de voz en la década de 1980 se caracterizó por un cambio en la metodología, cambiando de los sistemas basados en extraer conocimiento de forma inductiva, es decir, a partir de muestras basado en un patrón de reconocimiento directo, a un riguroso modelado estadístico predictivo de segmentación en tramas. Utilizando a partir de entonces un modelado acústico basado en los Modelos Ocultos de Markov (HMM), discretos y continuos. Se mejoraron también los sistemas de DTW para el reconocimiento de palabras

conectadas, más concretamente se desarrollaron algoritmos de búsqueda eficientes para determinar la secuencia óptima de patrones para una secuencia de vectores acústicos.

IBM desarrollo la estructura de un modelo de lenguaje (gramatical), representado por reglas sintácticas estadísticas que describían qué tan probable, en un sentido probabilístico, era que una secuencia de símbolos de idioma (por ejemplo, fonemas o palabras) pudiera aparecer en la señal de voz. Este era el modelo de n-gramas, el cual definía la probabilidad de ocurrencia de una secuencia ordenada de n palabras. El uso de modelos de n-gramas de idiomas, y sus variantes, se ha convertido en indispensable en los sistemas de reconocimiento de voz de gran vocabulario [25].

El mayor avance del reconocimiento de voz se dio a mediados de la década de 1980 cuando se presentaron los modelos estadísticos. Las redes neuronales se introdujeron por primera vez en la década de 1950, pero no resultaron útiles a causa de problemas prácticos. Sin embargo, en la década de 1980, se logró un entendimiento más profundo de las fortalezas y limitaciones de la tecnología, así como una comprensión de la relación de esta tecnología a los métodos de clasificación de patrones clásicos [26] [27]. Las redes neuronales artificiales (Artificial Neural Networks) compartían con los modelos ocultos de Markov su carácter inductivo, es decir, el aprendizaje a partir de muestras, pero sus configuraciones clásicas como los perceptrones multicapa (Multi-Layer Perceptron) no eran capaces de representar fenómenos dinámicos como la señal de voz, por lo cual tuvieron que desarrollarse arquitecturas recursivas específicas para superar estas limitaciones.

Década de 1990

En la década de 1990, la investigación del reconocimiento del habla se enfocó en reconocimiento de patrones. El reconocimiento de patrones se basaba en el teorema de Bayes y la estimación de las distribuciones de los datos, en los noventas se transformó en una optimización de la minimización del error de reconocimiento empírico de los patrones ya que las funciones de distribución de la señal de voz no podían ser elegidas o definidas con precisión

por medio de la teoría de decisión de Bayes [28]. El objetivo de los sistemas de reconocimiento en esa época era alcanzar el error de reconocimiento mínimo en lugar de proporcionar el mejor ajuste de una función de distribución a los datos dados (conocidos) establecidos por el criterio de Bayes. Este concepto de minimización de errores produce una serie de técnicas como; el entrenamiento discriminativo y métodos de Kernel.

El error de clasificación mínimo (MCE - Minimum Classification Error) fue propuesto junto con un algoritmo de entrenamiento llamado Desendencia Probabilistica Generelizada (GPD - Generalized Probabilistic Descent) para minimizar una función objetivo que actúa para aproximarse a la tasa de error [17]. El criterio de máxima información mutua (MMI- Maximum Mutual Information). En el entrenamiento de la MMI, la información mutua entre la observación acústica y el promedio del símbolo léxico correcto sobre un entrenamiento conjunto se maximiza. Tanto el MMI y MCE son ejemplos de entrenamiento discriminativo y pueden dar lugar a un rendimiento de reconocimiento de voz superior al enfoque basado en máxima verosimilitud [17].

Diversas técnicas fueron investigadas para aumentar la robustez de los sistemas de reconocimiento de voz en contra de la falta de correspondencia entre las condiciones de entrenamiento y prueba, causada por los ruidos de fondo, la individualidad de voz, micrófonos, canales de transmisión, reverberación de la sala, etc. Las técnicas principales incluyen la regresión lineal de máxima probabilidad (MLLR- Máximum Likelihood Linear Regression) [29], el modelo de descomposición [30] y el modelo de composición paralelo (PMC Parallel Model Composition) [31].

El desarrollo de procesadores más rápidos y de software permitieron que la empresa Dragon lanzara el primer producto de reconocimiento de voz hacia los consumidores, dictado del dragón. Siete años más tarde, el Dragon Naturally Speaking llegó muy mejorado. Se trataba de una aplicación de reconocimiento de voz continua, por lo que podía hablar naturalmente alrededor de 100 palabras por minuto. Sin embargo, se tenía que entrenar el programa durante

45 minutos.

La llegada del primer portal de voz, VAL de BellSouth, fue en 1996; VAL era un sistema de reconocimiento de marcación de voz interactiva que era usado para dar información basado en lo que el usuario decía en el teléfono. VAL abrió el camino para todos los menús activados por voz inexactos que podían conectar a las personas a llamar por los próximos 15 años y más allá.

Década de 2000

Para inicios del siglo XXI se desarrollaron tecnologías capaces de transformar la voz a texto (transcripción automática), con el objetivo de conseguir una salida con más información y mucho más precisa que los sistemas anteriores. Estos sistemas detectaban los límites y la falta de fluidez de la frase, además se centraban en el lenguaje natural y sin restricciones de la comunicación humano-humano en las emisiones en línea y el habla coloquial extranjera en varios idiomas. El objetivo era hacer posible que las máquinas hicieran un mejor trabajo de detección, extracción, recopilación y traducción de la información importante, esto permitió que los seres humanos entendieran lo que se dijo mediante la lectura de las transcripciones en lugar de escuchar las señales de audio [32].

En Japón se llevo a cabo un proyecto nacional de 5 años llamado "Habla Espontánea: Corpus y Tecnología de Procesamiento". Construyeron el corpus de habla espontánea más grande del mundo, "Corpus de japonés espontáneo (CSJ- Corpus of Spontaneous Japanese)" que constaba de aproximadamente 7 millones de palabras, que correspondían a 700 horas de discurso. Se investigaron varias nuevas técnicas como el modelado acústico flexible, la detección de las fronteras de frase, el modelado de la pronunciación, adaptación de modelo acústico así como de lenguaje y la recopilación automático del habla [33] [34].

Se siguieron investigando métodos para aumentar la robustez de los sistemas de reconocimiento de voz, especialmente para el habla espontánea con el fin de tener interacciones inteligentes como las del ser humano en aplicaciones de voz. Para detectar semánticamente partes significativas y rechazar partes irrelevantes en expresiones espontáneas, se ha investigado un enfoque basado en la detección y verificación de voz flexible [35]. Esta estrategia de combinación de reconocimiento y verificación funciona bien especialmente para las expresiones mal formadas.

Los seres humanos utilizan la comunicación multimodal cuando hablan entre sí. Los estudios realizados en la inteligibilidad del habla han demostrado que tener la información visual y la de audio aumenta la tasa de éxito de la transferencia de información, especialmente cuando el mensaje es complejo o cuando la comunicación se lleva a cabo en un entorno ruidoso. El uso de la información visual de la cara, en particular la información de labios, en el reconocimiento de voz se ha investigado, y los resultados muestran que el uso de los dos tipos de información da mejores actuaciones de reconocimiento que cuando sólo se usa el audio o sólo la información visual, en particular en ambiente ruidoso.

El desarrollo de la tecnología de reconocimiento de voz permitió la llegada de la aplicación Google Voice Search para el iPhone. El impacto de la aplicación de Google fue significativo por dos razones. En primer lugar, los teléfonos celulares y otros dispositivos móviles son ideales para el reconocimiento de voz, ya que el deseo de reemplazar sus diminutas teclas en la pantalla sirve como un incentivo para desarrollar mejores métodos de entrada alternativos. En segundo lugar, Google tenía la capacidad de descargar el procesamiento de su aplicación a sus centros de datos en la nube, aprovechando toda esa potencia informática para realizar el análisis de datos a gran escala necesario para hacer coincidencias entre las palabras del usuario y el enorme número de ejemplares de voz humana reunidos.

En 2010, Google añadió el reconocimiento personalizado a la búsqueda por voz en teléfonos Android, para que el software pudiera registrar las búsquedas de voz de los usuarios y

producir un modelo más preciso. La compañía también añadió la búsqueda por voz a su navegador Chrome a mediados de 2011, comenzando con un corpus de 10 a 100 palabras. Hoy en día el sistema de búsqueda por voz en inglés de Google incorpora 230 mil millones de palabras para las consultas reales de los usuarios.

En 2007 un grupo de desarrollo de software llamado "SRI venture group" creó una aplicación llamada "Siri", la cual fue adquirida por Apple Inc. en el 2010. Siri es una aplicación que funciona como asistente personal para todos los usuarios iOS y que fue implementada por primera vez en 2011 en el iPhone 4s. Esta aplicación tiene multiples usos y es capaz de soportar idiomas como; inglés, alemán, francés, japonés, español, italiano, chino y coreano. Al igual que la búsqueda de voz de Google, Siri confía en el procesamiento basado en el corpus de su nube.

El desarrollo de estas nuevas aplicaciones de reconocimiento de voz han hecho posible pasar de la utilidad al entretenimiento. No es difícil imaginar un futuro cercano cuando estaremos mandando a nuestros cafeteros, hablando con nuestros impresores, y dando órdenes en las habitaciones de nuestras casas para nuestra comodidad mediante comandos de voz.

1.3 Estado del Arte

El reconocimiento del habla ha marcado un punto de referencia para una gran cantidad de aplicaciones. Hasta el día de hoy su estudio ha permitido generar nuevos menesteres utilizando la manipulación de las señales de voz. En las técnicas de parametrización se han desarrollando las siguientes investigaciones:

Comparaison of Several Acoustic Modeling Techniques and Decoding Algorithms for Embedded Speech Recognition Systems [36]

Se evaluaron algunos métodos que cumplen los requerimientos de reconocimiento de voz por mapeo en los teléfonos celulares. Las técnicas propuestas fueron evaluadas con reconocimiento de dígitos en inglés y francés. Se investigaron particularmente tres aspectos del procesamiento de voz: parametrización acústica, algoritmos de reconocimiento y modelado acústico.

Los algoritmos de parametrización Coeficientes Cepstrales de Predicción Linel (LPCC - Linear Predictive Coding Coefficients), Coeficientes Cepstarales de Frecuencias Mel (MFCC - Mel Frequency Cepstral Coefficients) y Predicción Lineal Perceptual (PLP - Perceptual Predictive Coding) fueron comparados con el algoritmo LPC, el cuál está incluido en la norma del sistema global para las comunicaciones móviles (GSM -Global System for Mobile comunications). En los resultados se obtuvo que los algoritmos MFCC y PLP se desempeñaron significativamente mejor que el algoritmo LPC. Además el tamaño del vector de características PLP fue reducido de 13 a 6 coeficientes sin pérdida significativa de rendimiento.

En el modelado del sistema de reconocimiento se utilizaron dos técnicas; la distorsión de tiempo dinámico (DTW - Dynamic Time Warping) y los modelos ocultos de Markov (HMM - Hidden Markov Model). Los experimentos mostraron que la técnica de HMM superó a la técnica DTW en el reconocimiento independiente del locutor. Sin embargo, la complejidad de esta técnica es mayor a la de DTW. Con el fin de obtener un buen desempeño y un gasto razonable de recursos, se redujo la complejidad del modelado con HMM. Primero se disminuyó el número de estados por cada modelo (Gaussianas), después se disminuyó la cantidad total de Gaussianas uniendo todos las componentes Gaussianas en un Modelo de Mezcla Maussianas (GMM - Gaussian Mixture Model) de voz genérico.

En la tabla 1.1 se muestran los resultados del reconocimiento del corpus de dígitos en inglés (TI - TI_DIGITS [37]) y el corpus de dígitos en francés (BD - BDSONS [38]). Se puede observar que el algoritmo de LPCC es más eficiente que el LPC embebido en el celular. El error de reconocimiento por palabra decrece el 7% cuando es dependiente de locutor (Spk-ind) y el 32% cuando es independiente de locutor (Spk-dep). Al igual que LPCC, los algoritmos de

MFCC y PLP también superaron significativamente al algoritmo LPC. Sin embargo, requieren más gasto computacional.

Tabla 1.1 Tasa de error de reconocimiento por palabra basado en DTW y varios algoritmos de parametrización.

	BD		**TI**
	Spk-dep	**Spk-ind**	**Spk-ind**
LPC	11.28%	61.14%	65.81%
LPC MSR	15.78%	66.17%	67.51%
LPCC	4.60%	46.68%	42.54%
LPCC MCR	5.00%	34.36%	34.35%
MFCC	0.15%	35.58%	41.66%
MFCC MSR	0.25%	15.41%	33.30%
PLP	0.10%	25.75%	36.29%
PLP MSR	0.23%	22.80%	29.14%
PLP6	0.55%	23.00%	28.09%
PLP6 MSR	0.72%	17.18%	24.19%

La normalización, es decir, la sustracción de la media y reducción de la varianza (MSR) mejoró significativamente el reconocimiento en los casos independientes de locutor, especialmente usando el algoritmo MFCC. Sin embargo, se observa una ligera degradación de la tasa de reconocimiento cuando es dependiente del locutor. Es importante notar la muy ligera pérdida de desempeño (menor del 0.5%) usando el vector compacto de características (PLP6), compuesto de los primeros 5 coeficientes PLP y la energía.

Esquema Unificado de Parametrización de la Señal de Voz en Reconocimiento del Habla [39]

Se propuso un enfoque unificado de la etapa de parametrización de voz, donde se analizan particularmente las técnicas convencionales LPCC y el banco de filtros mel-ceptrum, al igual

que las técnicas híbridas LPC-mel-cepstrum (LMC) y Mel-LPC-cepstrum (MLC). Se incorporó una nueva técnica de deconvolución, nombrada "root homomorphic de deconvolution - RHD".

Las pruebas se hicieron con una base de datos compuesta de varias secuencias de dígitos en inglés libres de ruido [40], muestreada a 20kHz y se submuestreó a 8kHz. Se uso solamente de los locutores adultos (112 de entrenamiento y 113 de prueba).

Se utilizó el sistema de reconocimiento HTK v1.5, el cual está basado en Modelos Ocultos de Markov de densidad continua. Cada dígito fue caracterizado por un modelo de Markov de 10 estados de izquierda a derecha sin saltos, con una matriz de covarianza diagonal y una mezcla por estado. El modelo para el silencio es de mismo tipo pero con 5 estados.

El entrenamiento de los modelos tenía dos etapas: a) La inicialización con el algoritmo ¨Segmental K-means¨, con una segmentación manual previa, b) La reestimación con el algoritmo de ¨Baum-Welch¨, que usaba para ambas fases secuencias de un sólo dígito. Posteriormente los modelos obtenidos se reestimaban nuevamente con tres iteraciones, empleando las secuencias de más de un dígito.

En la fase de prueba se utilizaron tan sólo las cadenas de más de un dígito, para reconocerlas se utilizó el algoritmo clásico de Viterbi.

En los resultados se observó que para las parametrizaciones LPCC y MLC los resultados óptimos se hallan cuando el orden del modelo de predicción, p, y el número N de coeficientes cepstrales coinciden. La técnica LPCC tuvo su máxima tasa de reconocimiento del 70.84% con N=20. La técnica MC consiguió su máxima tasa de 70.57% con N=8. La técnica MLC obtuvo una tasa de 72.19% con N=16, valor intermedio entre los necesarios para las parametrizaciones

clásicas. La técnica LMC alcanzó un tasa de 69.38% y con N=16 y p=20 alcanza su tasa máxima de 70.84%.

Se puede observar que la técnica MC obtuvo peores resultados que las LPCC aunque lo hace con menos de la mitad de los coeficientes y la diferencias de las tasas de reconocimiento es de tan sólo 0.27%. Con el mismo número de parámetros y para valores bajos, MC supera claramente a LPCC. Por lo que es una técnica atractiva en aplicaciones que requieran vectores cepstrales de dimensiones reducidas. Además, se puede constatar que las técnicas híbridas consiguen superar los resultados de las parametrizaciones clásicas.

Al añadir la técnica de deconvolución RHD antes de la étapa de predicción lineal permitió reducir el orden de predicción y el número de cepstrums necesarios, mejorando la tasa de reconocimiento. Los parámetros cepstrales clásicos son más susceptibles a mejorar con la deconvolución RHD.

A New Nonlinear speaker parameterization algorithm for speaker identification [41]

Se propuso un nuevo algoritmo de codificación basado en la predicción no lineal: el modelo de Codificación Predictiva Neural (Neural Predictive Coding - NPC), el cual es una extensión no lineal del clásico codificador LPC. El rendimiento del algoritmo se estimó por dos métodos de identificación de locutor diferentes: Esfericidad Aritmética-Armónica (Aarithmetic-Harmonic Sphericity -AHS) y los Modelos Vectoriales Auto-Regresivos (Auto-Regressive Vectorial Models - ARVM). Fueron propuestos dos métodos diferentes de codificación para el algoritmo NPC: la inicialización de redes neuronales clásicas y la inicialización lineal. El primer método fue combinado con las técnicas de extracción de características clásicas (LPCC, MFCC, etc.) con el fin de mejorar los resultados de la codificación clásica sola, obteniendo una tasa de 97.55% para MFCC y una tasa de 98.78% para MFCC+NPC. Mientras que para la inicialización lineal se obtuvo el 100%.

Para evaluar el desempeño de la NPC, se aplicó la tarea de identificación de locutor. Identificando entre 49 locutores de la base de datos Gaudi [42]. La base de datos fue submuestreada a 8kHz, pre-enfatizada por un filtro de primer orden cuya función de transferencia es $H(z) = 1.0.95z - 1$. Se utilizaron ventanas Hamming de 30ms y un traslapa entre tramas de 2/3. Se calculó un vector de parametrización de orden 16. Se uso un minuto de lectura para el entrenamiento y cinco oraciones de entre 2-3 segundos de longitud para el entrenamiento.

Se compararon los métodos de codificación NPC, LPC (se basa en modelar linealmente el tracto vocal, por lo que se adapta más a los sonidos de voz), LPCC (calcula una envolvente espectral LPC y después la convierte a coeficientes cepstrales, es el método de codificación más usado en el reconocimiento de locutor), MFCC (es el método de codificación más implementado en los sistemas de reconocimiento, basado en la descomposición de la señal de voz con ayuda de un banco de filtros en escala Mel)y PLP (se basa en la escala de Bark no lineal, usado en reconocimiento de locutor y de voz).

Con el fin de medir los resultados de cada método de codificación, se utilizaron dos métodos de identificación de locutor diferentes: Esfericidad Aritmética-Armónica y los Modelos Vectoriales Auto-Regresivos. Para el método AHS se obtuvieron los siguientes resultados:

Tabla 1.2 Resultados experimentales para los diferentes métodos de codificación de voz con el método de identificación de locutor AHS.

Método de codificación de voz	**Tasa de reconocimiento (%)**
LPC	90.61
LPCC	96.73
MFCC	97.55
PLP	86.12
NPC(inicialización aleatoria)	61.63
NPC(inicialización lineal)	100

En la tabla 1.2 se puede observar que para los métodos tradicionales, MFCC y LPCC obtuvieron las mayores tasas de reconocimiento. Estos métodos tratan de modelar el contexto fonético y las características del locutor. El método PLP tuvo la tasa de reconocimiento más baja de los métodos clásicos, debido a que suprime las características dependientes del locutor. Esta es una buena técnica para el reconocimiento de voz y no de locutor. En cuanto al método NPC, se obtuvo una baja tasa de reconocimiento para la inicialización aleatoria y la máxima tasa de reconocimiento con la inicialización lineal.

Tabla 1.3 Resultados experimentales para los diferentes métodos de codificación de voz con el método de identificación de locutor ARVM.

Método de codificación de voz	**Tasa de reconocimiento (%)**
LPC	90.61
LPCC	93.06
MFCC	95.69
PLP	78.36
NPC(inicialización aleatoria)	88.57
NPC(inicialización lineal)	100

En la tabla 1.3 se muestran los resultados del método ARVM, se puede observar que se obtuvieron resultados similares para los métodos tradicionales de codificación LPC, LPCC, MFCC y PLP. El metodo NPC con inicialización linael siguió manteniendo su tasa del 100%, confirmando el gran desempeño de este método. Sin embargo, para el método NPC con inicialización aleatoria aumento la tasa casi 27%. Una de las razones es que el método AHS es un método unimodal que se limita a modelar correlaciones lineales.

Se concluyó que el método de extracción de características NPC con inicialización lineal brinda mejoras significativas (100% con AHS y ARVM) comparado con la inicialización aleatoria(61.63% con AHS y 88.57% con ARVM) y los métodos de parametrización tradicionales (LPC, LPCC, MFCC y PLP).

Comparative Wavelet and MFCC Speech Recognition Experiments on the Slovenian and English SpeechDat2 [43]

Se evaluó el desempeño de dos métodos de análisis de voz no lineales en el reconocimiento automático de voz. Se hizo un análisis comparativo entre dos métodos de parametrización; MFCC y un conjunto de características basado en wavelets, evaluando sus tasas de reconocimiento.

Para lograr una descomposición de frecuencia similar a la utilizada en la parametrización en escala de mel, se utilizó el árbol de descomposición perceptual de paquetes wavelet [44], el cual produce los parámetros de paquete wavelet (WPP).

La experimentación se llevo a cabo con la base de datos "Slovenian and English SpeechDat2" de 1000 hablantes, usando los métodos de parametrización MFCC y WPP. La herramienta utilizada en el reconocimiento automático de voz fue HTK. Esto permitió llevar a cabo experimentos controlados en seis diferentes subconjuntos de vocabulario Speech hDat2 (frases sí/no, nombres de ciudades, palabra ricas en fonética, dígitos, etc).

Con el fin de analizar el rendimiento de reconocimiento de línea de base que reflejaría las diferencias en la descomposición de frecuencia entre los parametrizaciones MFCC y WPP, se decidió no incluir la información dinámica en los vectores de características. Es bien sabido que las características MFCC delta mejoran el rendimiento de los modelos ocultos de Markov. Más bien se trató de analizar cómo las diferencias de transformación subyacentes inherentes influyen en el MFCC y el rendimiento de reconocimiento basado en WPP.

Se usaron ventanas de 25ms para la parametrización MFCC y de 32 ms para WPP debido a las estructuras de descomposición específicas. También se probó el rendimiento del reconocimiento usando una ventana de voz de 32 ms en el cálculo de MFCC, cuyos resultados tuvieron una menor tasa de reconocimiento. Los vectores de características estaban compuestos de 24 parámetros en ambos casos y contenían las energías logarítmicas. También se utilizó la

misma velocidad de salto de ventana de voz, 10 ms.

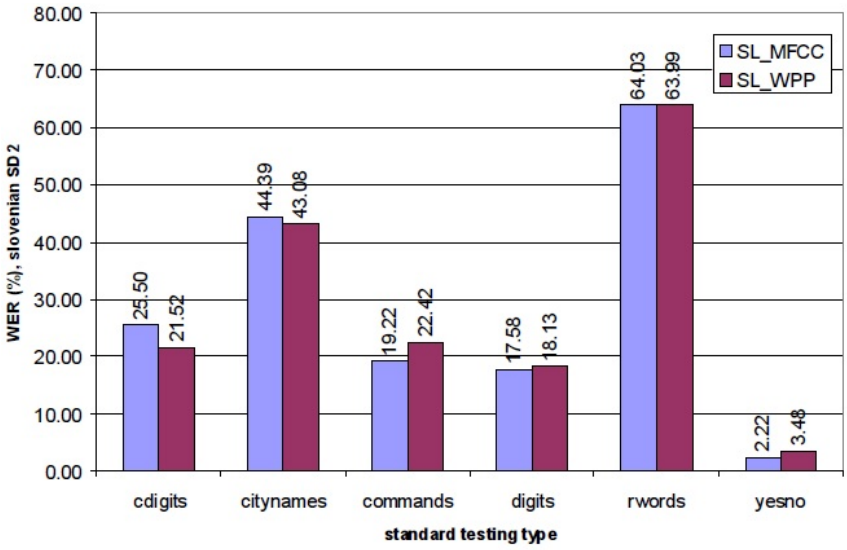


Figura 1.1 Tasas de error de palabras para seis pruebas estándar en esloveno usando MFCC y características wavelet (WPP)[43].

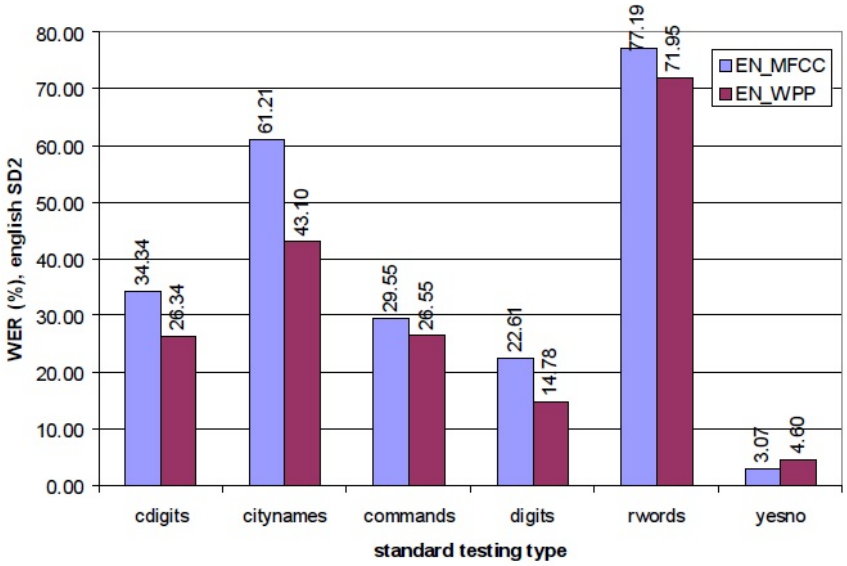


Figura 1.2 Tasas de error de palabras para seis pruebas estándar en inglés usando MFCC y características wavelet (WPP)[43].

Para determinar el rendimiento del reconocimiento entre Esloveno e Inglés de la base de datos, se destinó a proporcionar información sobre la robustez de las características de ruido.

Se determinó una diferencia significativa en el promedio global de la relación señal a ruido (SNR) en esloveno e inglés.

Se puede observar en las figuras 1.1 y 1.2 que los parámetros que brindan mayor información son los nombres de ciudades (citynames) y las palabras ricas en fonética (rwords). Estas pruebas son representativas debido al diverso contenido fonético y sirven de base para evaluar el éxito general de los métodos de parametrización implicados. Debido a su contenido fonético, fueron las dos pruebas con la tasa de error con mayor porcentaje.

Los experimentos en esloveno muestran una pequeña mejora de los resultados de reconocimiento con las características wavelet en las pruebas de nombres de ciudades, palabras ricas en fonética y dígitos. Mientras que en las pruebas de comandos, frases sí/no y otros dígitos la tasa de error por palabra fue menor en la parametrización MFCC.

Los experimentos en inglés produjeron resultados consistentemente mejores obtenidos por las características de la onda. Esto podría implicar que las características wavelet son más robustas en las condiciones de ruido variable. Únicamente en la prueba frases sí/no la parametrización WPP tuvo una mayor tasa de error por palabra que la parametrización MFCC.

Se concluyó que el uso de wavelets puede traer potenciales mejoras en el reconocimiento automático de voz. Otros trabajos y mejoras deberían incorporar el uso de coeficientes delta y delta-delta. El experimento de clasificación de fonemas dentro y entre lenguajes también podría ser considerado para dar información adicional sobre las propiedades específicas de las técnicas de parametrización. Dado que SpeechDat2 representa una base de datos de teléfono ruidosa el uso reducción de ruido podría ofrecer una base sólida para aumentar la robustez del método de parametrización wavelet de ruido y además mejorar los resultados de reconocimiento.

La importancia y contribución de la investigación presente en este documento es fundamental para el desarrollo de sistemas de reconocimiento de voz, ya que se obtuvo un correcto

funcionamiento de las técnicas de extracción de características que se desarrollaron (LPC y MFCC). Las tasas de reconocimiento obtenidas a partir de la técnica de extracción MFCC fueron mayores en todas las pruebas realizadas con respecto a LPC, tanto para las pruebas con una frecuencia de muestreo $Fm = 8kHz$ en las que se variaron el número de coeficientes entre 8 y 12, como para las pruebas a $Fm = 16kHz$ con una variación de coeficientes de dos en dos entre 16 y 24. Por otro lado, al aplicar la técnica de extracción LPC la tasa de reconocimiento se decrementó a medida que se aumentó la frecuencia de muestreo y el número de coeficientes. Se puede resaltar que los resultados obtenidos de esta investigación confirman los resultados obtenidos en las investigaciones presentadas en el estado del arte [36] [39] [41].

En el capitulo 2 se esclarece la diferencia entre voz y habla, se describe el proceso de producción del habla y la adquisición del sonido por el oído, se presenta la clasificación del sonido de la voz y sus características, finalmente se explica el proceso de comunicación oral. En el capitulo 3 se define que es el reconocimiento del habla y de voz, como se clasifica el reconocimiento del habla, se detalla que es un sistema de reconocimiento automático del habla y se describen cada una de las etapas que lo conforman (adquisición de voz, pre-procesamiento, extracción de características y reconocimiento de voz). En el capitulo 4 se presenta el desarrollo experimental de la investigación, la implementación de cada una de las etapas descritas en el capítulo 3 y se muestran los resultados obtenidos de la investigación. El último apartado corresponde a las conclusiones y trabajo futuro.

Capítulo 2

La voz

El propósito del habla es la comunicación. Mediante la comunicación oral el ser humano transmite mensajes e información especifica que se desea difundir entre sus escuchas. Para transmitir este mensaje primeramente es necesario producirlo. El proceso de producción del habla no solo es la expulsión de aire de los pulmones y la formación de las palabras en la boca, sino un proceso que inicia en el cerebro en ciertas aéreas específicas en las cuales se genera una idea de lo que se desea transmitir. Una vez formulado el mensaje, mediante impulsos eléctricos se genera el movimiento de los órganos involucrados en la producción del habla y por medio de ellos se formula la voz con las características correspondientes a cada individuo. En este capítulo se estudia el proceso de producción de voz a detalle; la producción del habla en el cerebro y de voz en el aparato fonador, la clasificación de los sonidos de la voz según sus propiedades y el proceso de comunicación oral entre los seres humanos.

2.1 La voz y el habla

El habla es el acto por medio del cual una persona hace uso de una lengua (como el español) con la finalidad de poder comunicarse, elaborando previamente un mensaje según las reglas gramaticales lingüísticas determinadas de la lengua en que se está comunicando. El habla determina la forma de expresarse (formalmente o informalmente) de cada individuo en particular al comunicarse e intercambiar pensamientos.

La voz es un sonido que se caracteriza por tres elementos [45]:

- **La intensidad:** Es equivalente al volumen. El aire al salir de los pulmones golpea la glotis y produce vibraciones. Cuanto más amplias sean, mayor será la fuerza. Se mide en decibelios (dB) y para tener una referencia, una conversación normal ronda entre los 50 dB. Tiene efectos en el oyente porque transmite emociones. Un volumen de voz alto se asocia a la agresividad, nerviosismo, tensión y lejanía. Al contrario, un volumen bajo puede sugerir depresión, cansancio y proximidad.

- **El tono:** Está relacionado con la cantidad de vibraciones que posee una onda de sonido. A mayor número más aguda será la voz. Estas vibraciones se producen en la laringe y se miden en Hertz (Hz). Las voces masculinas oscilan entre los 75Hz y los 200Hz. Las femeninas entre los 150 Hz y los 300Hz .

- **El timbre:** Es lo que permite que distingamos entre dos sonidos de igual intensidad y tono. El aire que sale de los pulmones, recorre y choca con la laringe, labios, dientes y lengua; tiene peculiaridades únicas dependiendo de la morfología de cada persona. Esta característica es el carnet de identidad de cualquier voz. Aporta mucha información real o imaginaria sobre la edad, la apariencia física e incluso una especie de retrato de la personalidad del hablante.

La voz humana es una combinación de sonidos en diferentes frecuencias, los hombres hacen vibrar las cuerdas vocales cien veces por segundo y las mujeres doscientas veces por segundo. Estas vibraciones provocan resonancias en el tracto vocal de 200Hz a 500Hz. La frecuencia sonadora de la voz oscila entre los 60 y 8000Hz y difiere entre hombres, mujeres y niños. Sin embargo, la voz humana sólo contiene información lingüística en el rango de los 200Hz a los 8kHz. Las frecuencias altas se denominan frecuencias de tono alto (más común en mujeres), mientras que las bajas son frecuencias de tono grave (más común en hombres). La amplitud de la señal de voz tiene una relación directamente proporcional a la energía o intensidad con la que se produce la señal. Por otro lado, el timbre de la señal de voz será determinado a partir de la forma de vibración de la señal, es decir, los armónicos de la señal [46].

Las señales de voz son ondas sonoras (ondas mecánicas longitudinales) que se originan por el movimiento de alguna porción de un medio elástico (sólido, líquido o gaseoso) con respecto a su posición de equilibrio, y debido a las propiedades elásticas del medio, esta perturbación puede desplazarse de un lugar a otro. Existe un gran margen de frecuencias entre las cuales se pueden generar ondas mecánicas longitudinales. Las ondas sonoras se reducen a los límites de frecuencia que pueden estimular el oído humano para ser percibidas en el cerebro como una sensación acústica. Estos límites de frecuencia audible se extienden de aproximadamente 20Hz a cerca de 20kHz y se llaman límites de audición [47].

2.2 Producción del habla en el cerebro

La producción del habla va más allá de la expulsión de aire de los pulmones y la formación de las palabras en la boca. Este proceso de producción inicia con la interacción neuronal en regiones específicas del cerebro y la actividad motora cerebral, los cuales activan y controlan el movimiento de los músculos, órganos y huesos involucrados en la generación del habla.

Los aspectos motores cerebrales para el proceso de producción del habla implican dos etapas principales de pensamiento; la primera es la formación de las ideas que se van a expresar y las palabras que se pretende emplear, la segunda es el control motor de la vocalización y la propia emisión del habla.

En la mayoría de los seres humanos, las actividades del lenguaje son controladas en los lóbulos frontal, temporal y parietal del hemisferio izquierdo del cerebro. Hay dos regiones fundamentales para la producción y comprensión del lenguaje. El área de Broca, que se localiza en la circunvolución frontal inferior justo por delante de la corteza motora (figura 2.1), se encarga del control del habla. El área de Wernicke, que se encuentra en las circunvoluciones supramarginal y angular del lóbulo parietal y en la parte posterior de la circunvolución temporal superior (figura 2.1), se encarga de la comprensión del lenguaje [48]. Ambas áreas se

conectan entre sí mediante un haz de fibras nerviosas llamado fascículo arqueado.

Figura 2.1 Regiones del cerebro involucradas con la producción del habla [49].

La figura 2.1 muestra las regiones del cerebro involucradas en la producción del habla; área de Broca, área de Wernicke.

El área de Broca está encargada de la producción del habla y se divide en dos sub-áreas fundamentales: la triangular (anterior), que se encarga de la programación de las conductas verbales; y la opercular (posterior), que se encarga de coordinar los órganos del aparato fonador para la producción del habla debido a su posición adyacente a la corteza motora. Una lesión en esta área (afasia motora) no impide la vocalización, pero imposibilita el correcto funcionamiento del control de la laringe, labios, boca, aparato respiratorio y otros músculos auxiliares del lenguaje, impidiendo la emisión de palabras completas y genera sonidos descoordinados o algunos términos sencillos como ¨si¨ o ¨no¨ [49].

El área de Wernicke está encargada de la elaboración de los pensamientos y la elección de las palabras. Una persona con lesión en esta región (afasia de Wernicke) no sería capaz de reunir una secuencia de palabras oportunas para expresarse, podría pronunciar palabras con fluidez pero sin sentido o en una lesión mayor no podría formular los pensamientos que quiera comunicar [49].

La producción del habla involucra el movimiento de varios músculos. Mediante la excitación de una neurona aislada en la corteza motora se activa un movimiento específico y no el movimiento de un músculo en específico. Para hacerlo, excita un patrón de músculos independientes, cada uno de los cuales aporta su propia dirección y fuerza de movimiento muscular.

El área cortical que se encuentra emparentada con el área Broca se encarga del funcionamiento respiratorio adecuado, por lo que la activación respiratoria de las cuerdas vocales puede producirse a la vez que los movimientos de la boca y la lengua durante el habla. El habla implica la región de producción de pensamiento, la región de producción del habla, las regiones de control respiratorio del encéfalo, el aparato respiratorio y las estructuras de articulación y resonancia de las cavidades oral y nasal.

2.3 Producción del habla en el aparato fonador (fonación y articulación)

El aparato fonador del ser humano está formado por un conjunto de huesos, cartílagos, músculos y órganos que tienen como función producir el habla humana. El aparato está conformado por los órganos de respiración donde se almacena y circula el aire (las cavidades infraglóticas: pulmones, bronquios y tráquea); los órganos de fonación, donde el aire se convierte en sonido (las cavidades glóticas: laringe, cuerdas vocales y resonadores nasal, bucal y faríngeo) y los órganos de articulación, donde el sonido adquiere sus cualidades que caracterizan a cada voz (las cavidades supraglóticas: paladar, lengua, dientes, labios y glotis).

El habla está formada por dos funciones mecánicas que se llevan a cabo en el aparato fonador; la fonación, que se realiza en la laringe, y la articulación, que se realiza en las estructuras de la boca [50].

2.3.1 Fonación

La fonación es el movimiento muscular producido en la laringe. Se realiza durante la respiración, cuando el aire contenido en los pulmones, sale de la caja torácica y el diafragma y, a través de los bronquios y la tráquea llega a la laringe donde se encuentran los pliegues vocales comúnmente llamados cuerdas vocales. La laringe está adaptada especialmente para actuar como vibrador.

En el proceso de generación del habla, el sonido inicial proviene de la vibración de las cuerdas vocales conocida como vibración glotal, es decir, el efecto sonoro se genera por la rápida apertura y cierre de las cuerdas vocales conjuntamente con el flujo de aire emitido desde los pulmones. El elemento vibrador son las cuerdas vocales, que son dos membranas ubicadas dentro de la laringe y cuya abertura entre ambas cuerdas se denomina glotis. Las cuerdas vocales se extienden desde las paredes laterales de la laringe hasta el centro de la glotis; se distienden y mantienen en su posición por varios músculos específicos de la propia laringe.

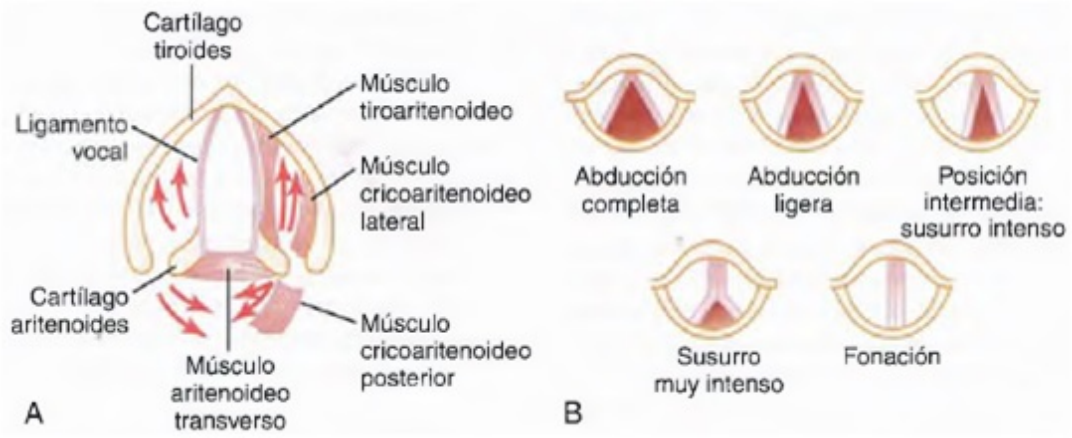


Figura 2.2 A) Anatomía de laringe. B) Posiciones de las cuerdas vocales durante la fonación [49].

La figura 2.2A muestra la imagen de una disección de los pliegues vocales. Se muestra que en el interior de cada una de las cuerdas hay un ligamento vocal. Este ligamento está unido por delante al cartílago tiroides coloquialmente llamado "nuez de Adán". Por detrás el ligamento vocal esta unido a las apófisis vocales de los dos cartílagos aritenoides. El cartílago tiroides

y los cartílagos aritenoides se articulan por abajo con el cartílago cricoides. La figura 2.2B muestra las posiciones de las cuerdas vocales en diferentes fonaciones.

Durante la respiración normal (12-20 respiraciones por minuto) las cuerdas vocales están muy abiertas para facilitar el paso del aire. Mientras que en la fonación las cuerdas se juntan entre sí, provocando que el flujo del aire que pasa entre ellas las haga vibrar. El tono de la vibración está determinado principalmente por el grado de distensión de las cuerdas, por el grado de aproximación de las cuerdas entre si y por la masa de sus bordes [49].

Cuando la glotis comienza a cerrarse, el aire proveniente desde los pulmones experimenta una turbulencia, emitiéndose un ruido de origen aerodinámico. El rango vocal lo determina la flexibilidad de las cuerdas vocales, que permite diferenciar los distintos tipos de voces que caracterizan a cada persona, en función del tono, intensidad y timbre. Al cerrarse más la glotis, las cuerdas vocales comienzan a vibrar a modo de lengüetas, produciéndose un sonido tonal, es decir periódico y cuya frecuencia varia en forma inversa al tamaño de las cuerdas y que carece de información lingüística. Este sonido es propio del hablante y es más agudo para el caso de mujeres y niños.

Después de que el aire atraviesa la glotis el sonido pasa a través de la cavidad supraglótica, que es la porción del aparato fonador que permite modificar el sonido dentro de márgenes muy amplios. Está conformado principalmente por tres cavidades, la cavidad oral, la cavidad labial y la cavidad nasal, correspondientes a la garganta, los labios y la nariz respectivamente. Estas cavidades constituyen resonadores acústicos, los cuales modifican los sonidos de acuerdo a la forma que adopten la lengua y los labios, permitiendo efectuar esta variación de manera voluntaria. El habla humana, una vez que sale de los resonadores, es moldeada por los articuladores (paladar, lengua, dientes, labios y glotis), transformándose en sonidos de voz: palabras, sílabas o fonemas.

2.3.2 Articulación

La articulación está constituida por las actividades musculares de la boca, la lengua, la laringe, la tráquea, las cuerdas vocales, la cavidad faríngea, la cavidad nasal, la cavidad oral, la lengua, los dientes, el paladar duro y blando y los labios, que son los responsables de la entonación, el ritmo y las variaciones rápidas de intensidad en los sonidos sucesivos. Los tres órganos principales de la articulación son los labios, la lengua y el paladar blando. Las regiones facial y laríngea de la corteza motora activan estos músculos, mientras que el cerebelo, los ganglios basales y la corteza sensitiva contribuyen a controlar la secuencia y la intensidad de los movimientos musculares. La destrucción de cualquiera de estas regiones puede provocar una incapacidad parcial o total para hablar con claridad. La posición concreta de los articuladores determinará el sonido de voz emitido.

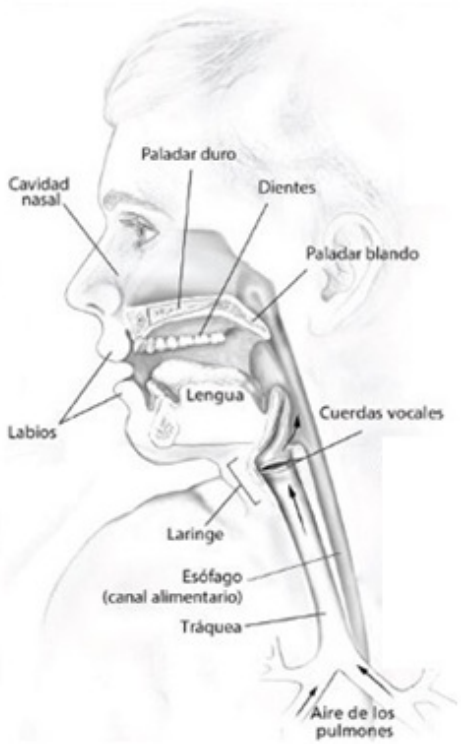


Figura 2.3 Anatomía del aparato fonador y órganos involucrados en la articulación [52].

2.4 Clasificación del sonido de la voz y sus características

La voz adquiere sus características primarias una vez que el aire pasa por los órganos de fonación y los órganos de articulación se encargan de agregar las características adicionales a la voz de cada persona.

Según la lingüística determinada de cada lengua o i-dioma se tiene un conjunto de sonidos de voz que son capaces de producirse por el tacto vocal. Estos sonidos propios de cada idioma se denominan ¨fonemas¨ y su sonoridad depende de las características lingüísticas de cada lengua. Cada idioma tiene sus propios fonemas; en español son alrededor de 29 fonemas, en inglés cerca de 42 fonemas y en francés casi 50 fonemas. Todas las oraciones, frases o palabras se pueden descomponer en fonemas y para representarlos se escribe cada sonido entre diagonales /fonema/. Por ejemplo, la palabra ¨Hola¨ está compuesta por los fonemas /o/ + /l/ + /a/.

En general, las dos primeras formantes F_1 y F_2 (un plano) permiten clasificar los diferentes sonidos o fonemas. La frecuencia de las formantes depende de la forma y de las dimensiones del tracto vocal y por cuestiones de modelado se toman como referencia las tres primeras formantes F_1, F_2 y F_3 (una nube). Las señales de voz pueden clasificarse en tres tipos según su sonido [51];

1. **Sonoras**
    (a) Vocales orales
    (b) Vocales nasales
    (c) Semi-vocales
    (d) Consonantes nasales
2. **No sonoras o fricativas**
    (a) Fricativas dentales

(b) Fricativas alveolares

(c) Fricativas post-alveolares

3. **Plosivos**

2.4.1 Sonoras

Las señales de voz sonoras se generan por la vibración de las cuerdas vocales al mantener la glotis abierta, lo que permite que el aire fluya a través de ella. Los sonidos son sonoros cuando se utilizan las cuerdas vocales. El tracto vocal actúa como una cavidad resonante reforzando la energía en torno a determinadas frecuencias (formantes). Los sonidos sonoros se caracterizan por tener alta energía y un contenido frecuencial en el rango de los 300Hz a 4000Hz presentando cierta periodicidad, es decir, son de naturaleza cuasi-periódica. La frecuencia de oscilación de las cuerdas vocales en los sonidos sonoros se denota por el símbolo F_0, a esta se le denomina frecuencia fundamental o pitch (en inglés). Esta frecuencia fundamental varía para cada persona.

Las vocales orales, las vocales nasales y las semi-vocales se caracterizan por ser sonoras, pero existen consonantes que también lo son, tales como, la /b/, /d/ y la /m/, entre otras.

Cuando se pronuncia una vocal la excitación de las cuerdas vocales es cuasi-periódica y hay una estabilidad en el tracto vocal. Las vocales son sonidos más estables en el tiempo y con alto nivel de energía. Las diferentes vocales /a/, /e/, /i/, /o/, /u/, se distinguen según se encuentran ubicadas las primeras formantes. Para determinar si el sonido es vocal oral o vocal nasal, depende de si el tracto nasal está abierto o cerrado. Los vocales nasales se distinguen de las orales por las anti-resonancias introducidas por las fosas nasales, algunos ejemplos son: an - /ã/ on - /õ/.

Un sonido semi-vocal se produce cuando la excitación glotal en el tracto vocal es rápida. Según la velocidad de la pronunciación de las semi-vocales, serán identificadas como una cantidad independiente o como un conjunto de fonemas diferentes, por ejemplo: /w/,/l/,/r/,/y/.

Las consonantes nasales tales como /m/, /n/ y /ñ/ son producidas con la excitación glotal y el tracto vocal totalmente oprimido en algún punto a lo largo de la boca. El velum se baja de tal forma que el aire fluye a través de la cavidad nasal, con un sonido rodeado por las fosas nasales. La cavidad vocal se acopla acústicamente con la faringe.

2.4.2 No Sonoras

Las señales de voz no sonoras también conocida como señales fricativas, sordas o no vocalizadas, se caracterizan por tener un comportamiento aleatorio en forma de ruido blanco (ruido que contiene todas las frecuencias). Tienen una alta densidad de cruces por cero y baja energía comparadas con las señales de tipo sonora. Durante su producción no se genera vibración de las cuerdas vocales, ya que el aire atraviesa un estrechamiento y genera una turbulencia en el tracto o cavidad vocal.

Los sonidos fricativos se dividen en tres, dentales, alveolares y post-alveolares. Los sonidos fricativos dentales se producen cuando se posicionan los dientes incisivos superiores sobre el labio inferior. Los fricativos alveolares se producen al colocar la lengua detrás de los incisivos. Los fricativos post-alveolares se producen al colocar la lengua entre los dientes incisivos inferiores y superiores. Si vibran las cuerdas vocales al mismo tiempo que se produce el ruido de fricción, los sonidos fricativos son sonoros. Por otra parte, si se deja pasar el aire sin emitir sonido, se denomina fricativo sordo o no vocalizado. Las consonantes que producen los sonidos fricativos son la 'f' y la 's'(sordos), la /v/ y la 'z' (vocalizados). Estos sonidos poseen toda su energía en las formantes.

2.4.3 Plosivas

Las señales de voz plosivas se generan cuando el tracto vocal se cierra en algún punto, lo que causa que el aire se acumule para después salir expulsado repentinamente (explosión). Se caracterizan por que la expulsión de aire está precedida de un silencio. Si las cuerdas vocales vibrar al producir este sonido se obtiene un sonido oclusivo vocalizado, de lo contrario será un oclusivo sordo o no vocalizado. Estos sonidos se generan por ejemplo, cuando se pronuncia la palabra 'campo'. La p es una consonante de carácter plosivo, y existe un silencio entre las sílabas 'cam' y 'po'. Los sonidos plosivos son producidos por las consonantes /b/, /d/, /g/ (vocalizados) o /k/, /p/, /t/ (no vocalizados).

2.5 Proceso de Comunicación Oral

La comunicación oral se lleva a cabo entre dos personas con el fin de transmitir un mensaje. Este proceso involucra a un emisor, un receptor y un medio de transmisión. El proceso de comunicación oral inicia cuando el emisor formula el mensaje que desea trasmitir al receptor mediante el habla. Una vez que se completa el proceso de producción de habla en el cerebro, se adquieren las características primarias del habla (tono y timbre) en el aparato fonador y se agregan las características secundarias (tipo de sonido de voz) en los órganos articulatorios, se produce la señal acústico-fonética de voz a transmitir. La señal de voz se propaga como ondas a través del medio de trasmisión (el aire) y llegan al oído del receptor, el cual capta las ondas sonoras y las transforma en información que el cerebro sea capaz de interpretar (como el habla o la música).

Cuando llegan las ondas acústicas de la señal de voz al aparato auditivo del receptor comienza el proceso de percepción (reconocimiento del habla). El sonido entra al oído a través del conducto auditivo y llega hasta la membrana timpánica y la hace vibrar. Al vibrar, genera un movimiento en la cadena osicular del oído medio, la cual está compuesta por el martillo, el yunque y el estribo. Los cuales transmiten el sonido hacia el oído interno, donde se localiza

la cóclea. La cóclea se encarga de identificar los diferentes tonos o frecuencias del sonido a través de los distintos grados de sensibilidad que poseen las células sensoriales o ciliadas a lo largo de su estructura espiral, permitiendo al oído percibir todo el espectro audible del sonido y transformar las ondas sonoras en señales eléctricas que se transmiten al cerebro. La transformación de las vibraciones mecánicas en pulsos eléctricos se debe al movimiento de las células sensoriales, las células localizadas en la base de la cóclea o región inferior detectan las frecuencias altas, mientras que las células localizadas en la terminación de la cóclea o región superior detectan las frecuencias bajas. El movimiento de los cilios que se localizan en la superficie de las células ciliadas provoca diferencias de voltaje que producen las señales eléctricas y se transmiten a través del nervio auditivo hasta llegar al cerebro. Por último, el córtex auditivo cerebral interpreta estas señales como información. Este proceso es tan rápido, que permite al ser humano identificar oír el sonido al instante y de forma continua [53].

En el reconocimiento automático del habla se pretende realizar este mismo proceso. Sin embargo, el receptor en lugar de ser una persona es una máquina, la cual tiene que ser capaz de comprender los comandos de voz que se le indican. Para llevar a cabo este procedimiento se le debe de tratar a la señal acústico-fonética de voz para que la maquina sea capaz de comprender el mensaje. Dicho proceso de compresión se explica a detalle en el siguiente capítulo.

Capítulo 3

Reconocimiento de Voz

El Reconocimiento Automático del Habla (RAH) o ASR (por sus siglas en inglés de Automatic Speech Recognition) es una tecnología que le permite al ser humano comunicarse con una computadora de tal manera que sea lo más semejante a una conversación entre humanos. El RAH considera diferentes factores (acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática) y diversos parámetros (fluctuaciones de la voz, el medio ambiente, los medios de captación, el ruido ambiental, etcétera) al analizar la señal de voz.

El reconocimiento de voz data de poco más de 60 años. Sin embargo, los resultados obtenidos hasta la actualidad aún no satisfacen las necesidades humanas. Con el paso del tiempo se incluyen nuevos parámetros de estudio y se convierten en necesarios e indispensables. Los sistemas de reconocimiento por computadora aún distan mucho de ser el sistema perfecto que posee el ser humano, sin embargo, se pretende aproximarlo en gran medida.

El objetivo del reconocimiento del habla es extraer el contenido lingüístico de una locución. Sin embargo, uno de los problemas mas graves que se ha de afrontar en las tecnologías del habla es la variabilidad. Esta variabilidad depende de factores como el estado de animo, la edad, el estado de salud o la existencia de patologías fonatorias y/o fisiológicas.

Este capítulo describe la clasificación del reconocimiento de voz, los factores que afectan el reconocimiento de voz y las etapas de los Sistemas de Reconocimiento Automático del

Habla (SRAH), enfatizando las etapas de caracterización y modelado, las cuales son la parte fundamental de la investigación.

3.1 Reconocimiento del habla y de voz

Las necesidades del ser humano han impulsado a que hoy en día existan una infinidad de aplicaciones con tecnologías del habla, como las ya mencionadas anteriormente; humanoides, casas y oficinas inteligentes, traductores, sistemas de seguridad y en el área automotriz como; manos libres, búsqueda de música por comandos de voz, información audible del vehículo, lectura y dictado de mensajes de texto y en sistemas de navegación, etc. Todas estas aplicaciones utilizan el reconocimiento de voz o en su caso el reconocimiento del habla.

Ambas técnicas utilizan métodos similares de procesamiento de señal de voz en un inicio, pero la diferencia es que el reconocimiento del habla cuando es independiente del hablante (no importa quién es el hablante) ignora las características de la voz del hablante como: si el hablante es hombre o mujer, su estado emocional, la actitud, el tracto vocal, etc., y se centran en los aspectos de las características del habla como: acústica, fonética, fonología, léxicos, sintácticos, semánticos, etc.[54]. Por otro lado, el reconocimiento de voz amplifica las características de la voz que identifican a una persona y suprime las características lingüísticas.

En consecuencia, cuando el reconocimiento del habla es dependiente del hablante, todos los usuarios del sistema deben tener las mismas características de la voz del hablante original, es decir, quien diseño y entreno el sistema, los cuales son más comúnmente empleados en reconocimiento y verificación de locutor. Por tal motivo, es mejor usar sistemas independientes de locutor, así no importará quien entrenó y probó el sistema, en otras palabras, quien creó la base de datos en la cual se encuentra la información de los hablantes conocidos ni quien habla para que el sistema determine si es uno de los hablantes pertenecientes a la base de datos previamente elaborada. Sin embargo, todo dependerá de la aplicación que se le quiera dar.

3.2 Reconocimiento automático de voz

El reconocimiento automático de voz (RAV) o del inglés Automatic Speech Recognition (ASR), es una tecnología que permite a una computadora o dispositivo electrónico identificar las palabras adquiridas mediante un micrófono y convertirlas en texto escrito o en información de interés para ser procesada [55].

El objetivo que se plantea hoy en día la investigación de ASR es permitir a una computadora reconocer en tiempo real y con un 100% de precisión todas las palabras que son inteligiblemente habladas por cualquier persona, independientemente de tamaño del vocabulario, el ruido, las características de los locutores o el acento con que se habla. A pesar de que la tecnología ASR está en el punto donde las máquinas no comprenden todas las conversaciones, en cualquier ambiente acústico o por cualquier persona, es usada día a día en una gran cantidad de aplicaciones y servicios.

Los sistemas comerciales disponibles de RAV generalmente requieren un corto periodo de entrenamiento del locutor y pueden capturar con éxito el habla continua con un amplio vocabulario a un ritmo normal y con una precisión muy alta. Las empresas más comerciales afirman que el software de reconocimiento puede alcanzar tasas de reconocimiento superiores el 90% de precisión si se opera bajo óptimas condiciones (a nivel laboratorio). Dichas condiciones suelen suponer que los usuarios: tienen características de voz que se ajustan a la información del entrenamiento y trabajar en ambientes libres de ruido.

Hay tres razones fundamentales por las cuales se ha desarrollado y se seguirá desarrollando investigación y esfuerzo para resolver el problema de enseñar a las máquinas a reconocer y entender la voz:

i) Accesibilidad para las personas sordas y con problemas de audición

ii) Reducción de costos gracias a la automatización

iii) Compatibilidad de búsqueda de texto

Con la finalidad de generar sistemas más robustos para las tecnologías del habla, la investigación de los sistemas de ASR se ha centrado en tres características; la longitud del vocabulario, las capacidades de la continuidad del habla y la independencia del locutor. El reconocimiento automático de voz se clasifica de la siguiente forma:

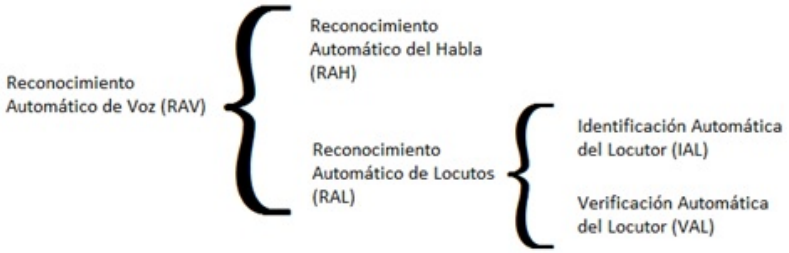


Figura 3.1 Clasificación del reconocimiento automático de voz.

Los sistemas de RAH salmonete consideran el texto (es decir, el que se dice y no el quién y cómo se dice), mientras que los sistemas de RAL además del texto considera la información biométrica de la persona. Para el caso de la IAL se considerar las características biométricas de un grupo de personas y par la VAL las características biométricas que le dan la identidad a una sola persona. Los sistemas de RAH y RAL el vocabulario puede ser dependiente o independiente. Cuando el vocabulario es dependiente el locutor (emisor) debe de articular un mensaje que ya fue determinado con anterioridad, mientras que cuando es independiente el locutor dispone de una teórica libertad de articular cualquier mensaje, ya que en la práctica se utiliza un conjunto finito de palabras o frases. Los sistemas dependientes de vocabulario presentan tasas de reconocimiento más altas que los independientes, debido a que no ha sido posible separar las características del texto textuales del mensaje y las inherentes en cada locutor.

Por otro lado, los sistemas IAL deben de hacer un juicio razonablemente exacto acerca de quién, de una lista de locutores potenciales está hablando. Mientras que el VAL solo se debe determinar si el locutor que está hablando es realmente quien dice ser.

Algunas de las dificultades que se presentan en los sistemas de reconocimiento automático de voz son;

1. **Dependencia o independencia del locutor.**
2. **Variaciones del Locutor:** No siempre la misma persona habla exactamente de la misma forma.
   (a) Velocidad de la fluctuación de la voz.
   (b) Estilo de voz: Depende de la articulación del locutor (claro, espontaneo, entre dientes, mal articulado, sentimental), de su acento, dialecto, de la región donde vive (norte, sur, costa) y del uso de las palabras foráneas.
   (c) Estado emocional: triste, enojado, feliz, exaltado, etc.
3. **Frecuencia de voz:** Niños (frecuencias muy altas), Mujeres (altas frecuencias) y Hombres (frecuencias bajas).
4. **Medio ambiente y ruido de fondo:** depende de la calidad del medio donde se está rea-lizando el reconocimiento de voz, si está libre de ruido acústico de fondo y de reverberaciones.
5. **Patrones lingüísticos:**
   (a) Tamaño de vocabulario: cantidad de palabras con las que fue entrenado el sistema y es capaz de reconocer.
   (b) Ausencia de marcas de frontera en voz continua.
   (c) Ambigüedades inherentes: son ambigüedad acústica que generan un grado de confusión ¨Casa¨ o ¨Caza¨.

3.3 Sistemas de reconocimiento automático de voz

Los sistemas de reconocimiento automático de voz (SRAV) presentan diferentes estructuras y no todos tienen las mismas etapas. Esto depende de la configuración que el investigador crea más conveniente para el desempeño de su sistema y la cual brinde la mejor tasa de reconocimiento según su investigación. Sin embargo, hay etapas que son fundamentales para que sean considerados como sistemas de reconocimiento de voz y además sean capaces de reconocer las palabras dichos sistemas.

El proceso de un SRAV comienza cuando un locutor pronuncia una oración (secuencia de voz y pausas), y a partir de ahí el sistema debe de ser capaz de realizar el proceso de reconocimiento de voz de manera automática. Las etapas fundamentales para el reconocimiento de voz son [56];

- Adquisición de voz
- Pre-procesamiento
- Extracción de características
- Entrenamiento y prueba

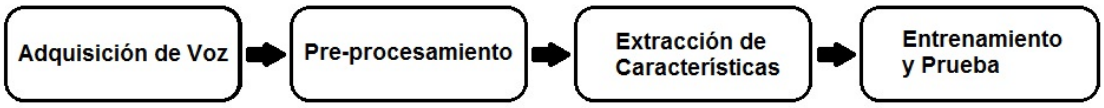


Figura 3.2 Etapas fundamentales para el reconocimiento de voz.

3.3.1 Adquisición de voz

La adquisición de voz es la primera etapa de los SARV en la que las ondas acústicas de la señal de voz son obtenidas mediante un micrófono. Se deben predefinir las características de la señal de entrada; como el canal (mono- o multi-canal), el formato de codificación de la muestra (Lin8, Lin8offset, Lin16, Lin24, Lin24packed, Lin32, Alaw, Mulaw, Float o Double),

la frecuencia de muestreo (8kHz, 11.025kHz, 16kHz, 22.05kHz, 32kHz, 44.1kHz, 48kHz o 96kHz) y el formato del archivo (MP3, Wav, Smp, Snd, AU, AIFF, CSL, etc.) [57].

Otros factores a tomar en cuanta a parte de las características de la señal de voz son; las características del micrófono, las propiedades de la computadora con la que se realizará el procesamiento, el medio ambiente o características acústicas del entorno donde se realiza la adquisición de las señales de voz, el ruido acústico de fondo, que el nivel de energía de la señal de voz obtenida no sobrepase los niveles de energía de saturación (-32768 a 32767 o de -1 a 1 si está normalizada la señal) o que su nivel de energía no se encuentre cercano a cero (o podrá ser confundido con ruido). Es importante definir estas características y factores, ya que de ellos dependerá el procesado que se le dé a la señal de voz en las etapas posteriores. Si no se conocen o se ignoran estos parámetros, la tasa de reconocimiento de voz podría decrecer significativamente.

En esta etapa se debe definir el número de palabras que conformarán el corpus de voces, es decir, la cantidad de palabras adquiridas por el micrófono y almacenadas en la computadora con un nombre o etiqueta diferente para cada palabra. Las palabras que conforman el corpus de voces son utilizadas posteriormente en la etapa de entrenamiento, las cuales el SRAV será capaz de reconocer. De la misma manera que se definen las palabras que conformarán el corpus de voz, se debe definir si el sistema será mono- o multi-locutor.

3.3.2 Pre-procesamiento

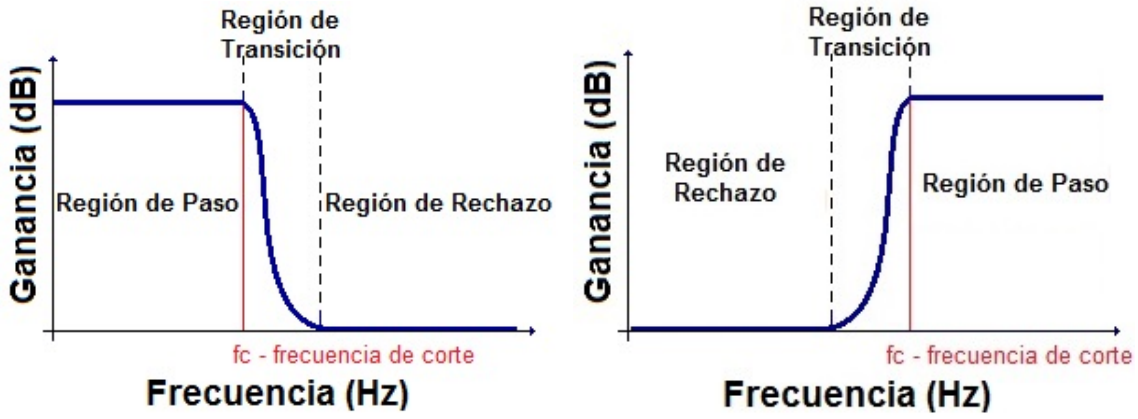
La segunda etapa de los sistemas de reconocimiento de voz es el pre-procesamiento de las señales de voz digitalizadas, las cuales conforman el corpus de voz del sistema. El pre-procesamiento de la señales de voz es considerado una etapa crucial en el desarrollo de un sistema de reconocimiento de voz robusto, permitiendo el mejoramiento, la precisión y la eficiencia de los procesos de extracción de características. Esta etapa incluye típicamente; un filtro pasa bajas y uno pasa altas, la detección de la señal de voz (inicio y fin de palabra), el pre-énfasis de voz, normalización, la segmentación y el ventaneo de la señal [59].

3.3.2.1 Filtro pasa bajas

El filtro pasa bajas en señales de audio es comúnmente llamado filtro de corte agudo. Como ya se había mencionado, la voz solamente contiene información lingüística en el rango de 200Hz a 8kHz. Por lo que es necesario aplicar un filtro pasa bajas con una frecuencia de corte apropiada para eliminar las frecuencias altas de ruido ambiental, la cual es información innecesaria captada durante la etapa de adquisición.

3.3.2.2 Filtro pasa altas:

El filtro pasa altas en señales de audio se conoce como filtro atenuador de bajos. Este filtro se aplica a la señal de voz para eliminar las interferencias innecesarias de baja frecuencia (comúnmente introducidas por el micrófono) e incrementar las altas frecuencias.



(a) Filtro pasa bajas (b) Filtro pasa altas

Figura 3.3 Filtro pasa bajas y pasa altas

Es recomendable aplicar un filtro pasa bajas y otro pasa altas en lugar de un filtro pasa banda, ya que si se aplica un filtro pasa banda la señal de voz se atenúa y afecta al resto del proceso del sistema de reconocimiento de voz.

3.3.2.3 Detección de la señal de voz (inicio y fin de palabra)

Con el fin de obtener una tasa de reconocimiento alta, reducir el gasto computacional y el tiempo de procesamiento, sólo se deben de extraer las características de las señales de voz pura y no los intervalos de silencio o pausas entre palabras. Por lo que es necesario un algoritmo capaz de diferenciar entre lo que es voz y lo que no lo es dentro de una señal de voz.

La detección de actividad de voz (VAD- Voice Activity Detection) también se conoce como detección de actividad del habla o simplemente detección del habla. Es una técnica utilizada en pre-procesamiento del reconocimiento del habla para detectar la presencia o ausencia de voz, es decir, cuándo una palabra inicia y termina. Los segmentos de silencio corresponden a los cierres de las consonantes oclusivas (producidas por la detención de flujo de aire y su posterior liberación), principalmente las consonantes oclusivas sordas (/p/, /t/, /k/) o simplemente al fin de la palabra o frase. En el reconocimiento de voz de palabras aisladas es más sencillo aplicar el algoritmo VAD en comparación con el reconocimiento de voz continua.

El rendimiento de la VAD se evalúa comúnmente por cuatro parámetros [58]:

- FEC (Front End Clipping): recorte al pasar del ruido a la actividad del habla;
- MSC (Mid Speech Clipping): recorte de la palabra debido a ser incorrectamente clasificada como ruido;
- OVER: ruido interpretado como voz en el paso de actividad de voz al ruido;
- NDS (Noise Detected as Speech): ruido interpretado como habla dentro de un período de silencio.

Se debe ser consciente de que probablemente se haya detectado voz como el ruido o el ruido detectado como voz, lo que significa falso positivo (información adicional) y falsos negativos (voz eliminada). La mayor dificultad en la detección del habla en este entorno es la muy baja relación de señal-a-ruido (SNR) en que se encuentra. Es muy complicado distinguir entre el

habla y el ruido utilizando técnicas de detección de nivel de energía simples cuando partes de la expresión verbal están inmersas en el ruido que se produce normalmente.

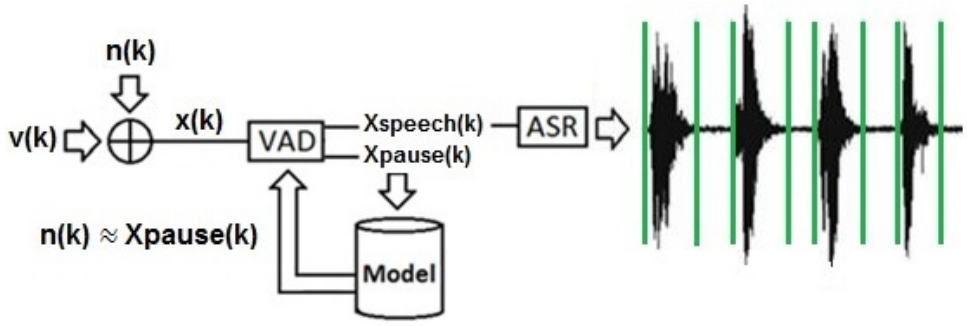


Figura 3.4 Diagrama básico de reconocimiento de voz usando VAD.

En la figura 3.4 se muestra un diagrama básico de reconocimiento de voz usando VAD, el cual está compuesto por un solo canal de entrada (mono-canal), en el que la señal de entrada es una señal de voz $v(k)$ y se adhiere una señal de ruido acústico de fondo $n(k)$, de las cuales se obtiene la señal de voz con ruido $x(k)$. La señal resultante es procesada a través del bloque de VAD con el fin de estimar los segmentos que contienen parte de la señal de voz. El algoritmo de detección de actividad de voz representa los segmentos de voz como $s_{speech}(k)$ y las pausas (ruido acústico de fondo) como $s_{pause}(k)$, todas ellas extraídas de la señal $x(k)$. Posteriormente, los segmentos $s_{speech}(k)$ son transferidas directamente a la etapa de reconocimiento (ASR) , mientras que los segmentos $s_{pause}(k)$ son almacenadas en un "módulo" para ser analizadas nuevamente (la señal $s_{pause}(k)$ permite una adecuada estimación del ruido $n(k)$). El principal reto da la VAD es determinar correctamente el inicio y el final de cada segmento de voz para evitar estimaciones erróneas del ruido que se desea eliminar.

Al añadir un bloque de mejoramiento del habla como en la figura 3.5, se obtiene una estimación previa de la señal del ruido. Con dicha estimación el ruido se puede reducir o eliminar, obteniendo una mejor calidad de la señal de voz $x(k)$ determinada como $\bar{x}(k)$ (señal sin ruido o ruido reducido). Esta señal libre de ruido es procesada con el algoritmo VAD y posteriormente el algoritmo de ASR. Como resultado, el bloque de mejoramiento del habla incrementa la tasa

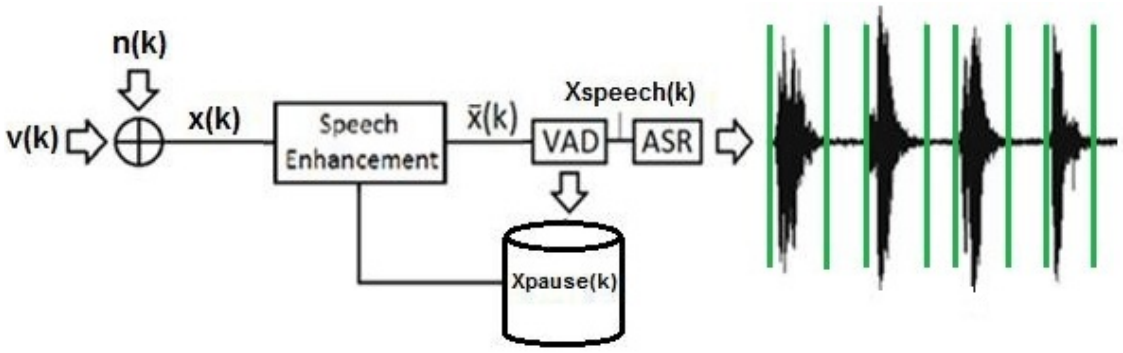


Figura 3.5 Diagrama básico de reconocimiento de voz usando VAD y un bloque de mejoramiento del habla. .

de identificación de inicio y fin de las palabras y a su vez la tasa de reconocimiento del sistema.

Al detectar los segmentos de voz contenidos en la señal de análisis (palabras) no se analizarán segmentos innecesarios que no contienen información de interés y se reducirá el gasto computacional y el tiempo de procesamiento. Sin embargo, si se hace una detección errónea del inicio o fin de palabra (falso inicio o fin de palabra) y se elimina información de la señal de voz, se afectará la tasa de reconocimiento del sistema.

La mayoría de los algoritmos de detección de actividad de voz se basan en la extracción de características temporales de la señal de audio, aplicando una regla de clasificación para identificar que secciones (tramas) de la señal son voz y cuales silencio. Estos algoritmos de VAD se basan en dos propiedades de la señal de voz; la energía y los cruces por cero [60].

La detección de inicio y fin de palabras mediante la propiedad de energía se obtiene a partir del cálculo de energía a corto plazo de la señal de entrada. El valor obtenido se compara con el producto de la energía promedio de la señal de voz y un umbral de energía previamente definido, el cual generalmente se determina durante los periodos de ausencia de voz y de él depende detectar los segmentos de voz correctamente (si la trama es o no vocalizada). El

cálculo de la energía promedio está dado por:

$$E_{average} = \frac{1}{N}\sum_{n=0}^{N-1} s(n)^2, \tag{3.1}$$

donde $s(n)$ es la señal de voz y N el número total de muestras. Y la energía de corto plazo está dada por:

$$E_{frame} = \frac{1}{F}\sum_{n=0}^{F-1} s(n)^2 f(m-n), \tag{3.2}$$

donde $f(m-n)$ es la trama actual a la que se le está calculando la energía y F el tamaño de la trama. Como resultado de la comparación entre la energía de corto plazo y el producto de la energía promedio y el umbral, sólo las tramas de la señal cuyas energías superen el umbral se consideran como segmentos de voz.

Para una señal de voz digital (señal discreta) el cruce por cero ocurre cuando dos muestras consecutivas difieren de signo o cuando una muestra toma el valor de cero. Las señales de alta frecuencia presentan una gran cantidad de cruces por cero al igual que el ruido inmerso en la señal de voz. La densidad de cruces por cero de la señal de voz se calcula con la siguiente ecuación;

$$D_{cc} = \sum_{n=0}^{N-1} |\mathrm{sgn}[x(n)] - \mathrm{sgn}[x(n-1)]|, \tag{3.3}$$

donde N es el número de muestras de la señal y sgn es el cambio de signo y está dado por;

$$\mathrm{sgn}(x) = \begin{cases} +1, & x \geq 0, \\ -1, & x < 0. \end{cases} \tag{3.4}$$

El algoritmo de densidad de cruces por cero analiza los cambios de signo de la señal y los va acumulando. Sin embargo, también se van acumulado los cambios que presenta el ruido de fondo de alta frecuencia. Por eso, es necesario aplicar otro algoritmo que no sólo relacione los cambios de signo, sino que a la vez proporciona información sobre los cambios de energía de la señal.

3.3.2.4 Pre-énfasis

En el análisis de voz, es deseable que un espectro medido pueda tener un rango dinámico similar en toda la banda de frecuencias. Sin embargo, existe una caída de las altas frecuencias de voz generado por el proceso fisiológico del mecanismo de producción del habla (específicamente al salir de la boca); es una tendencia general en su espectro de -6dB/octava a medida que aumenta la frecuencia. Este proceso enfatiza las formantes (frecuencias de resonancia de la cavidad acústica del tracto bocal) de alta frecuencia. Por lo tanto, es necesario introducir un filtro de pre-énfasis para enfatizar las frecuencias formantes y compensar ese caída de -6dB/octava (dando un incremento de +6dB/octava) [59].

El pre-énfasis es diseñado a partir de un filtro digital pasa altas de primer orden que suavizará el espectro de la señal, está descrito por:

$$H(z) = 1 - \alpha z^{-1}. \tag{3.5}$$

El filtro pasa altas también se puede describir por:

$$S_{pp}[n] = S_{bp}[n] - \alpha S_{bp}[n-1], \tag{3.6}$$

donde $S_{pp}[n]$ denota la señal de salida actual del filtro, $S_{bp}[n]$ es la señal de entrada actual y $S_{bp}[n-1]$ es la señal de entrada previa. La constante de suavizado α tiene un valor entre 0.9 y 1. Mediante este filtro se aumenta la magnitud de las muestras, enfatizando la variación entre las muestras adyacentes.

3.3.2.5 Segmentación y Enventanado

Las señales de voz son estocásticas, es decir, señales aleatorias y no estacionarias, cuyo contenido frecuencial y nivel de energía varían en largos periodos de tiempo, esta variación impide analizar la señal. Para poder analizar la señal de voz es necesario que sea estacionaria, por lo que se debe seccionar en segmentos de corta duración (del orden de mili-segundos). A estos segmentos de la señal de voz se les llama "tramas" y debido a su corta duración (típicamente de 10-30ms) y poca variabilidad permiten considerar a la señal de voz como una señal

cuasi-estacionaria. Cada trama se debe traslapar con la trama adyacente, normalmente entre 1/3 o un 1/2 de la longitud de la trama, para generar transiciones suaves entre tramas.

Una vez segmentada la señal de voz y solapadas las tramas se debe aplicar el proceso de enventanado o ventaneo, en el cual elimina los problemas causados por los cambios rápidos de la señal de voz en los extremos de cada trama. Algunos tipos de ventana son: rectangular, Hamming, Hanning, Blackman, Bartlett, Kaiser, entre otras. Sin embargo, la técnica de ventaneo más usada en el reconocimiento de voz es la ventana Hamming (caso especial de la ventana Hanning) cuya amplitud decae gradualmente a un mínimo en cada orilla de la trama como se muestra en la figura 3.6. Esta ventana está representada matemáticamente por:

$$W_{hmm}(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), \qquad 0 \leq n \leq N, \tag{3.7}$$

donde N es el número de muestras de la trama. La ventana se debe multiplicar por cada una de las tramas traslapadas de la señal de voz.

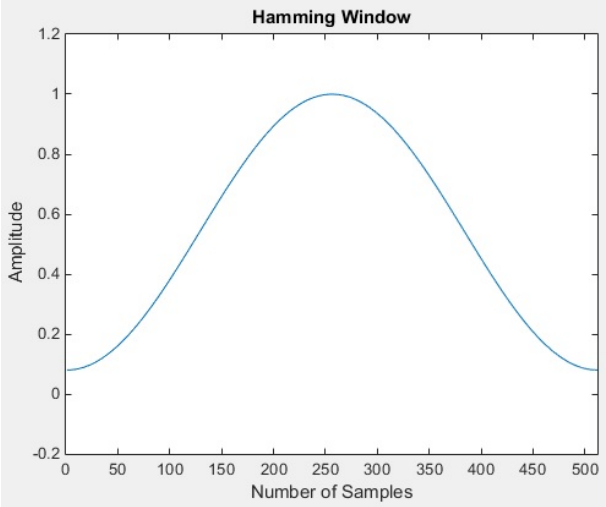


Figura 3.6 Ventana Hamming.

Si las muestras que se encuentran cerca de los extremos de la ventana representan una característica de voz significativa, su baja ponderación impedirá que se caractericen efectivamente en el análisis de la trama. Es por eso que se superponen las dos tramas adyacentes y así

las señales con baja ponderación en una trama pueden ponderarse en la otro trama de forma complementaria [59].

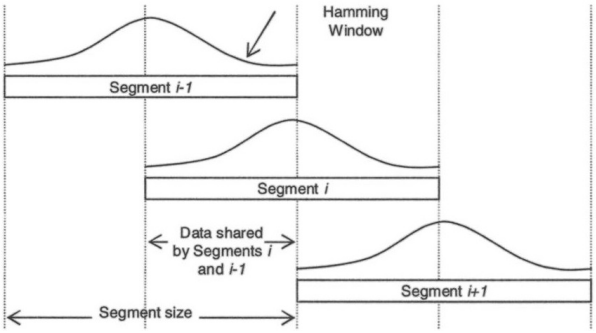


Figura 3.7 Segmentación con un traslape de 50% y ventaneo Hamming [59].

La figura 3.7 muestra tres segmentos de la señal de voz traslapadas y ventaneadas con una ventana Hamming. Se puede observar como las ventanas tienen la misma longitud que las tramas y como al superponer las tramas se comparten la mitad de las muestras de cada trama adyacente.

En la práctica es deseable normalizar la ventana para que la potencia de la señal sea aproximadamente igual a la potencia de la señal antes del ventaneo.

3.3.3 Extracción de características

La tercera etapa de los sistemas de reconocimiento de voz es la extracción de características, en la cual se obtendrá la información contenida en la señal de voz. En esta etapa la señal de voz es transformada en una serie de parámetros que representan la información contenida en la señal de la voz, los cuales serán almacenados en un vector. Entre mayor sea el número de parámetros contenidos en el vector característico de cada segmento de la señal de voz, más complicado es el tratamiento y su implementación.

Se puede decir que el problema fundamental de la parametrización es la elección de un modelo adecuado de la señal de voz. Este modelo tiene como propósito la estimación de la envolvente espectral de la señal, cuya evolución temporal se considera que caracteriza las

diferentes realizaciones acústicas. El modelo elegido para la señal debe ser capaz de estimar una envolvente útil para el sistema de reconocimiento, es decir, robusta a variaciones interlocutor, fenómenos de coarticulación, ruido ambiental, etc. Esta envolvente debe ser además susceptible de ser representada con el número reducido de parámetros, el cálculo de los cuales debe exigir la mínima carga computacional posible.

La información espectral extrae las particularidades del tracto vocal de cada locutor así como su dinámica de articulación. Esta información puede dividirse en dos grupos: información estática e información dinámica. La información estática es la extraída del análisis de cada trama procesada de manera individual. Por el contrario, la información dinámica se extrae del análisis de las tramas de forma conjunta, por lo que es posible recoger el cambio entre las posiciones de articulación.

El presente trabajo se enfoca en dos técnicas de extracción de características de parametrización espectral clásicas que se utilizan hoy en día en los sistemas de reconocimiento de voz; codificación predictiva lineal (LPC Linear Prediction Coding) y coeficientes cepstrales de frecuencias Mel (MFCC Mel-Frequency Cepstral Coefficients).

3.3.3.1 Codificación Predictiva Lineal (LPC)

La predicción lineal juega un papel importante en la tecnología moderna de procesamiento del habla [61]. Los coeficientes de codificación predictiva lineal (LPC) pueden representar de forma eficiente la información de la envolvente espectral de corto tiempo de las señales de voz [62]. Esta técnica de parametrización considera que la señal de voz que se produce no ocurre de forma aleatoria por lo que existe una relación entre cada una de las muestras de voz.

El método de LPC se basa en que cada muestra de habla puede predecirse o representarse mediante una combinación lineal de varias muestras pasadas. Es decir, que dada una muestra de voz $s(n)$ en un tiempo n, puede ser aproximada como una combinación lineal de las muestras

de voz anteriores :

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \cdots + a_p s(n-p), \tag{3.8}$$

que es un modelo autoregresivo (AR) de la señal de voz, donde p es el orden de predicción y $a_1, a_2, \cdots, a_p$ son los coeficientes de predicción que se asumen constantes en la trama de voz que se analiza. El orden de predicción se elige a partir de varias características, las más importantes son la frecuencia de muestreo y la longitud del tracto bocal [64]. Al convertir la ecuación 3.8 en una igualdad se debe agregar un error de predicción $e(n)$:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n), \tag{3.9}$$

el cual se calcula de la siguiente manera

$$e(n) = s(n) - \tilde{s}(n), \tag{3.10}$$

donde $\tilde{s}(n)$ es la predicción lineal (la estimación de la combinación lineal de las muestras pasadas) y está dada por:

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k). \tag{3.11}$$

Como las características espectrales de la voz varían con el tiempo, los coeficientes de predicción $a_k (k = 1, 2, \cdots, p)$ en un tiempo n, deben ser estimados a partir de un segmento corto de la señal de voz:

$$s_n(m) = s_n(n+m). \tag{3.12}$$

Por lo que se debe segmentar, traslapar y ventanear la señal de voz (previamente realizado en la etapa de pre-procesamiento) y a partir de estos segmentos (tramas) obtener el conjunto de coeficientes de predicción.

El problema básico del análisis de predicción lineal es determinar el conjunto de coeficientes de predicción a_k directamente de la señal de voz, los cuales minimizarán el error cuadrático medio en el intervalo corto de tiempo. Al minimizar la suma del error cuadrático entre las muestras reales y las predichas en el intervalo finito, se encuentra el conjunto único de

coeficientes llamados coeficientes de predicción a_k. El error cuadrático medio en una muestra n se define por:

$$E_n = e_n^2(m). \tag{3.13}$$

Sustituyendo en 3.13 la ecuacíon 3.10 se obtiene:

$$E_n = \sum_m \left(s_n(m) - \tilde{s}_n(m)\right)^2 = \sum_m \left[s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k)\right]^2. \tag{3.14}$$

Para resolver la ecuación 3.14 y obtener el mínimo error de predicción, se debe derivar E_n con respecto a cada a_k e igualar el resultado a cero:

$$\frac{\partial}{\partial a_k} E_n = \frac{\partial}{\partial a_k}\left[s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k)\right]^2 = 0, \tag{3.15}$$

resolviendo la ecuación 3.15 se obtiene:

$$\sum_m s_n(m-i)s_n(m) - \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m-i)s_n(m-k), \tag{3.16}$$

donde el termino $\sum_m s_n(m-i)s_n(m-k)$ corresponde a la covarianza de $s_n(m)$:

$$\phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k). \tag{3.17}$$

Sustituyendo 3.17 en 3.16 se obtiene:

$$\phi_n(i,0) = \sum_{k=1}^{p} \hat{a}_k \phi_n(i,k), \tag{3.18}$$

la cual describe un conjunto de p ecuaciones con p incógnitas las cuales son los coeficientes de predicción a_k y son conocidas como las ecuaciones de Yule-Walker. De 3.18 se puede expresar el error cuadrático medio como:

$$E_n = \phi_n(0,0) - \sum_{k=1}^{p} a_k \phi_n(0,k). \tag{3.19}$$

Para obtener los coeficientes de predicción a_k a partir de 3.19 se debe calcular $\phi_n(i,k)$ para $[1 \leq i \leq p]$ y $[1 \leq k \leq p]$ y resolver el conjunto de p ecuaciones resultantes.

Los algoritmos comúnmente usados para calcular los coeficientes de predicción a_k son; el método de autocorrelación, el método de covarianza y el método de enrejado (lattice). Según el análisis teórico y los resultados experimentales de [66] el método de autocorrelación con la cantidad de operación mínima es apto en sistemas de tiempo real pero no puede dar una alta precisión de estimación, el método de covarianza da la mayor precisión pero la cantidad de cálculos incrementa rápidamente a medida que aumenta el orden de predicción, mientras que el método de enrejado presenta la ventaja de obtener los coeficientes directamente de las muestras de voz sin recurrir a una función de correlación. En señales de voz es común usar el método de autocorrelación, debido a que se puede calcular mediante algoritmos computacionales eficientes y a que los filtros obtenidos son estables. En este trabajo se definió la autocorrelación como el método para calcular los coeficientes de predicción. El esquema básico para el cálculo de LPC se representa en el siguiente diagrama [65]:

Figura 3.8 Diagrama de bloques para el cálculo de LPC.

La figura 3.8 muestra las etapas de LPC para obtener los coeficientes de predicción lineal. Se puede observar que las primeras dos etapas; pre-énfasis y la de segmentación y ventaneo, se consideraron parte de la etapa de pre-procesamiento de la señal de voz.

Para desarrollar el método de autocorrelación se debe segmentar la señal de voz en tramas de "N" muestras, por lo que el intervalo de evaluación irá de 0 a $N-1$ y se asume que los valores fuera de este rango son iguales a cero. Esto es equivalente a asumir que la señal de voz, $s(m+n)$, es multiplicada por una ventana de longitud finita, $w(m)$, la cual es cero fuera del rango definido.

$$s_n(m) = s(m+n)w(n), \qquad 0 \leq m \leq N-1. \tag{3.20}$$

La función de covarianza de la señal enventanada se expresa de la siguiente forma:

$$\phi_n(i,k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k), \qquad 1 \leq i \leq p, \quad 1 \leq k \leq p. \tag{3.21}$$

Dado que sólo es función de las variables i y k, es decir, sólo depende de la ubicación temporal de la muestra, la función de covarianza se reduce a una simple función de autocorrelación:

$$\phi_n(i,k) = r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k). \tag{3.22}$$

Como la función de autocorrelación es simétrica para señales no complejas como la señal de voz $r_n(-k) = r_n(k)$, las ecuaciones de predicción lineal quedan:

$$r(i) = \sum_{k=1}^{p} a_k r(i-k), \qquad i = 1, 2, \cdots, p, \tag{3.23}$$

donde $r(\tau) = E[s(n)s(n-\tau)]$ es la función de autocorrelación de $s(n)$. Dado que el valor real de $r(\tau)$ no se puede obtener, la señal de voz de entrada debe ser ventaneada y $r(\tau)$ calculada de la siguiente manera:

$$r(\tau) = \sum_{k=1}^{N} s(n)s(n-\tau). \tag{3.24}$$

La función de autocorrelación es simétrica, es decir, $r(\tau) = r(-\tau)$. Se puede representar apartir de la ecuación 3.23 de la forma:

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(p) \\ r(1) & r(0) & r(1) & \cdots & r(p-1) \\ r(2) & r(1) & r(0) & \cdots & r(p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(p) & r(p-1) & r(p-2) & ... & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ \vdots \\ r_p \end{bmatrix}, \tag{3.25}$$

$$R\overrightarrow{a} = \overrightarrow{r}, \tag{3.26}$$

donde R es la matriz de autocorrelación y es una matriz de tipo Toeplitz (simétrica con todos los elementos de la diagonal) la cual puede resolverse eficientemente mediante varios procedimientos conocidos, $\overrightarrow{a}$ el vector de coeficientes de predicción y $\overrightarrow{r}$ el vector de autocorrelciones. Cada una de las tramas se debe de autocorrelacionar para saber si presentan periodicidad y obtener los coeficientes de predicción lineal de cada una de ellas.

El siguiente paso del procesamiento es el análisis LPC, que convierte cada trama de p+1 autocorrelaciones en un conjunto de parámetros LPC, en el que el conjunto representa los coeficientes LPC. El método formal para convertir los coeficientes de autocorrelación en un conjunto de parámetros LPC (para el método LPC de autocorrelación) es conocido como método de Levinson-Durbin y está dado por el siguiente algoritmo [8]:

$$E^{(0)} = r(0), \tag{3.27}$$

$$k_i = \left\{ r(i) - \sum_{j=1}^{L-1} a_j^{(i-1)} r(|i-j|) \right\} / E^{(i-1)}, \qquad 1 \leq i \leq p, \tag{3.28}$$

$$a_i^{(i)} = k_i, \tag{3.29}$$

$$a_i^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}, \qquad 1 \leq j \leq i-1, \tag{3.30}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)}, \tag{3.31}$$

donde E representa el error de predicción, k_i los coeficientes de reflexión entre las partes consecutivas del tubo acústico y a_i los coeficientes de predicción lineal. El algoritmo de Levinson-Durbin se utiliza para resolver de forma recursiva las ecuaciones lineales de la matriz Toeplitz de autocorrelaciones y obtener los coeficientes de predicción.

3.3.3.2 Coeficientes Cepstrales de Frecuencias Mel (MFCC)

Los Coeficientes Cepstrales de Frecuencias Mel es la técnica de extracción de características más usada en reconocimiento del habla. Fue introducida por Davis y Mermelstein en la década de 1980 [67], y desde entonces ha sido parte del estado del arte. La palabra cepstales se deriva de "cepstrum" la cual es un anagrama de la palabra spectrum.

Los sonidos generados por un ser humano se filtran por la forma del tracto vocal, incluyendo la lengua, los dientes, labios, etc. Esta forma determina qué sonido se produce. Si se determinara la forma con precisión, esto proporcionaría una representación precisa del fonema que se produce. La forma del tracto vocal se manifiesta en la envolvente del espectro

de potencia de corto tiempo, por lo que los MFCC tratan de representar con precisión esta envolvente.

El objetivo de esta técnica de extracción de características es obtener una representación de la amplitud del espectro del habla de manera compacta, robusta y apropiada para caracterizar la señal de voz. Se basa en los criterios de percepción del sistema auditivo del ser humano, en la variación de los anchos de banda de las frecuencias críticas del oído. El esquema básico para la extracción de los MFCC se representa en el siguiente diagrama [70]:

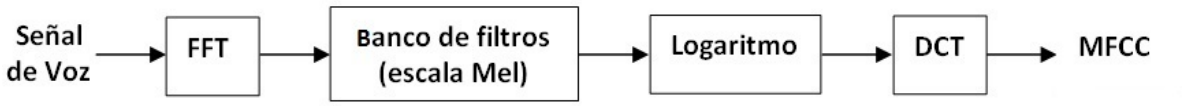


Figura 3.9 Diagrama de bloques para el cálculo de los MFCC.

El primer bloque calcula la transformada de Fourier de tiempo corto (STFT - Short time Fourier Transform) a cada una de las tramas obtenidas de la etapa de pre-procesamiento, está dada por:

$$STFTx[n] \equiv X(m,w) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-jwn}, \tag{3.32}$$

donde $x[n]$ es la señal de voz y $w[n]$ la ventana. Sin embrago, al momento de codificar se hace uso de la Tranasformada Rápida de Fourier (FFT - Fast Fourier Transform) y no la STFT. Al transformar la señal de voz del dominio temporal al frecuencial mediante la FFT se calcula la magnitud y la densidad espectral de potencia de cada una de las tramas de la señal de voz para identificar qué frecuencias están presentes en cada trama. La densidad espectral de potencia está definida por:

$$P(n) = |x(n)|^2. \tag{3.33}$$

La densidad espectral de potencia aún contiene información innecesaria para el reconocimiento automático de voz. Este efecto incrementa a medida que aumentan las frecuencias. Para ello se toman agrupaciones de las frecuencias de la densidad espectral de potencia y se suman para tener una idea de cuánta energía existe en varias regiones de frecuencia. Esto se realiza mediante el banco de filtros Mel, estos filtros están distribuidos en escala Mel, la cual es

una escala musical perceptual de tonos juzgados por los oyentes para ser equiespaciados uno de otro, fue propuesta por S. Stevens, J. Volkmann y E. Newman en 1937 [68].

En el bloque de banco de filtros, la densidad espectral de potencia es ponderada por una serie de filtros distribuidos en escala de frecuencias Mel. Estos filtros calculan el promedio del espectro alrededor de cada frecuencia central y están espaciados linealmente para frecuencias menores a 1000 Hz y logarítmicamente para frecuencias mayores de 1000 Hz. Se encuentran espaciados de esta forma con el fin de capturar las características fonéticamente importantes del habla. Los filtros que conforman el banco pueden ser de forma triangular, Hanning, Hamming o Kaiser, pero el filtro triangular pasa banda es el más utilizado. Cada uno de los filtros corresponde una banda crítica del oído humano.

El banco de filtros se encuentra limitado por un rango de frecuencias, el cual se define a partir de una frecuencia mínima y una frecuencia máxima que no puede ser mayor a la mitad de la frecuencia de muestreo. Las frecuencias que limitan el banco de filtros están expresadas en Hertz y deben ser transformadas a frecuencias de escala Mel, se calculan a partir de:

$$Mel(f) = 2595log\left(1 + \frac{f}{700}\right). \tag{3.34}$$

Las frecuencias centrales que conforman el banco de filtros deben ser calculadas en escala Mel, ya que están espaciadas linealmente dentro del rango de frecuencias del banco de filtros. Posteriormente se transforman las frecuencias centrales de escala Mel obtenidas a frecuencias lineales a partir de:

$$Mel^{-1}(f) = 700(exp^{Mel/1125} - 1). \tag{3.35}$$

El primer filtro que se obtiene es muy estrecho e indica cuánta energía existe cerca de 0 Hz. A medida que las frecuencias aumentan, los filtros se amplían y las variaciones son menores.

La respuesta no lineal del oído humano a los componentes de frecuencia en el espectro de audio permite que seamos capaces de distinguir mucho mejor los cambios en el tono a bajas frecuencias (<1 kHz) que en el extremo superior del espectro audible (altas frecuencias), es

decir, en un análisis de frecuencia no lineal realizado por el oído humano cuanto mayor sea la frecuencia menor será su resolución. La escala Mel hace que las características coincidan más estrechamente con lo que los humanos escuchan.

El banco de filtros se calcula con las siguientes ecuaciones:

$$B(m,k) = \begin{cases} 0 & si \quad k > f(m-1), \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & si \quad f(m-1) \leq k \leq f(m), \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & si \quad f(m) \leq k \leq f(m+1), \\ 0 & si \quad k > f(m+1), \end{cases} \tag{3.36}$$

donde $B(m,k)$ es la matriz del banco de filtros cuyas filas m representan el número de filtros del banco y las columnas k el número de ventanas de análisis. Resultando un banco de filtros por cada ventana de la señal de voz. Para conocer la energía de los bancos de filtros se debe multiplicar cada banco de filtros con las ventanas de densidad espectral de potencia (obtenida en el bloque de transformada de Fourier) y posteriormente sumar los coeficientes.

$$E(m,k) = \sum_{m=1}^{M} B(m,k)P(k) \qquad k = 1, 2, \cdots, K, \tag{3.37}$$

donde $P(k)$ es la densidad espectral de potencia y k representa el número de la ventana que va desde la primera hasta la K-ésima ventana de la señal de voz.

En el tercer bloque se calcula el logaritmo de las energías de los bancos de filtros obtenidas en el bloque anterior. Esta operación hace que las características obtenidas coincidan más estrechamente con lo que los humanos realmente escuchan. Es decir, el logaritmo aproxima la relación entre la percepción humana del volumen y la intensidad del sonido. El oído humano no escucha sonoridad en escala lineal, pero si es percibida de la señal aproximadamente logarítmica.

$$E_{log}(m,k) = log\left(\sum_{m=1}^{M} B(m,k)P(k)\right) \qquad k = 1, 2, \cdots, K. \tag{3.38}$$

Generalmente, para duplicar el volumen percibido de un sonido, se necesita poner ocho veces más energía en él. Esto significa que grandes variaciones en la energía pueden no sonar

tan diferentes si el sonido es alto al comenzar.

El cuarto bloque es la trasformación al dominio temporal, debido a que los coeficientes espectrales Mel son números reales, se pueden convertir al dominio temporal mediante la transformada de coseno discreta (DCT - Discrete Cosine Transform). Se hace uso de la DCT en lugar de la transformada discreta de Fourier inversa (IFDT - Inverse Discrete Fourier Transform) para disminuir el gasto computacional [72]. Solamente la parte real, es decir, la magnitud contiene información de interés y está relacionada con la función coseno, mientras que la función seno representa la parte imaginaria que se refiere a la fase, esta no es información de interés para el reconocimiento de voz y por lo tanto al hacer uso de la DCT se evitaría hacer cálculos innecesarios.

En este último bloque se calcula la DCT del logaritmo de las energías del banco de filtros. Debido a que los bancos de filtros se superponen, las energías del banco de filtros están altemente correlacionadas entre si y al realizar la DCT se decorrelacionan las energías. Al final, sólo algunos de los coeficientes DCT se mantienen, debido a que los coeficientes de DCT más altos representan cambios rápidos en las energías del banco de filtros y estos cambios rápidos degradan el rendimiento del sistema de reconocimiento de voz, por lo que se obtiene una pequeña mejora al eliminarlos. Como resultado de la DCT del logaritmo real del espectro de energía se obtienen los coeficientes cepstrales de frecuancias Mel [73].

Los coeficientes de orden superior representan la información de excitación, o la periodicidad en la forma de onda, mientras que los coeficientes cepstrales de orden inferior representan la forma del tracto vocal o la forma espectral lisa [69]. La DCT se puede definir como:

$$MFCC(n) = \sum_{m=1}^{M} E_{log}(m,k) cos\left[n\left(m-\frac{1}{2}\right)\frac{\pi}{M}\right], \qquad m = 1, 2, \cdots, M, \tag{3.39}$$

donde M representa el número es el número total de coeficientes MFCC que varía con respecto a $n = 1, 2, \cdots, M$, m representa el número de filtros del banco, k el número de la ventana de análisis.

3.3.4 Reconocimiento de Voz

El reconocimiento de voz es la última etapa de un SRAV. Es la etapa en la que se lleva a cabo el entrenamiento de los vectores característicos de las palabras del corpus de voz (se obtuvieron en la extracción de características mediante las técnicas de LPC y MFCC) y la prueba o reconocimiento de la señal de voz de entrada. La técnica de reconocimiento de voz que se aplicó en este trabajo fue la de modelos ocultos de Markov o mejor conocida como Hidden Markov Models (HMM).

Los modelos ocultos de Markov es una técnica de modelado estadístico que fue implementada por primera vez en reconocimiento del habla en la decada de 1970 y siguen siendo implementados hasta hoy en día en sistemas de reconocimiento de voz. Esta técnica se basa en que la señal de voz se puede caracterizar como un proceso estocástico paramétrico y que los parámetros del proceso pueden ser estimados de manera precisa y definida [8].

Los HMM constan de dos procesos estocásticos anidados, donde cada vector característico (observación) es a su vez una función estocástica de cada estado del modelo. La función estocástica subyacente no es directamente observable, es decir, está oculta y sólo puede observarse a través de otro conjunto de procesos estocásticos que producen la secuencia de observación O_t en un instante t. Un modelo HMM tiene estados finitos y cada estado tiene asociada una Función de Densidad de Probabilidad (FDP) para cada vector característico.

Los parámetros que definen un HMM son:

- **Número de estados N del modelo.** El conjunto de todos los estados posibles se denota por: $S = \{S_1, S_2, \cdots, S_N\}$.
- **Número de símbolos de observaciones M por cada estado.** Corresponde a las salidas físicas del sistema a modelar (coeficientes) y se denota por: $V = \{V_1, V_2, \cdots, V_M\}$.
- **La matriz de probabilidad de transición de estados ($A = [a_{ij}]$).** Indica la probabilidad de que estando el sistema en un estado, en un instante determinado, el mismo evolucione

a otro estado o permanezca en el mismo estado, en el instante subsiguiente. Se denota por:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,j} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,j} \end{bmatrix} = P(q_{t+1} = S_j / q_t = S_i), \qquad 1 \leq i, j \leq N. \quad (3.40)$$

- **La distribución de probabilidades de observación de un símbolo en el estado ($B = [b_j(k)]$).** Se pueden definir como un vector B de tantos elementos como símbolos posibles M:

$$B = [b_1(k), b_2(k), \cdots, b_j(k),] = P(O_t = V_k / q_t = S_i) \qquad 1 \leq k \leq M. \quad (3.41)$$

- **La distribución inicial de probabilidades de cada estado ($\pi = [\pi_i]$).**

$$\pi = [\pi_1, \pi_2, \cdots, \pi_i] = P(q_i = S_i) \qquad 1 \leq i \leq N. \quad (3.42)$$

Una manera compacta de representar un modelo HMM es:

$$\lambda = (A, B, \pi). \quad (3.43)$$

Para la transición de estados existen tres topologías:

- *Modelo ergódico*; también conocido como completamente conectado, este modelo permite llegar (en un solo paso) a todos los estados del modelo desde cualquier otro estado del modelo. Como se muestra en la figura 3.10(a).

- *Modelo left-right*; es llamado así debido a que la secuencia del estado subyacente asociado con el modelo tiene la propiedad de que a medida que aumenta el tiempo, el índice de estado aumenta (o se mantiene igual). No permite transiciones a estados cuyo indice sea menor al estado actual. Por lo tanto, el estado inicial debe ser el estado uno y el último el estado N. Adicionalmente, se aplican limitaciones a los coeficientes de transición del estado para garantizar que no se produzcan grandes cambios en los índices de

estado. Es decir, que las transiciones de estados no superen un número definido en la configuración del modelo (normalmente 2 estados como se muestra en la figura 3.10(b)). No se permiten transiciones de estados mayores al número limite definido. Es uno de los modelos más usados en el reconocimiento de voz ya que restringe la transición a estados anteriores aprovechando las propiedades que presenta la señal de voz. Este modelo tiene un caso particular llamado "modelo paralelo"

- *Modelo paralelo*; se llama así, debido a que se encuentran dos modelos left-right en paralelo. Por lo que sigue las reglas y propiedades de dicho modelo. Se puede ver que tiene cierta flexibilidad no presente en un modelo left-right estricto.

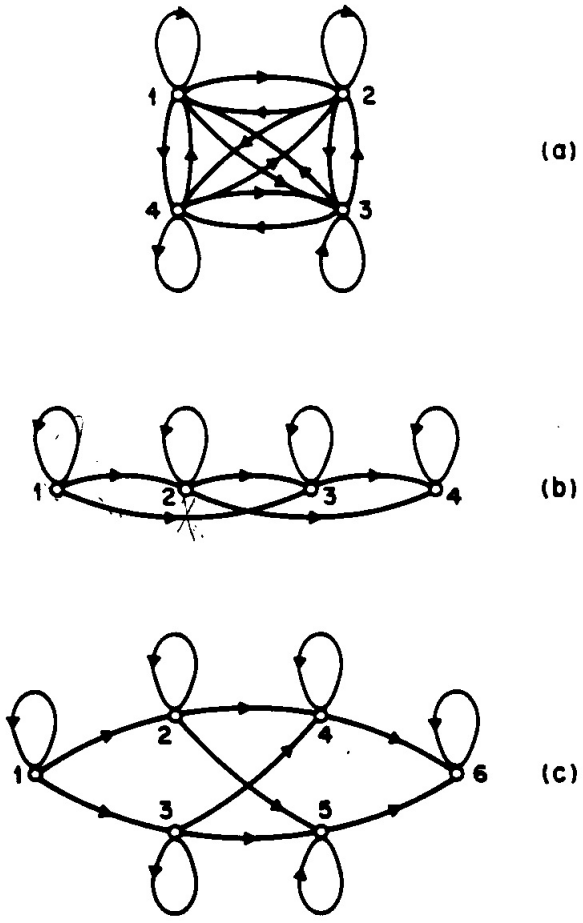


Figura 3.10 Modelos de HMM. (a) Modelo ergódico de 4 estados. (b) Modelo left-right de 4 estados. (c) Modelo paralelo de 6 estados. [75]

Los sistemas de reconocimiento de voz basados en HMM son creados y evaluados por vectores de coeficientes y constan principalmente de tres etapas: inicialización del modelo, entrenamiento y reconocimiento. Primeramente se debe definir un vector de probabilidades para iniciar en un estado cualquiera. En la etapa de entrenamiento, se crean los modelos de referencia a partir de los vectores característicos de las repeticiones de las palabras que conforman el corpus de voz (entre más repeticiones haya mejor será el modelo de la palabra). En la etapa de reconocimiento o prueba, se realiza la comparación entre los modelos entrenados y la muestra a identificar (palabra), y se define que palabra fue reconocida a partir del cálculo de la probabilidad más alta entre las palabras que conforman el corpus de voz y la señal de entrada.

Existen tres problemas básicos que se deben resolver al aplicar los modelos HMM:

- **Problema 1 (Evaluación):** Dada una secuencia de observaciones $O = \{O_1, O_2, \cdots, O_T\}$ y un modelo $\lambda = (A, B, \pi)$. ¿Cómo calcular eficientemente la probabilidad de producir una secuencia de observación a partir del modelo? - Es decir, que la secuencia O haya sido producida por el modelo λ. Para resolver este problema se hace uso del procedimiento adelanto-atraso o mejor conocido como algoritmo forward-backward. El cual calcula todas las secuencias de estados posibles con una longitud de estados, igual a la cantidad de observaciones T.

  Si se considera la secuencia de estados fija:

$$Q = \{q_1, q_2, \cdots, q_T\}, \tag{3.44}$$

  donde q_1 es el estado inicial. La probabilidad de la secuencia de observación O para la secuencia de estados 3.44 es:

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T), \tag{3.45}$$

  asumiendo la independencia estadística de las observaciones. La probabilidad de tal secuencia de estado Q se puede expresar como:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}. \tag{3.46}$$

La probabilidad de tener una secuencia O dado λ se obtiene por la suma de las probabilidades conjuntas (ocurren de forma simultanea) sobre todas las secuencias de estados posibles:

$$\begin{aligned} P(O|\lambda) &= \sum_{\forall Q} P(O|Q,\lambda)P(Q,\lambda), \\ &= \sum_{q_1,q_2,\cdots,q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T). \end{aligned} \tag{3.47}$$

El cálculo de esta ecuación se interpreta de la siguiente manera; Inicialmente (cuando el tiempo $t = 1$), se encuentra en el estado q_1 con la probabilidad π_{q_1} y genera el símbolo O_1 (en ese estado) con probabilidad $b_{q_1}(O_1)$. Cuando el reloj cambia de t a $t + 1$ ($t = 2$) se hace una transición al estado q_2 dese el estado q_1 con probabilidad $a_{q_1 q_2}$ y genera el símbolo O_2 con probabilidad $b_{q_2}(O_2)$. Así continua el proceso hasta la ultima transición de estado (en el tiempo T) del estado q_{T-1} al estado q_T con la probabilidad $a_{q_{T-1}}$ y se genera el símbolo O_T con probabilidad $b_{q_T}(O_T)$.

El algoritmo de forward-backward define la variable de avance $\alpha_t(i)$ como:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i|\lambda), \tag{3.48}$$

donde $\alpha_t(i)$ es la probabilidad de que se generen las observaciones $O_1 O_2 \cdots O_t$ con el estado S_j en el tiempo t dado λ. Y se resuelve $\alpha_t(j)$ de la siguiente manera:

1. Inicio:

$$\alpha_t(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N. \tag{3.49}$$

   Inicializa las probabilidades hacia adelante como la probabilidad conjunta de un estado S_j en el tiempo t dado λ.

2. Inducción:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \leq t \leq T-1, \qquad 1 \leq j \leq N. \tag{3.50}$$

   Es el paso principal del algoritmo, en el cual el estado S_j puede ser alcanzado en el tiempo $t + 1$ desde cualquier estado N posible.

3. Terminación:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{3.51}$$

Se obtiene el cálculo deseado de $P(O|\lambda)$ como la suma de las variables de avance terminal $\alpha_t(j)$, es solo la suma de los $\alpha_t(j)$'s. Es la probabilidad que se obtiene del algoritmo forward-backward dadas las secuencias de observación O y el modelo λ, para resolver el problema uno (evaluación).

- **Problema 2 (Decodificación):** Dada una secuencia de observaciones $O = O_1, O_2, \cdots, O_T$ y un modelo $\lambda = (A, B, \pi)$ ¿Cómo se puede elegir la secuencia de estados correspondientes $Q = \{q_1, q_2, \cdots, q_T\}$ que sea óptima en algún sentido, es decir, que explique mejor las observaciones? - El problema es descubrir la parte oculta del modelo, de hallar la secuencia o transición "correcta" de estados ocultos. Debido a que la secuencia "correcta" no existe, se utiliza un criterio de optimización para resolver el problema de la mejor manera posible. Para resolver este problema, existen algoritmos como el de Viterbi o el de Máxima expectación. Siendo el algoritmo de Viterbi, el cual es una técnica de programación dinámica que trata de obtener la mejor secuencia de estados que maximice la probabilidad conjunta $P(Q|O, \lambda)$. Según el algoritmo de Viterbi, para encontrar la mejor secuencia de estados $Q = \{q_1, q_2, \cdots, q_T\}$ para la secuencia de observación $O = O_1, O_2, \cdots, O_T$ dada, se debe definir la variable:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t|\lambda]. \tag{3.52}$$

La cual es la máxima probabilidad a lo largo de un único camino en el tiempo t, que ocurre en las primeras t observaciones y termina en el estado S_i. Por lo que se tiene:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}). \tag{3.53}$$

Para recuperar realmente la secuencia del estado, se necesita rastrear el argumento que maximice la probabilidad 3.53, para cada t y j mediante el arreglo $\psi_t(j)$. El procedimiento completo para encontrar la mejor secuencia del estado se define en los siguientes pasos:

1. Inicio:

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N, \tag{3.54}$$

$$\psi_1(i) = 0. \tag{3.55}$$

2. Recursión:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \qquad 2 \leq t \leq T, \qquad 1 \leq j \leq N, \tag{3.56}$$

$$\psi_t(j) = \arg\max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \qquad 2 \leq t \leq T, \qquad 1 \leq j \leq N. \tag{3.57}$$

3. Fin:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)], \tag{3.58}$$

$$q_T^* = \arg\max_{1 \leq i \leq N} [\delta_T(i)]. \tag{3.59}$$

4. Trayectoria en retroceso (secuencia de estados):

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \cdots 1. \tag{3.60}$$

A partir de esta ecuación se encuentra la transición óptima de estados ocultos que resuelve el problema dos (decodificación).

- **Problema 3 (aprendizaje):** ¿Cómo se puede ajustar los parámetros del modelo λ para maximizar la probabilidad $P(O, \lambda)$? - Se refiere al ajuste adecuado de los parámetros del modelo λ para satisfacer algún criterio de optimización. No hay forma analítica de encontrar el conjunto de parámetros de λ que maximice la probabilidad de generar una secuencia de observaciones de forma cerrada. Sin embargo, es posible elegir los parámetros de λ de tal forma que la probabilidad $P(O, \lambda)$ pueda ser maximizada de manera local y no global mediante un proceso iterativo como el del algoritmo Baum-Welch.

Con el fin de describir el proceso para reestimar los parámetros del modelo HMM, se define $\xi_t(i, j)$ como la probabilidad de estar en el estado S_i en el tiempo t y en el estado S_j en el tiempo $t + 1$ dado el modelo λ y la secuencia de observación O:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | 0, \lambda). \tag{3.61}$$

A partir de la variable de avance 3.48 del algoritmo forward-backward y de la variable de retroceso:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \cdots, O_T | q_t = S_i, \lambda). \tag{3.62}$$

Se puede definir $\xi_t(i,j)$ en términos de estas dos variables de la siguiente manera:

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}, \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}, \end{aligned} \tag{3.63}$$

donde el numerador corresponde a la probabilidad de que el estado oculto actual S_i y el estado oculto siguiente S_j sean igual al cambio de estados, dado el modelo λ y la secuencia de observación O, al dividirlo por $P(O|\lambda)$ se obtiene la medida de probabilidad deseada. Si se define a $\gamma_t(i)$ como la probabilidad del estado S_i en el tiempo t dados λ y O:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda), \tag{3.64}$$

se puede relacionar a $\gamma_t(i)$ con $\xi_t(i,j)$:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), \tag{3.65}$$

y si se suma $\gamma_t(i)$ sobre el índice de tiempo t, se obtiene una cantidad que se puede interpretar como el número esperado de veces que se ha transitado por el estado S_i, o de forma equivalente, el número esperado de transiciones realizadas desde el estado S_i (si se excluye el término $t = T$ de la suma).

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{No. esperado de transiciones desde el estado } S_i. \tag{3.66}$$

De manera similar, la suma de $\xi_t(i,j)$ sobre el índice de tiempo t (desde $t = 1$ hasta $t = T - 1$) se puede interpretar como el número esperado de transiciones del estado S_i al estado S_j.

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{No. esperado de transiciones del estado } S_i \text{ al estado } S_j. \tag{3.67}$$

Usando las ecuaciones anteriores se puede establecer un método para reestimar los parámetros de un HMM. Las formulas para reestimar π, A y B son:

$$\overline{\pi}_i = \gamma_1(i), \tag{3.68}$$

donde $\overline{\pi}_i$ es el número esperado de veces en el estado S_i en el tiempo $t = 1$,

$$\overline{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{3.69}$$

$\overline{a}_{i,j}$ es el resultado de dividir el número esperado de transiciones del estado S_i al estado S_j entre el número esperado de transiciones desde el estado S_i

$$\overline{b}_j(k) = \frac{\sum_{t=1}^{T-1} \gamma_t(i)}{\gamma_t(i)}, \tag{3.70}$$

y $\overline{b}_j(k)$ es el resultado de dividir el número esperado de veces en el estado j y observar el símbolo v_k entre el número esperado de veces en el estado j.

Estás formulas de reestimación se pueden obtener directamente de la maximizando la función auxiliar de Baum:

$$Q(\lambda, \overline{\lambda}) = \sum_{Q} P(Q|O,\lambda) log[P(O,Q|\lambda)], \tag{3.71}$$

sobre λ, utilizando técnicas de optimización restringidas estándar. Ha sido probado por Baum que al maximizar $Q(\lambda, \overline{\lambda})$ se incrementa la verosimilitud.

El resultado de este procedimiento de reestimación es nombrado *estimación de máxima verosimilitud* del HMM. A partir de la reestimación de estos parámetros es posible mejorar la probabilidad de observar la secuencia de observaciones en un modelo, resolviendo de esta manera el problema tres (aprendizaje).

Capítulo 4

Diseño del Sistema de Reconocimiento de Voz y Resultados

Para el diseño del sistema se siguió la metodología presentada en el capítulo anterior. Para el desarrollo y codificación de todas las etapas que conforman el sistema de reconocimiento de voz se hizo uso del software Matlab, excepto la etapa de adquisición, la cual se hizo con el software WaveSurfer 1.8.8p4. Este software fue desarrollado por el Centro de Tecnología del Habla en la Universidad KTH de Estocolmo, Suecia.

En cuanto al hardware, se utilizó una computadora con Windows 7, un procesador Intel Core i5 y memoria RAM de 8Gb, además un micrófono con una respuesta en frecuencia de 50 Hz - 16 kHz, impedancia $\leq$2.2 kOhms y una sensibilidad de -30 dB $\pm$3dB.

Como recursos humanos fueron necesarias diez personas para la conformación del corpus de voz del sistema: cinco eran hombres y cinco mujeres, todos con diferente edad y características de voz. No se eligieron con ningún fin en particular de acuerdo a sus características fonéticas u otra característica, simplemente son colaboradores, amigos y familiares.

En este capítulo se describen como fueron diseñadas cada una de las etapas del sistema de reconocimiento voz, con las configuraciones que se consideraron más optimas para su funcionamiento y obtención de la mayor tasa de reconocimiento posible. Además, se presentan los resultados que se obtuvieron del sistema.

4.1 Adquisición de datos

Para la adquisición de datos se utilizó un micrófono marca Steren, modelo: MIC-550. Con una respuesta en frecuencia de 50 Hz - 16 kHz, impedancia $\leq$2.2 kOhms y una sensibilidad de -30 dB $\pm$3dB. El cual se conecta a la computadora mediante el puerto USB, sin necesidad de software adicional para su funcionamiento.

Se seleccionaron a diez locutores (cinco hombres y cinco mujeres) con edades entre 23 y 61 años de forma aleatoria, sin alguna característica específica en su voz. Los cuales grabaron diez repeticiones de veinte palabras con un solo canal de entrada (monocanal), frecuencia de muestreo de 44.1 kHZ (calidad CD) y extensión ".wav". Las señales de voz de entrada se adquirieron con la herramienta computacional WaveSurfer 1.8.8p4, el cual permite observar la forma de onda y su espectrograma. Con este mismo software se sub-muestrearon las grabaciones a 16kHZ y a 8kHZ para su posterior procesamiento.

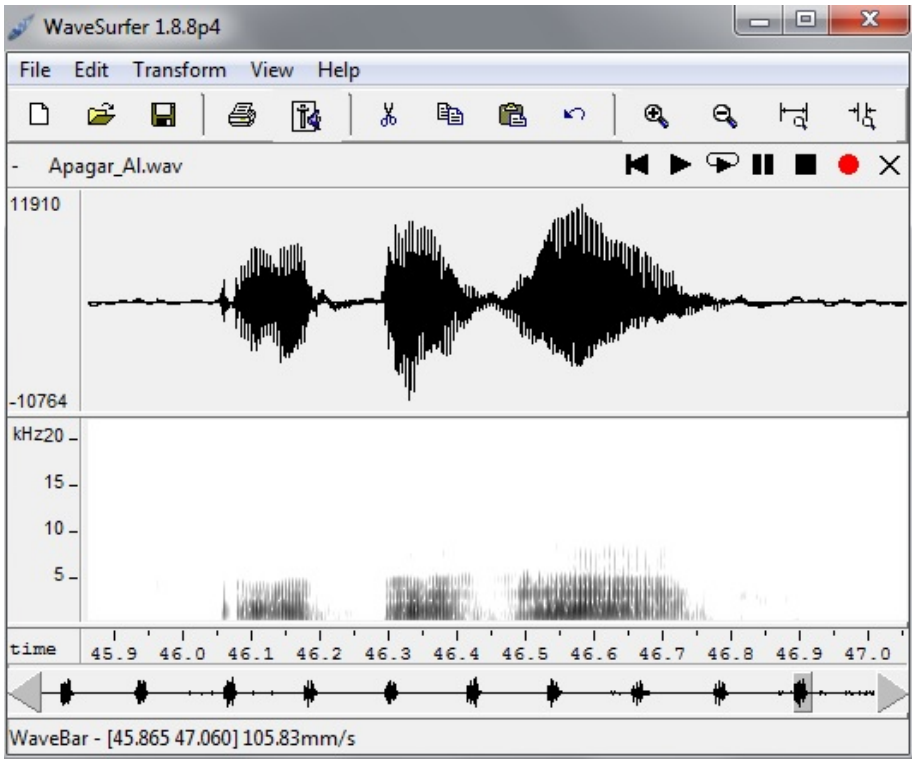


Figura 4.1 Interfaz del programa WaveSurfer mostrando la décima repetición de la palabra "Apagar" del locutor "Alma".

Las palabras que se grabaron son las siguientes:

- Encender
- Apagar
- Subir
- Bajar
- Volumen
- Buscar
- Llamar
- Colgar
- Enviar
- Mensaje
- Los dígitos del 0 al 9

Cada uno de los diez hablantes o locutores vocalizaron diez repeticiones de la misma palabra en una sola grabación de 51 segundos de duración, como se muestra en la figura 4.1. Obteniendo veinte grabaciones por cada locutor, una por cada palabra. Posteriormente se segmentaron las grabaciones con la herramienta computacional Matlab y se obtuvieron diez audios con una duración de 2 segundos, cada uno de ellos correspondiente a una repetición de la misma palabra. Este proceso se hizo para las veinte grabaciones de cada locutor, formando un vocabulario pequeño de 20 palabras y un corpus de voz grande de 2000 repeticiones, el cual fue almacenado en la computadora. Las grabaciones contienen ruido ambiental, ya que se realizaron en los cubículos de la Maestría, por lo tanto fue necesario hacer un pre-procesado de cada una de las señales de voz (2000 repeticiones).

4.2 Pre-Procesamiento

4.2.1 Filtrado

En el pre-procesamiento de la señal de voz se llevan a cabo varias sub-etapas, la primera de ellas es el filtrado de la señal de voz. En la cual se aplican un filtro pasa bajas y posteriormente un filtro pasa altas a cada una de las señales de voz (repeticiones) obtenidas en la adquisición de datos. Se diseñaron dos filtros pasa bajas con la aplicación "Filter disign and Analising" incorporada en Matlab; uno para las señales de voz con frecuencia de muestreo $Fm = 16kHz$

y otro para las señales muestradas a 8kHz. Para las señales con $Fm = 16kHz$ se diseñó un filtro pasa bajas de tipo Butterworth y respuesta infinita al impulso (IIR - Infinite Impulse Response), de orden 25 y frecuencia de corte $Fc = 6000Hz$, como se muestra en la figura 4.2.

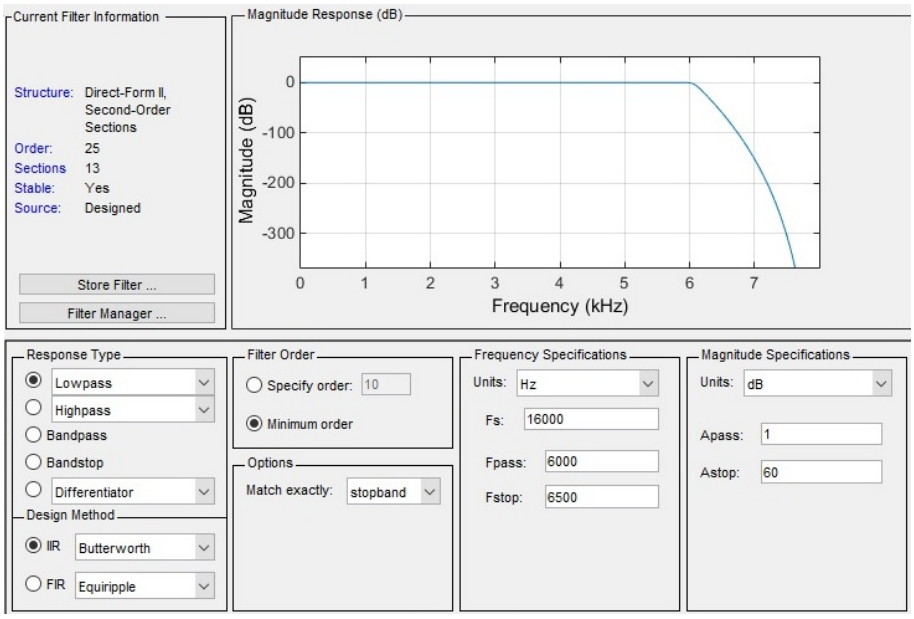


Figura 4.2 Diseño del filtro pasa bajas para las señales de voz con $Fm = 16$kHz.

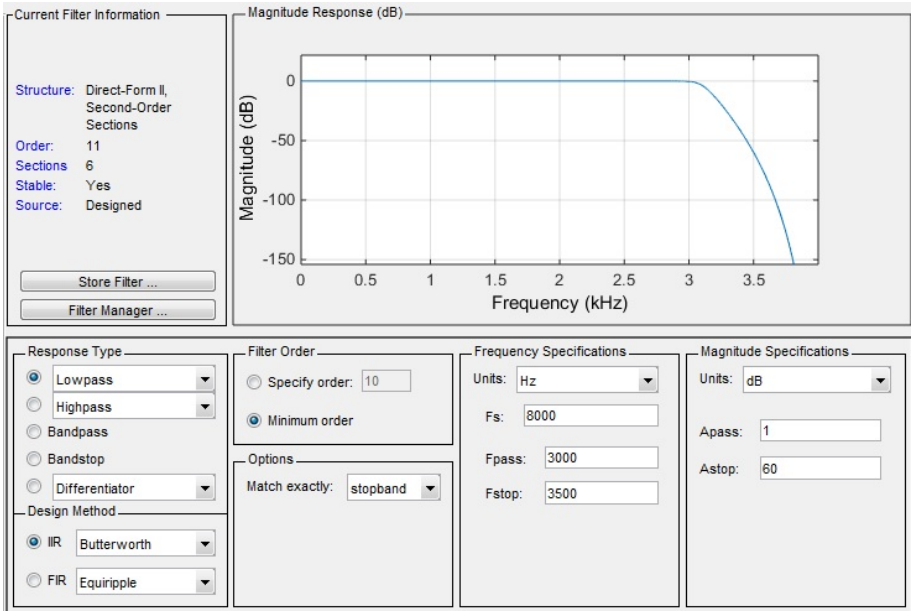


Figura 4.3 Diseño del filtro pasa bajas para las señales de voz con $Fm = 8$kHz.

Para las señales de voz con $Fm = 8kHz$ también se diseñó un filtro pasa bajas de tipo Butterworth IIR de orden 25 y frecuencia de corte $Fc = 3000Hz$ como se muestra en la figura 4.3.

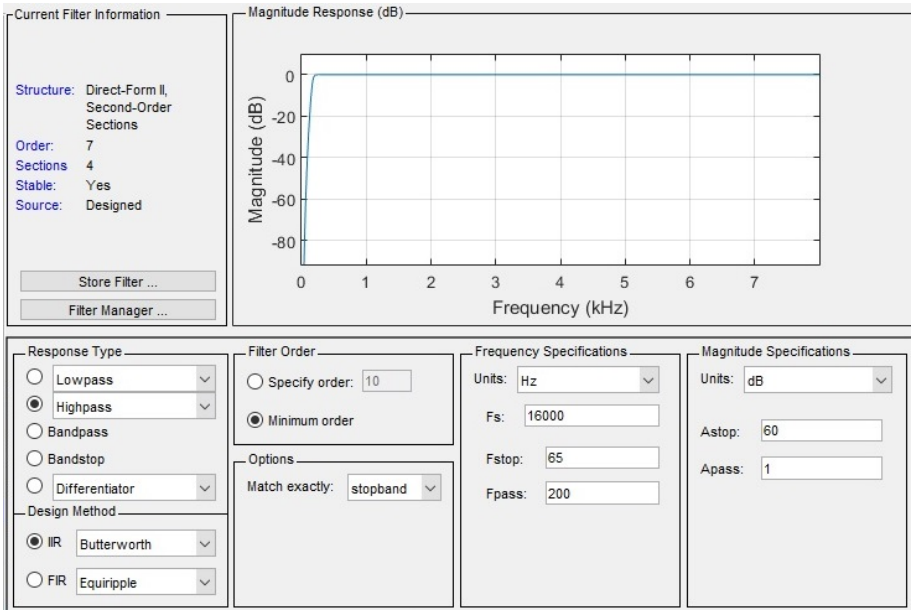


Figura 4.4 Diseño del filtro pasa altas para las señales de voz con $Fm = 16$kHz.

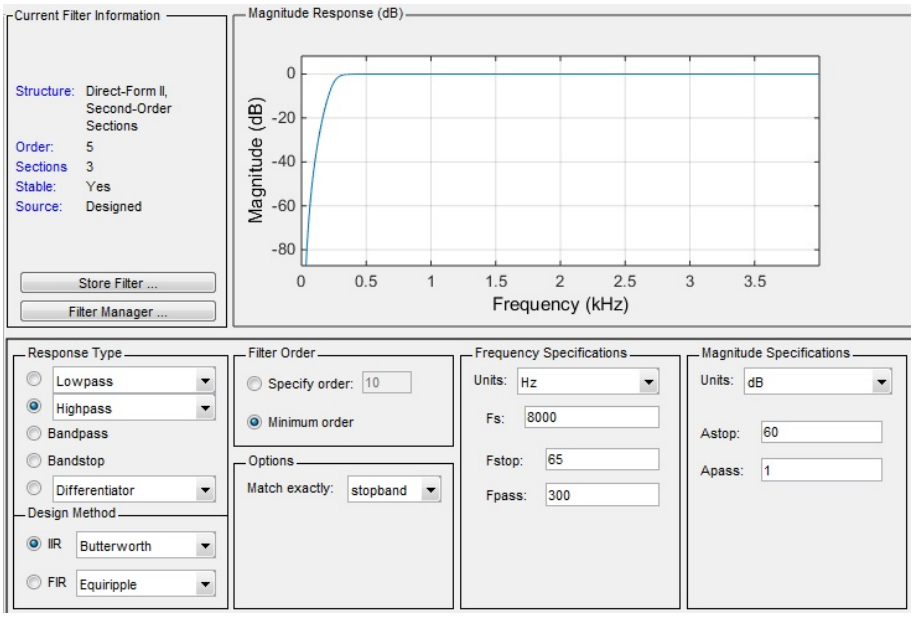


Figura 4.5 Diseño del filtro pasa altas para las señales de voz con $Fm = 8$kHz.

Para el diseño de los filtros pasa altas se llevó a cabo el mismo procedimiento como se muestra en las figuras 4.4 y 4.5. Para las señales de voz con $Fm = 16kHz$ se diseñó un filtro pasa altas de tipo Butterworth IIR de orden 7 y $Fc = 200Hz$, para las señales de voz con $Fm = 8kHz$ sólo cambio el orden de l filtro a 5.

Al diseñar el filtro se genera el vector columna "G" que contiene los valores de ganancia del filtro, también se genera la matriz "SOS (Second-Order Section)". Ambos valores son transformados mediante la función de Matlab $sos2tf$ para obtener la función de transferencia del filtro. Esto se hace para cada uno de los cuatro filtros.

Una vez que se tiene el corpus de voz y se obtienen las funciones de transferencia de los filtros pasa bajas y pasa altas, se filtran por separado cada una de las señales (repeticiones) del corpus de voz. Se selecciona al locutor, la palabra y la repetición que se desea filtrar, el filtro pasa bajas y el filtro pasa altas correspondiente según la frecuencia de muestreo (8kHz o 16kHz) de la señal de voz y mediante la función $filter$ de Matlab se filtran las frecuencias no deseadas de las señal en cuestión.

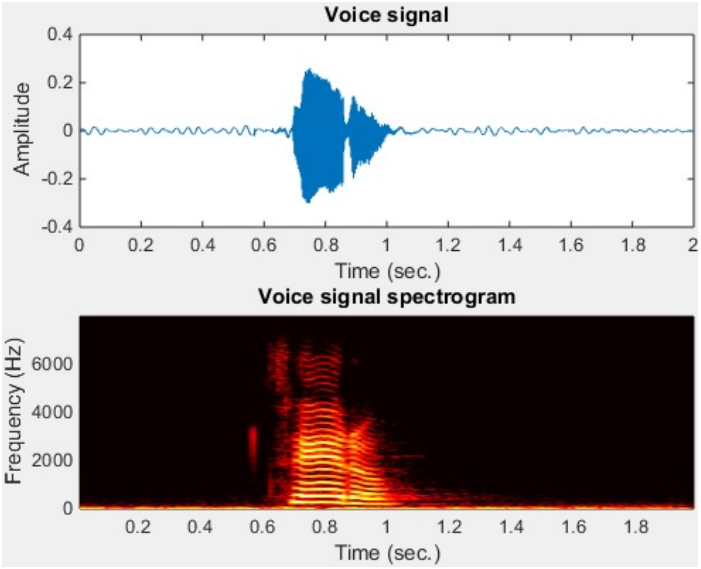


Figura 4.6 *Superior* - Forma de onda de la primera repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz. *Inferior* - Su espectrograma.

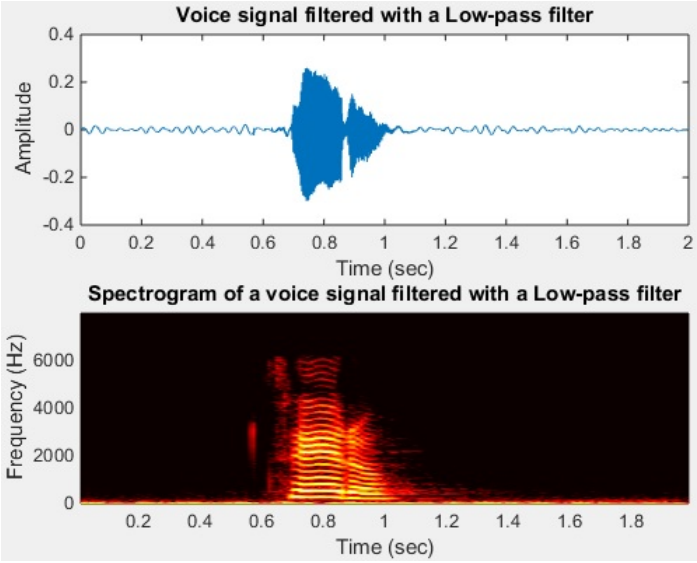


Figura 4.7 *Superior* - Forma de onda de la primera repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz filtrada con un pasa bajas. *Inferior* - Su espectrograma.

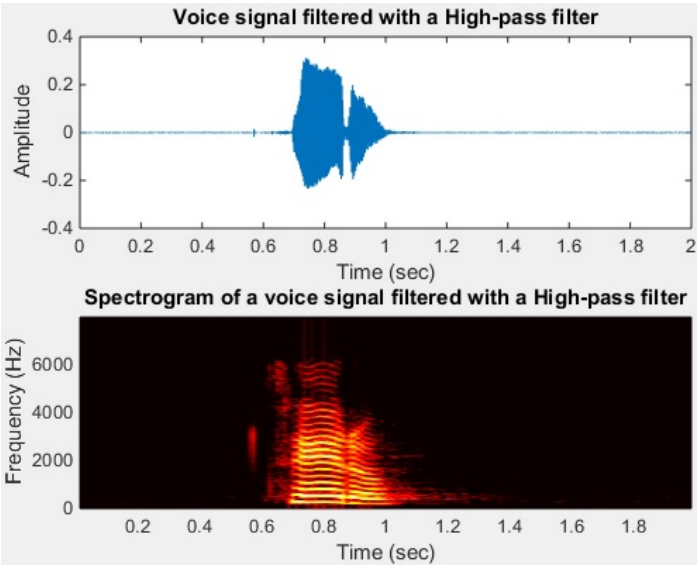


Figura 4.8 *Superior* - Forma de onda de la primera repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz filtrada con un pasa altas. *Inferior* - Su espectrograma.

En la figura 4.6 se muestra la señal de voz digitalizada de la palabra "cero" del locutor "Lulú" y su espectrograma, la figura 4.7 muestra la señal filtrada por un pasa bajas y su respectivo espectrograma y la figura 4.8 muestra la señal filtrada por un pasa altas y su espectrograma

correspondiente. Las señales fueron graficadas en Matlab. Los códigos de los filtros pasa bajas y pasa altas se muestran en el Apéndice A.

4.2.2 Detección de Actividad de Voz

Para la detección de actividad de voz se realizó el cálculo de la energía promedio de la señal de voz filtrada (en la etapa de filtrado), la energía promedio se obtuvo a partir de la ecuación 3.1 y se definió un umbral de decisión de 0.01. Posteriormente se calculó la energía de corto plazo en tramas con una duración de 25 mseg con la ecuación 3.2 (véase capítulo 3 para ambas ecuaciones). Las tramas cuya energía supere el producto de la energía promedio y el umbral de decisión fueron consideradas como segmentos de voz, las demás como información innecesaria.

Una vez obtenidas las tramas vocalizadas que contienen información lingüística, se definió a partir de la distancia entre cada una de las tramas subsecuentes el inicio y el fin de la palabra. En este caso, al ser palabras aisladas es más fácil determinar el inicio y fin de la palabra. El resultado del recorte de la señal de voz se muestra en la figura 4.9.

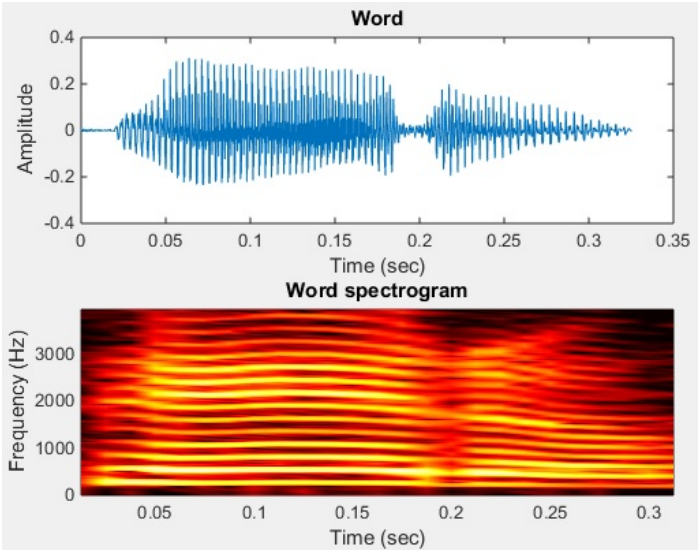


Figura 4.9 *Superior* - Forma de onda de la primera repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz sin intervalos de silencio. *Inferior* - Su espectrograma.

Se puede observar en la imagen superior de la figura 4.9 como se eliminaron los segmentos de la señal de voz que no contenían información lingüística y sólo permaneció la palabra "cero" del locutor "Lulú". En la imagen inferior se muestra su espectrograma. Las señales fueron graficadas en Matlab y los códigos de detección de voz se muestran en el Apéndice B.

4.2.3 Pre-énfasis

Lo primero que se debe de tomar en cuanta en el diseño del pre-énfasis es el valor que va tomar el factor de suavizado α. Aunque α puede tomar cualquier valor entre 0.9 y 1, si se asigna un valor arbitrario se puede incrementar el costo de implementación y la latencia del procesamiento. Para el pre-énfasis de la señal se eligió un factor de suavizado de 0.96875, que al sustituirse en la ecuación 3.6 se tiene:

$$S_{pp}[n] = S_{bp}[n] - 0.96875 S_{bp}[n-1] \tag{4.1}$$

valor asignado a partir del artículo "Pre-processing Speech Signals in FPGAs" [59] debido a los resultados que se obtuvieron.

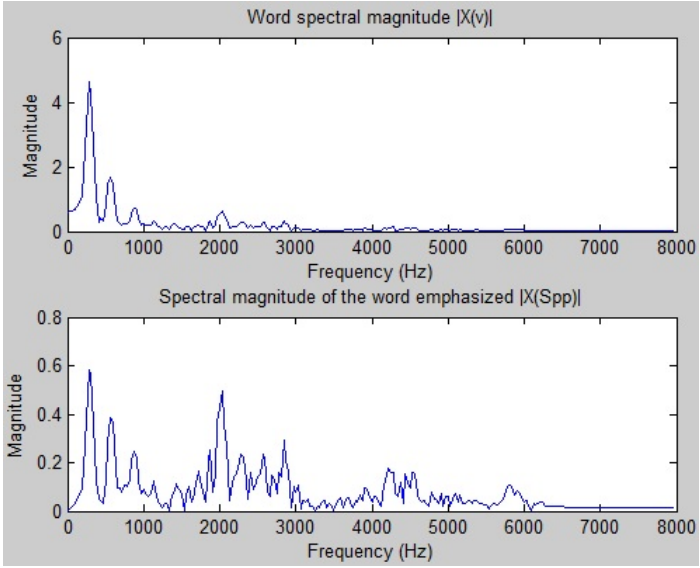


Figura 4.10 Superior - Magnitud espectral de la primera repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz. *Inferior* - Magnitud espectral de la primera repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz tras el filtro de pre-énfasis.

Para observar el enfatizado de las frecuencias altas de la señal de voz después de aplicar el filtro de pre-énfasis es necesario transformar la señal al dominio frecuencial. Para ello, se obtuvo la magnitud espectral de la señal de voz con la transformada rápida de Fourier (FFT) antes y después de aplicar el filtro.

La imagen superior de la figura 4.10 muestra la magnitud espectral de la señal de voz y la imagen inferior la magnitud espectral de la señal de voz pre-enfatizada. Ambas representan la distribución de aplitudes de cada frecuencia de la señal y se puede observar como se enfatizaron las frecuencias altas de la señal después de que se aplicó el filtro. Las señales fueron graficadas en Matlab y los códigos de pre-énfasis se muestran en el Apéndice C.

4.2.4 Segmentación y enventanado

Las señales de voz con $Fm = 8kHz$ se segmentaron en tramas de 256 muestras y las señales de voz con $Fm = 16kHz$ se segmentaron en tramas de 512 muestras. Las tramas fueron enventanadas mediante la ventana Hamming y traslapadas en un 50%. Las ventanas se almacenaron en una matriz, una ventana por fila y columnas igual al número de muestras por ventana (256 o 512).

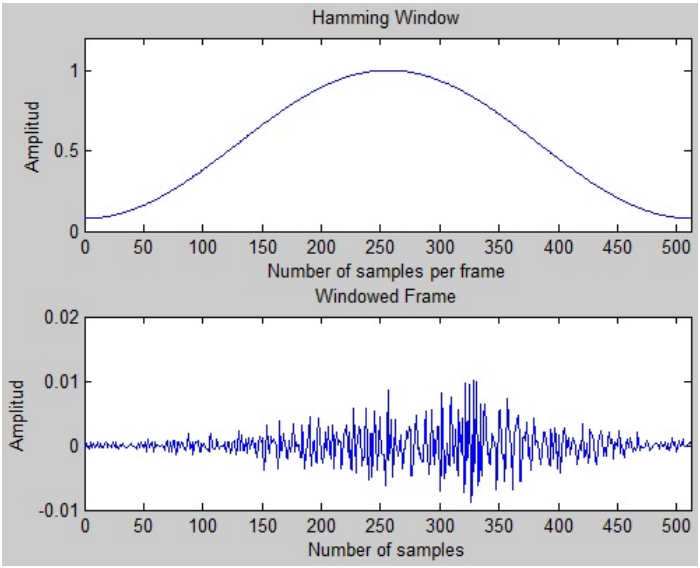


Figura 4.11 Superior - Ventana Hamming. *Inferior* - Ventana de una trama de la primera repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz.

En la imagen superior de la figura 4.11 se muestra una ventana Hamming de 512 muestras y en la imagen inferior se muestra una trama ventaneada de la señal de voz pre-enfatizada. Las señales fueron graficadas en Matlab y los códigos de segmentación y enventanado se muestran en el Apéndice D.

4.3 Extracción de Características

4.3.1 LPC

Como ya se mencionó en el capitulo anterior, para calcular los coeficientes LPC son necesarios cuatro etapas fundamentales: el pré-enfasis, segmentación y ventaneo, autocorrelación y análisis LPC. De las cuales las primeras dos (el pré-enfasis, segmentación y ventaneo) ya se realizaron en el pre-procesamiento.

A partir de la matriz de ventanas, se calculó la autocorrelación en cada una de esas ventanas (en cada fila) y se determinó que tanto se relacionan o se parecen las muestras de esa ventana para determinar su periodicidad.

Posteriormente se hizo el análisis LPC, donde se convierte cada ventana autocorrelacionada en un conjunto de parámetros LPC que representa los coeficientes LPC. Para realizar este proceso se debe determinar el orden de predicción lineal o número de coeficientes LPC que se desea obtener. El número de coeficientes que se calculó para las señales de voz con $Fm = 8kHz$ fue seleccionado a partir de [64], donde se menciona que el número de coeficientes que brindan buenos resultados para esta frecuencia de muestreo van de 8-12, por lo que se fue variando el número de coeficientes de 8-12 con incrementos unitarios.

Para calcular los coeficientes se implementó el algoritmo Levinson-Durbin, donde se obtiene una matriz cuyas filas son iguales al número de coeficientes (8-12) y las columnas igual al número de ventanas de la repetición de la palabra que se está analizando (varía según el tiempo de duración de la repetición de la palabra). El método de Levinson-Durbin está dado por el

siguiente algoritmo [8]:

$$E^{(0)} = r(0), \tag{4.2}$$

$$k_i = \left\{ r(i) - \sum_{j=1}^{L-1} a_j^{(i-1)} r(|i-j|) \right\} / E^{(i-1)}, \qquad 1 \leq i \leq p, \tag{4.3}$$

$$a_i^{(i)} = k_i, \tag{4.4}$$

$$a_i^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}, \qquad 1 \leq j \leq i-1, \tag{4.5}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)}, \tag{4.6}$$

donde E representa el error de predicción, k_i los coeficientes de reflexión entre las partes consecutivas del tubo acústico y a_i los coeficientes de predicción lineal. El algoritmo de Levinson-Durbin se utiliza para resolver de forma recursiva las ecuaciones lineales de la matriz Toeplitz de autocorrelaciones y obtener los coeficientes de predicción. Al graficar la matriz que contiene los coeficientes LPC se obtiene:

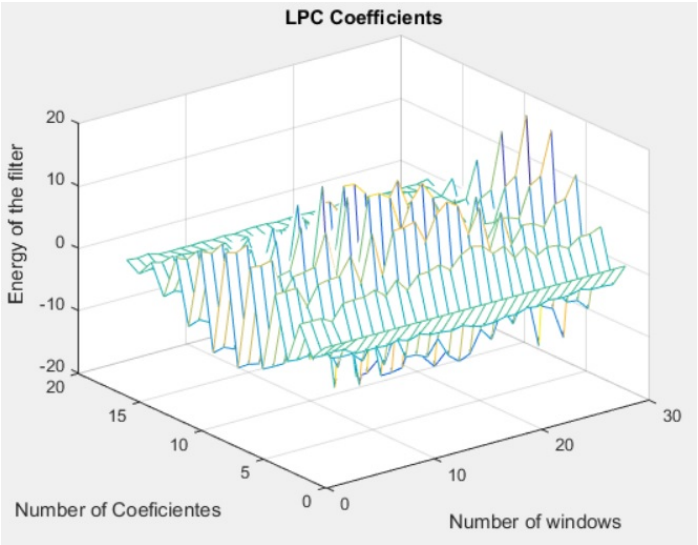


Figura 4.12 Matriz de coeficientes LPC de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz.

La figura 4.12 muestra la gráfica de la matriz de coeficientes LPC de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz. En la cual, se puede observar que el número de ventanas es igual a 29 y el número de coeficientes LPC que se calcularon por

ventana fueron 16. La señal fue graficada en Matlab y los códigos para obtención de los coeficientes LPC se muestran en el Apéndice E.

Este proceso se hace para cada una de las repeticiones (10) de todas las palabras (20) y todos los locutores (10). Se obtuvieron un total de 2000 matrices que contienen los coeficientes LPC que caracterizan a cada una de las señales de voz.

Para las señales de voz con $Fm = 16kHz$ el número de coeficientes LPC que se calcularon fueron el doble de los seleccionados en las señales de voz con $Fm = 8kHz$, es decir, de 16-24, con incrementos de dos en dos. Se seleccionó de tal forma que se tuviera la misma distribución frecuencial en las bandas de 0Hz a 4kHz para que fuera una comparación objetiva de la existencia de una mejora al duplicar la frecuencia de muestreo.

4.3.2 MFCC

Para el cálculo de los Coeficientes Cepstrales de Frecuencias Mel se llevan a cabo cuatro etapas de procesamiento para la señal de voz; La transformación de la señal de voz del dominio temporal al dominio frecuencial mediate la FFT, el cálculo del banco de filtros en escala Mel, el cálculo del logaritmo de las energías de los filtros y la trasformada de coseno discreto (DCT).

Una vez que la señal de voz de entrada fue pre-procesada y se obtuvo la matriz de ventanas, se transformó cada una de las ventanas del dominio temporal al frecuencial mediante la función de Matlab *fft*. Esta función calcula la transformada rápida de Fourier de la ventana, solamente se debe asignar la ventana que se desea transformar y su tamaño. Para el caso de las señas de voz muestreadas a $8kHz$ el tamaño de las ventanas es de 256 muestras y para las señales de voz muestreadas a $16kHz$ el tamaño de las ventanas es de 512 muestras. Al transformar cada una de las ventanas se obtiene la magnitud y la fase de cada una de ellas. Sin embargo, soló la magnitud contiene información de interés para el reconocimiento de voz. Al elevar al cuadro la magnitud se obtiene la densidad espectral de potencia, a partir de la cual es posible observar

la energía de las bandas de frecuencia que se encuentran presentes en la ventana que se está analizando.

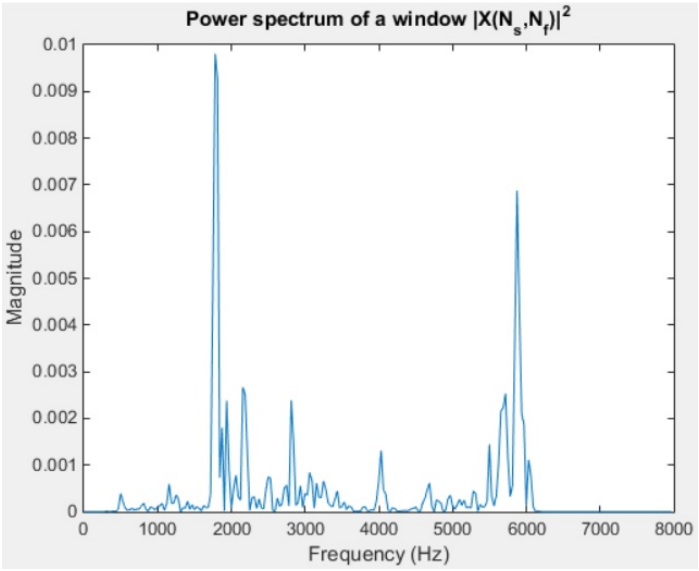


Figura 4.13 Densidad espectral de la segunda ventana de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz.

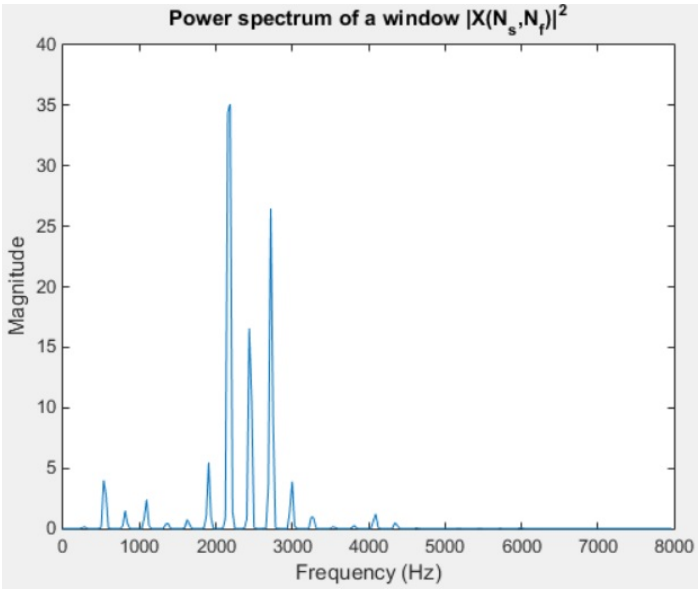


Figura 4.14 Densidad espectral de la onceava ventana de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz.

Se puede observar en la figura 4.13 la densidad espectral de una ventana de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz, en la que las bandas

de frecuencia que presentan mayor energía se encuentran en 1.8kHz y 6.9kHz. Sin embargo, la energía de estas dos bandas de frecuencia y las que se encuentran entre ellas varían muy cercanas a cero, esto se debe a que la ventana que se está analizando no contiene parte de la señal vocalizada. En cambio, en la figura 4.14 se puede observar tres bandas de frecuencia con un contenido energético muy por encima de cero, esto se debe a que la onceava ventana de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz, contiene información lingüística.

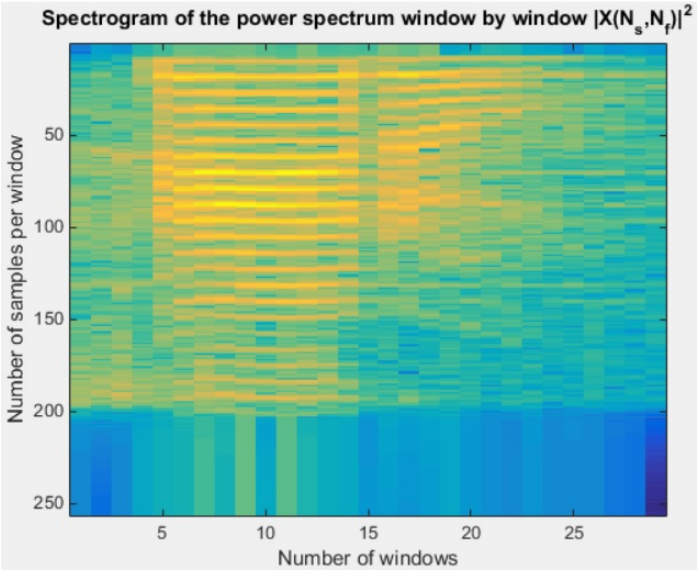


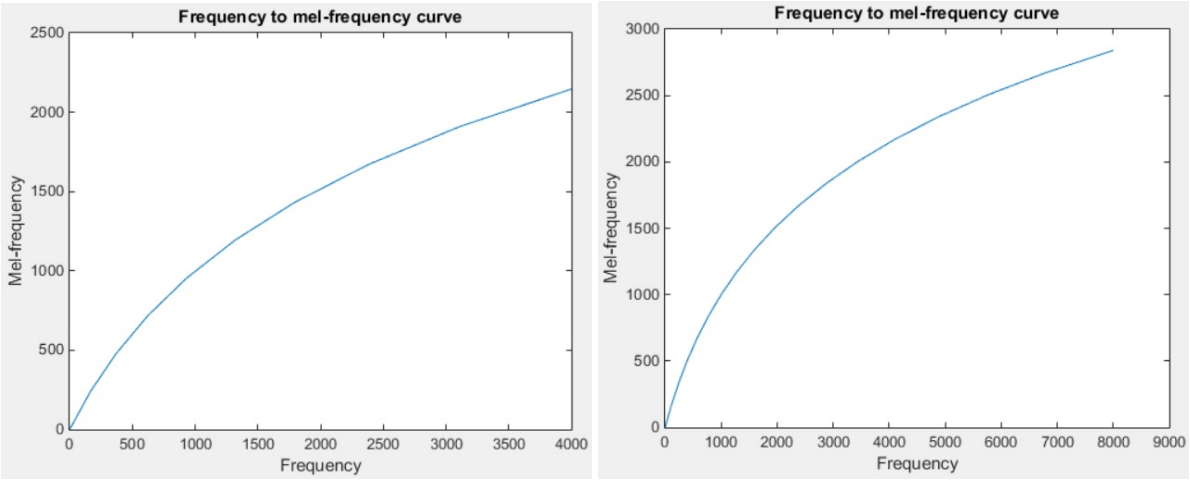
Figura 4.15 Espectrograma de la densidad espectral de todas las ventanas de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz.

En la figura 4.15 se muestra el espectrograma de todas las ventanas de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz. Se puede observar como se distribuye la energía en las 29 ventanas que conforman la señal de voz. Las ventanas centrales (5-22) son las de mayor contenido energético en esta señal de voz.

En la siguiente etapa se calculó el banco de filtros, para el cual se eligieron usar filtros triangulares. Se definieron la frecuencia mínima y máxima del rango del banco de filtros, la frecuencia mínima fue igual a cero ($f_{min} = 0Hz$) tanto para las señales de voz con $Fm =$

$8kHz$ como para las señales de voz con $Fm = 16kHz$. La frecuencia máxima se definió a partir del teorema de muestreo, el cual especifica que la frecuencia máxima no puede ser mayor a la mitad de la frecuencia de muestreo. Por lo tanto, para las señales de voz con $Fm = 8kHz$ la frecuencia máxima que se eligió fue $f_{max} = 4kHz$ y para las señales de voz con $Fm = 16kHz$ la frecuencia máxima fue $f_{max} = 8kHz$. Estas frecuencias máximas y mínimas se convirtieron a escala Mel, mediante la siguiente ecuación:

$$Mel(f) = 2595log\left(1 + \frac{f}{700}\right). \tag{4.7}$$



(a) $Fm = 8kHz$ (b) $Fm = 16kHz$

Figura 4.16 Curvas de relación de la frecuencia Mel y la frecuencia en Hertz para las señales de voz con $Fm = 8kHz$ y $Fm = 16kHz$.

Debido a que la frecuencia mínima en Hertz que se definió es igual a cero, la frecuencia mínima en escala Mel también es igual a cero ($Mel_{min}(f) = 0$) para ambas frecuencias de muestreo (8 y $16kHz$). En cambio, para las señales de voz muestreadas a $8kHz$ su frecuencia máxima en escala Mel es $Mel_{max}(f) = 2146.06$ y para las señales de voz muestreadas a $16kHz$ su frecuencia máxima en escala Mel es $Mel_{max}(f) = 2840$, esto se puede observar en la figura 4.16.

Para conocer las frecuencias centrales de los filtros triangulares en Hertz, primero se definieron las frecuencias centrales en escala Mel debido a que están linealmente distanciadas (en

Hertz no lo están). Esta distribución depende del rango de frecuencia en el que el banco de filtros se definió ($Mel_{min}(f)$ y $Mel_{max}(f)$) y de la cantidad de filtros en el banco. Para las señales de voz con $Fm = 8kHz$ y 8 filtros por banco se obtuvo el siguiente vector que contiene las frecuencias centrales de los filtros triangulares en escala Mel:

$$[0 \quad 238.45 \quad 476.90 \quad 715.35 \quad 953.80 \quad 1192.25 \quad 1430.70 \quad 1669.16 \quad 1907.61 \quad 2146.06]. \tag{4.8}$$

Los valores de las frecuencias centrales de los filtros triangulares en escala Mel del vector 4.8 se convirtieron a Hertz con la siguiente ecuación:

$$Mel^{-1}(f) = 700(exp^{Mel/1125} - 1) \tag{4.9}$$

y se obtuvo el siguiente vector:

$$[0 \quad 164.94 \quad 368.74 \quad 620.57 \quad 931.74 \quad 1316.24 \quad 1791.32 \quad 2378.36 \quad 3103.72 \quad 4000]. \tag{4.10}$$

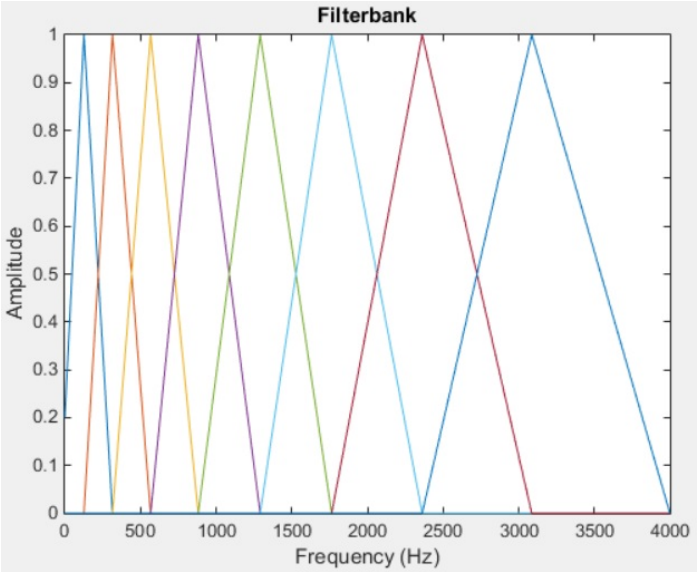


Figura 4.17 Banco de filtros para las señales de voz con $Fm = 8kHz$.

Se puede observar que el vector 4.10 contiene diez valores, de los cuales, el primer valor (0Hz) corresponde a la frecuencia en la que inicia el primer filtro y el último valor (4000Hz)

corresponde a la frecuencia en la que termina el último filtro, los ocho valores restantes del vector corresponden a las frecuencias centrales de los filtros triangulares. En la figura 4.17 se muestra el banco de filtros triangulares de las señales con $Fm = 8kHz$, cada uno de los filtros se caracteriza por tener su frecuencia central con amplitud unitaria.

Para las señales de voz con $Fm = 16kHz$ y 16 filtros por banco el vector que se obtuvo fue:

$$[0 \quad 167.06 \quad 334.12 \quad 501.18 \quad 668.24 \quad 835.30 \quad 1002.36 \quad 1169.42 \quad 1336.48 \quad 1503.54 \quad 1670.60 \quad 1837.66 \quad 2004.72 \quad 2171.78 \quad 2338.84 \quad 2505.90 \quad 2672.96 \quad 2840.02] \tag{4.11}$$

y al convertir el vector de las frecuencias centrales de los filtros en escala Mel a Hertz se obtuvo el siguiente vector:

$$[0 \quad 111.84 \quad 241.57 \quad 392.02 \quad 566.51 \quad 768.881 \quad 1003.58 \quad 1275.79 \quad 1591.49 \quad 1957.64 \quad 2382.30 \quad 2874.80 \quad 3446.01 \quad 4108.48 \quad 4876.81 \quad 5767.90 \quad 6801.38 \quad 8000]. \tag{4.12}$$

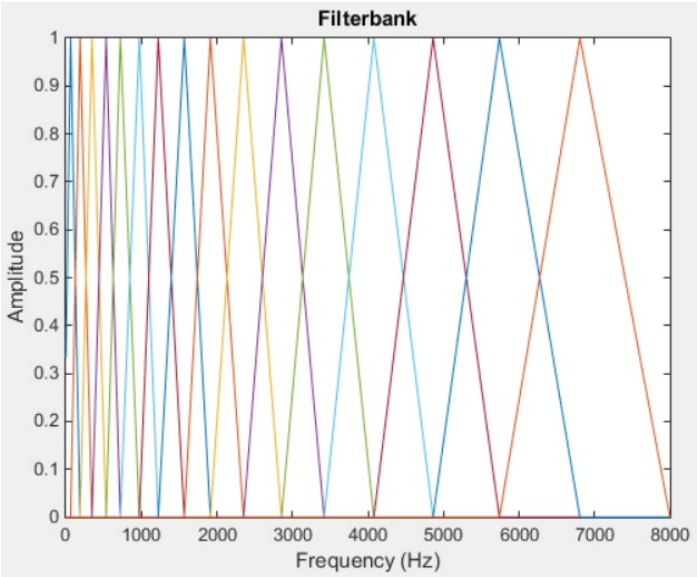


Figura 4.18 Banco de filtros para las señales de voz con $Fm = 16kHz$.

Se puede observar que el vector 4.12 contiene dieciocho valores, de los cuales, el primer valor (0Hz) corresponde a la frecuencia en la que inicia el primer filtro y el último valor

(8000Hz) corresponde a la frecuencia en la que termina el último filtro, los ocho valores restantes del vector corresponden a las frecuencias centrales de los filtros triangulares. En la figura 4.18 se muestra el banco de filtros triangulares de las señales con $Fm = 16kHz$, cada uno de los filtros se caracteriza por tener su frecuencia central con amplitud unitaria.

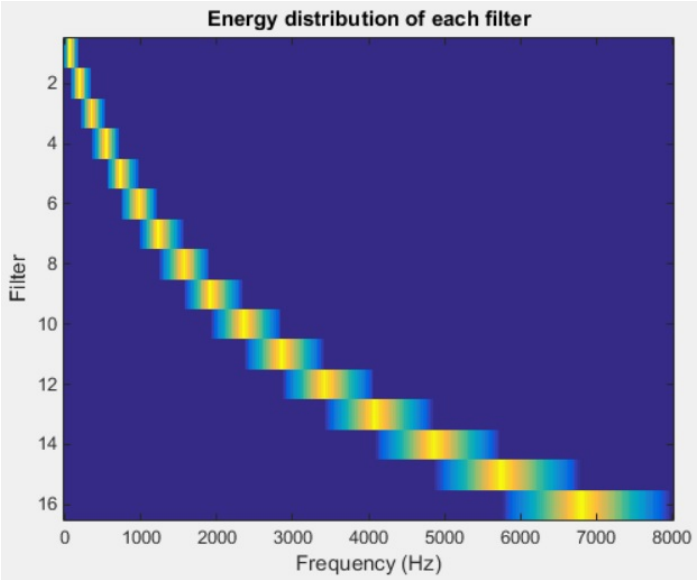


Figura 4.19 Distribución de energía del banco de filtros de las señales de voz con $Fm = 16kHz$.

La figura 4.19 muestra el espectrograma de la distribución de energía de cada uno de los filtros del banco de las señales de voz con una $Fm = 16kHz$, se puede observar de manera más simple en esta figura que frecuencias conforman cada filtro que en la figura 4.18. Cada uno de los filtros del banco se debe multiplicar por cada una de las ventanas de densidades espectral obtenidas de la FFT. Esto se hace para saber cuanta energía hay en cada punto de la ventana de densidad espectral con respecto al filtro con el que se está multiplicando. En la figura 4.20 se puede ver que cada ventana está representada por un linea de color distinto y que los filtros van de uno a dieciséis, los filtros que presentan una mayor energía son el 4, 9, 10 y 11. Esto se debe a que en las bandas de frecuencia que corresponden a cada uno de estos filtros, se encuentra presente la señal de voz vocalizada, la cual tiene contenido energético relevante. Esto es más fácil observarlo en el espectrograma de la figura 4.21, en el cual, entre más claro es el color del cuadro, mayor es la energía que presenta.

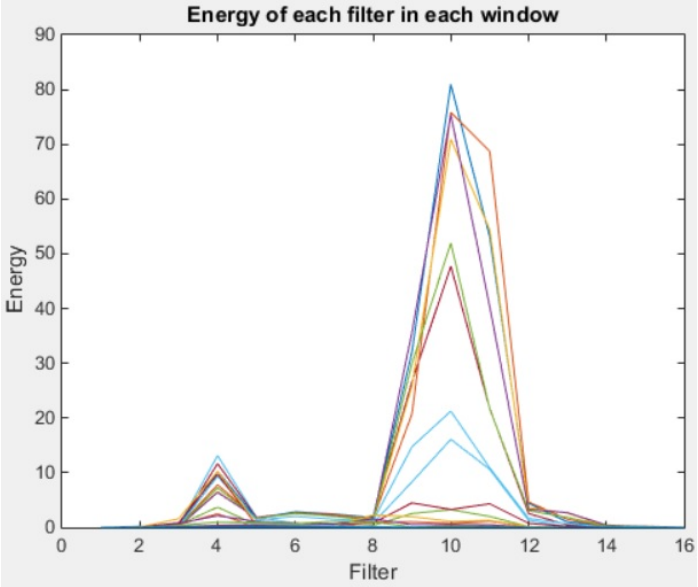


Figura 4.20 Energía de cada filtro en cada ventana de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16kHz$.

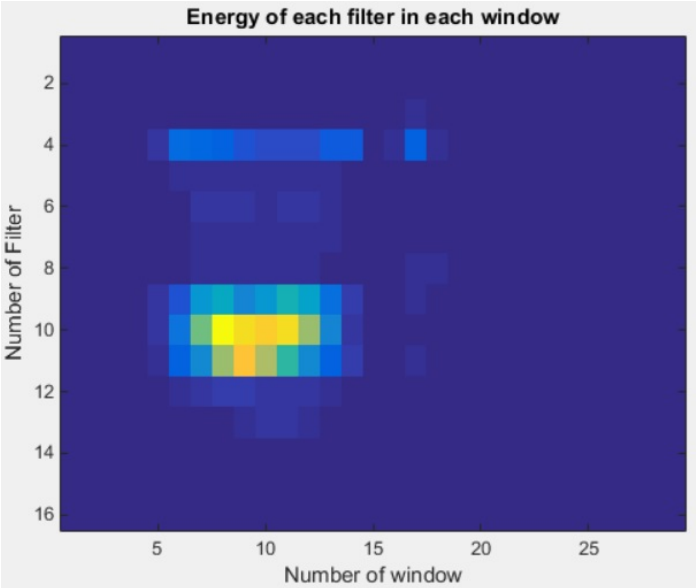


Figura 4.21 Espectrograma de la energía de cada filtro en cada ventana de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16kHz$.

Se puede observar en la figura 4.21 que sólo son visibles los filtros de las ventanas con mayor energía. Sin embargo, hay mas filtros que presentan energía y para hacerlos visibles gráficamente y perceptibles al oído humano se calcula el logaritmo de las energías de los bancos

de filtros obtenidos, el logaritmo aproxima la relación entre la percepción humana del volumen y la intensidad del sonido. Al calcular el logaritmo de las energías de cada filtro de cada ventana se obtiene la figura 4.22, la cual muestra la energía logarítmica de cada uno de los filtros de cada ventana de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16kHz$.

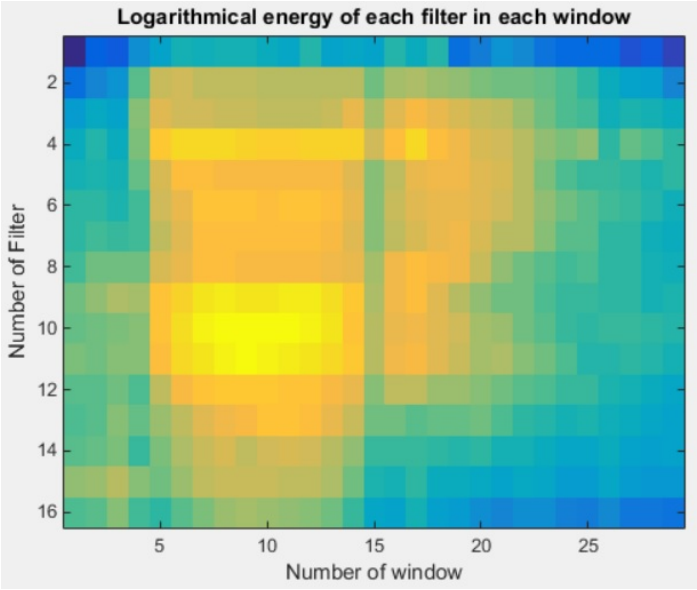


Figura 4.22 Espectrograma de la energía logarítmica de cada filtro en cada ventana de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16kHz$.

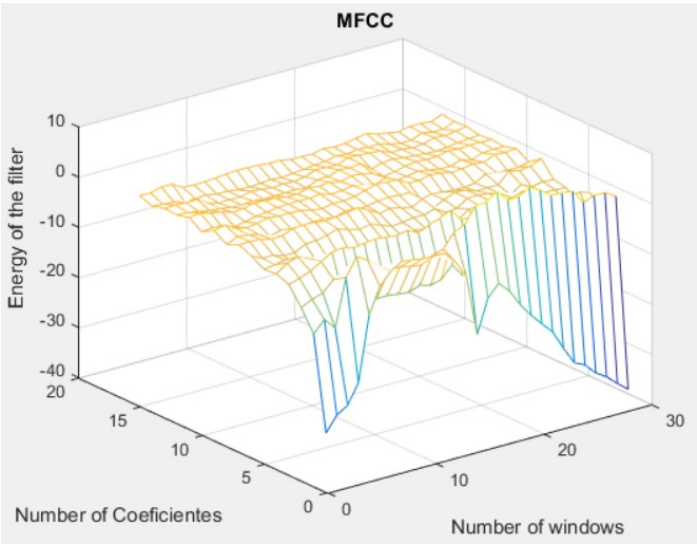


Figura 4.23 Matriz de coeficientes MFCC de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz.

En la última etapa se calculó la DCT del logaritmo de las energías del banco de filtros mediante la función *dtc* de Matlab, la cual devuelve la transformada de coseno discreto unitaria de la energía logarítmica de los filtros, obteniendo como resultado los coeficientes cepstrales de frecuancias Mel.

La figura 4.23 muestra la gráfica de la matriz de coeficientes MFCC de la primer repetición de la palabra "cero" del locutor "Lulú" con $Fm = 16$kHz. En la cual, se puede observar que el número de ventanas es igual a 29 y el número de coeficientes MFCC que se calcularon por ventana fueron 16. La señal fue graficada en Matlab y los códigos para obtención de los coeficientes MFCC se muestran en el Apéndice F.

4.4 Reconocimiento de Voz

Una vez obtenidas las matrices de coeficientes de las 2000 repeticiones que conforman el corpus de voz (20 repeticiones de 20 palabras dichas por 10 locutores) a partir de técnicas de extracción de características; LPC y MFCC con frecuencias de muestreo de $8kHz$ y $16kHz$ para ambos métodos. Se realizó la etapa de reconocimiento de voz mediante los modelos ocultos de Markov (HMM) desarrollados por Kevin Murphey [76].

Las matrices de coeficientes LPC y MFCC se almacenaron como datos tipo celdas y no como vectores característicos (como comúnmente se hace). Esto se debe a que las palabras que conforman el corpus de voz tienen diferente duración, lo cual genera que el número de ventanas de cada repetición de la misma palabra sea diferente y por lo tanto no se pueda comparar una repetición respecto a otra, incluso si la palabra es pronunciada por el mismo locutor. Existen técnicas como el alineamiento dinámico temporal (DTW - Dynamic Time Warping) que permiten alinear temporalmente a las muestras de voz. Sin embargo, este tipo de algoritmos generan mayor gasto computacional y a la vez vuelven al proceso de reconocimiento más lento conforme aumenta la frecuencia de muestreo, aunque también es cierto que estos

algoritmos mejoran la tasa de reconocimiento.

A partir de las matrices de coeficientes LPC y MFCC almacenadas, fueron creados los modelos de entrenamiento mediante el algoritmo de estimación máxima Baum-Welch. Se creo un modelo por cada una de las 20 palabras que conforman el vocabulario del sistema de reconocimiento de voz. Para la prueba o etapa de reconocimiento de las palabras de entrada que deseaban reconocer, se hizo una comparación con respecto a los modelos almacenados en el sistema en la etapa de entrenamiento, esta comparación se hizo mediante el algoritmo de Viterbi. Se entrenó el 70% de las repeticiones de cada palabra (1400 repeticiones) y se probó el 30% de las repeticiones restantes de las mismas palabra (600 repeticiones). Con el fin de que el sistema fuera capaz de reconocer 20 palabras aisladas distintas. El numeró de repeticiones (muestras) a reconocer se eligió a partir de la *Fórmula de la toma de la Muestra*:

$$muestra = \frac{N \cdot Z^2 \cdot p \cdot q}{(N-1) \cdot E^2 + Z^2 \cdot p \cdot q}, \tag{4.13}$$

donde N es el tamaño de la población, Z el nivel de confianza, p la probabilidad de éxito, q la probabilidad de fracaso y E error máximo admisible. Según esta fórmula, para tener un error menor al 4% al probar el sistema se deben de tomar como muestra 600 repeticiones.

Todos los modelos creados en la etapa de entrenamiento están conformados por 5 estados, para los cuales se usó una secuencia de transición basada en la topología left-right. Se usaron 3 mezclas gaussianas por estado. El número de observaciones del modelo se fue variando según la cantidad de coeficientes que caracterizaron a la palabra, usando el mismo número de observaciones que coeficientes según la configuración. Es decir, para el caso de LPC y MFCC con una frecuencia de muestro de $8kHZ$ tanto el número de coeficientes como el número de observaciones se varió de uno en uno entre 8 y 12, mientras que para el caso de LPC y MFCC con $Fm = 8kHZ$ se varió de dos en dos entre 16 y 24.

Definiendo el número de estados, de mezclas gaussianas, observaciones y haciendo uso del algoritmo de Baum-Welch se obtuvieron los modelos base. A partir de los cuales se estimaron

los parámetros que definieron a cada modelo HMM por palabra; A, π, B, μ, σ. Para estimar estos parámetros, se aplicaron los algoritmos de Baum-Welch y de forward-backward. Se definió la probabilidad pasada, la probabilidad actual, la convergencia, una tolerancia de convergencia y un número de iteraciones. Mientras el número de iteraciones fue menor al número de iteraciones máximo y la tolerancia de covarianza menor a la covarianza, se calcularon la función de densidad probabilidad para cada vector característico y las probabilidades logarítmicas, además se estimó la convergencia. Al existir una convergencia entre los vector característico, se obtenían los parámetros A, π, B, μ, σ estimados.

Para la etapa de prueba o reconocimiento de voz, se hizo uso del algoritmo de Viterbi. A partir del cual, se comparó el 30% de las repeticiones no modeladas contra los modelos de las palabras creados en la etapa de entrenamiento. El modelo que arrojó la mayor probabilidad logarítmica fue el de la palabra reconocida. Y posteriormente se definió si la palabra fue reconocida correctamente o incorrectamente, como se muestra en el siguiente apartado de resultados.

Los algoritmos principales de entrenamiento LPC con $Fm = 8kHZ$ y $Fm = 16kHZ$ se muestran en el apéndice G, y los de prueba en el apéndice H. Mientras que los algoritmos principales de MFCC con $Fm = 8kHZ$ y $Fm = 16kHZ$ se muestran en el apéndice I, y los de prueba en el apéndice J.

4.5 Resultados

En las siguientes tablas y gráficas se presentan los resultados de la investigación, se muestran las tasas de reconocimiento de voz de cada una de las palabras y el tiempo de procesamiento con las dos técnicas de extracción de características; LPC y MFCC, para las señales muestreadas a 8kHz y a 16kHz.

Tabla 4.1 Tasa de reconocimiento de voz usando LPC con una $Fm = 8kHz$ y diferente número de coeficientes.

	No. de Coeficientes LPC para Fm=8kHz				
Palabra	**8**	**9**	**10**	**11**	**12**
Cero	90%	93.3%	90%	93.3%	96.6%
Uno	100%	86.6%	86.6%	93.3%	100%
Dos	100%	96.6%	100%	100%	100%
Tres	96.6%	96.6%	83.3%	83.3%	90%
Cuatro	93.3%	86.6%	83.3%	83.3%	90%
Cinco	100%	100%	96.6%	100%	100%
Seis	93.3%	96.6%	96.6%	96.6%	96.6%
Siete	96.6%	83.3%	96.6%	96.6%	96.6%
Ocho	96.6%	93.3%	100%	100%	93.3%
Nueve	96.6%	90%	100%	100%	93.3%
Encender	96.6%	100%	96.6%	96.6%	100%
Apagar	96.6%	96.6%	100%	100%	100%
Subir	100%	100%	96.6%	100%	100%
Bajar	100%	90%	96.6%	80%	100%
Volumen	96.6%	100%	96.6%	96.6%	96.6%
Llamar	93.3%	100%	96.6%	100%	93.3%
Colgar	96.6%	93.3%	86.6%	80%	83.3%
Enviar	96.6%	96.6%	96.6%	100%	93.3%
Mensaje	96.6%	93.3%	96.6%	93.3%	96.6%
Buscar	96.6%	96.6%	90%	96.6%	93.3%
Tasa de Reconocimiento Máxima	100%	100%	100%	100%	100%
Tasa de Reconocimiento Mínima	93.3%	83.3%	83.3%	80%	83.3%
Promedio de la Tasa de Reconocimiento	96.62%	94.46%	94.29%	94.47%	95.64%
Tiempo de Entrenamiento	4’03”	3’58”	3’18”	3’14”	2’50”
Tiempo de Reconocimiento	51”	42”	47”	38”	37”

En la tabla 4.1 se muestra la tasa de reconocimiento de voz de cada una de las veinte palabras con diferente número de coeficientes LPC a una $Fm = 8kHz$. En la parte inferior de la tabla se muestra la tasa máxima, mínima y promedio de cada número de coeficientes LPC, además del tiempo de entrenamiento y de reconocimiento.

La tasa de reconocimiento máxima para LPC con $Fm = 8kHz$ fue del 100% para cada conjunto de coeficientes característicos (8-12). Esto indica que por lo menos una de las palabras en cada conjunto de coeficientes LPC tuvo el 100% de reconocimiento. No se graficó debido a que no hay diferencias que observar.

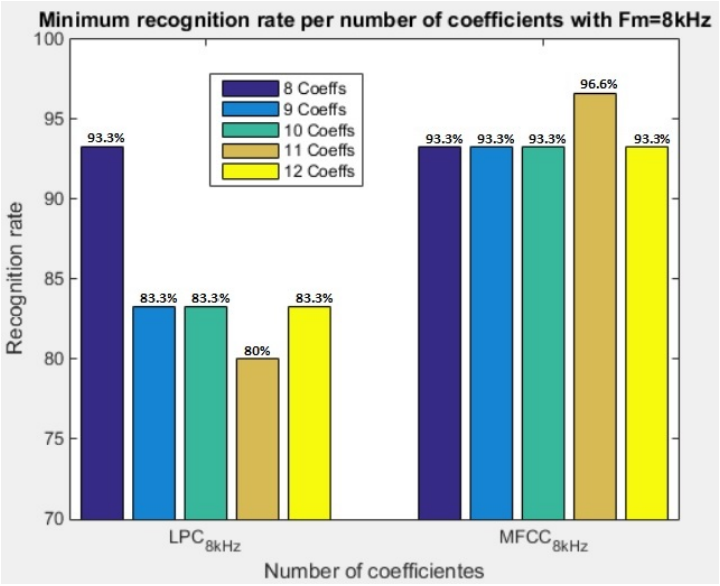


Figura 4.24 Tasa de reconocimiento mínima por número de coeficientes LPC y MFCC con $Fm = 8kH$.

La figura 4.24 muestra la tasa de reconocimiento mínima por número de coeficientes LPC con $Fm = 8kHz$. Se puede observar que fue menor para 11 coeficientes LPC con 80% de reconocimiento, la mayor tasa fue de 93.3% para 8 coeficientes LPC y para el resto de los conjuntos de coeficientes LPC la tasa de reconocimiento fue 83.3%. Esto demuestra que la extracción de características LPC de alguna de las palabras con $Fm = 8kHz$ y con cierto número de coeficientes, tiene por lo menos una tasa de reconocimiento del 80%. Las palabras ¨Bajar¨ y ¨Colgar¨ con 11 coeficientes LPC y $Fm = 8kH$ tuvieron la tasa mínima de 80%.

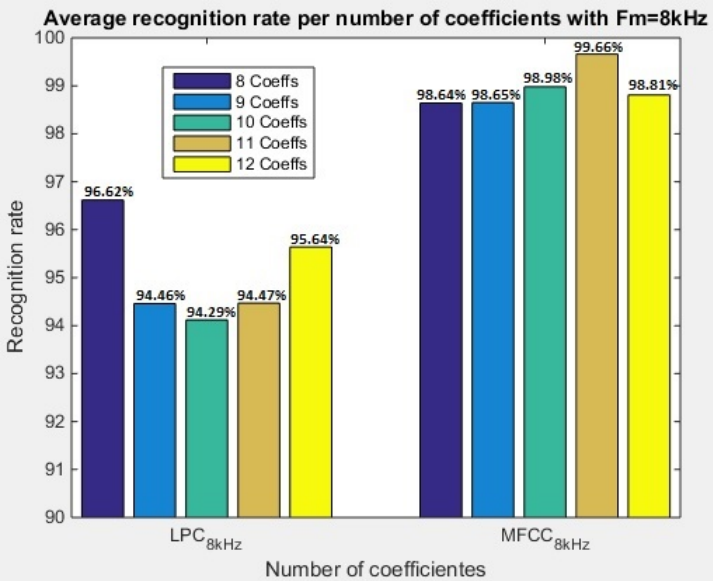


Figura 4.25 Tasa de reconocimiento promedio por número de coeficientes LPC y MFCC con $Fm = 8kH$.

La figura 4.25 muestra la tasa de reconocimiento promedio por número de coeficientes LPC con $Fm = 8kHz$. Se puede observar que la tasa de reconocimiento de los cinco conjuntos de coeficientes varió entre 94.29% y 96.62%, siendo 10 coeficientes LPC la tasa de menor certeza y 8 coeficientes LPC la mayor tasa de reconocimiento. Para el resto de coeficientes LPC la tasa de reconocimiento varió entre 94.46% y 95.64%. Está gráfica muestra que el promedio de reconocimiento para los cinco conjuntos de coeficientes LPC supera el 94.29%.

Las palabras con la mayor tasa de reconocimiento para LPC con $Fm = 8kHz$ fueron "Dos", "Cinco" y "Subir" con 99.32%, la palabra con menor tasa de reconocimiento fue "Cero" con 95.96%. Estas tasas de reconocimiento se obtuvieron del promedio de los cinco conjuntos de coeficientes LPC de cada palabra.

El tiempo de entrenamiento para LPC con $Fm = 8kHz$ fue menor para la configuración de 12 coeficientes LPC (2'50") y mayor para 8 coeficientes LPC (4'03"). Lo mismo ocurre en el tiempo de reconocimiento para está técnica, fue menor para 12 coeficientes LPC (37") y mayor para 8 coeficientes LPC (51").

Tabla 4.2 Tasa de reconocimiento de voz usando MFCC con una $Fm = 8kHz$ y diferente número de coeficientes.

	No. de Coeficientes MFCC para Fm=8kHz				
Palabra	**8**	**9**	**10**	**11**	**12**
Cero	93.3%	96.6%	100%	96.6%	93.3%
Uno	100%	100%	100%	100%	100%
Dos	100%	100%	100%	100%	100%
Tres	96.6%	96.6%	100%	100%	96.6%
Cuatro	96.6%	100%	100%	100%	100%
Cinco	100%	100%	100%	100%	100%
Seis	100%	96.6%	96.6%	100%	96.6%
Siete	96.6%	100%	96.6%	96.6%	100%
Ocho	100%	100%	100%	100%	100%
Nueve	100%	100%	100%	100%	100%
Encender	100%	100%	100%	100%	100%
Apagar	100%	100%	100%	100%	100%
Subir	100%	96.6%	100%	100%	100%
Bajar	96.6%	93.3%	100%	100%	96.6%
Volumen	100%	100%	96.6%	100%	100%
Llamar	100%	100%	93.3%	100%	100%
Colgar	100%	100%	100%	100%	100%
Enviar	96.6%	93.3%	96.6%	100%	96.6%
Mensaje	96.6%	100%	100%	100%	96.6%
Buscar	100%	100%	100%	100%	100%
Tasa de Reconocimiento Máxima	100%	100%	100%	100%	100%
Tasa de Reconocimiento Mínima	93.3%	93.3%	93.3%	96.6%	93.3%
Promedio de la Tasa de Reconocimiento	98.64%	98.65	98.98%	99.66%	98.81%
Tiempo de Entrenamiento	2’38”	2’14”	2’19”	2’33”	2’01”
Tiempo de Reconocimiento	41”	1’04”	1’02”	48”	37”

En la tabla 4.2 se muestra la tasa de reconocimiento de voz de cada una de las veinte palabras con diferente número de coeficientes MFCC a una $Fm = 8kHz$. En la parte inferior de la tabla se muestra la tasa máxima, mínima y promedio de cada número de coeficientes MFCC, además del tiempo de entrenamiento y de reconocimiento.

Al igual que para la extracción de características LPC con $Fm = 8kHz$, la tasa de reconocimiento máxima para MFCC con $Fm = 8kHz$ fue del 100% para cada conjunto de coeficientes (8-12), . Por lo que también al menos una de las palabras en cada conjunto de coeficientes MFCC tuvo el 100% de reconocimiento.

La figura 4.24 muestra la tasa de reconocimiento mínima por número de coeficientes MFCC con $Fm = 8kHz$. Se puede observar que fue mayor para 11 coeficientes MFCC con 96.6% de reconocimiento y para el resto de coeficientes MFCC la tasa mínima de reconocimiento fue 93.3%. Esto demuestra que la extracción de características MFCC de alguna o varias de las palabras con $Fm = 8kHz$ y cierto número de coeficientes, tiene por lo menos una tasa de reconocimiento del 93.3%. Las palabras ¨Cero¨ con 8 y 12 MFCC's, "Bajar" con 9 MFCC's y "Enviar" con 9 y 10 MFCC's a una $Fm = 8kH$ tuvieron la tasa mínima de reconocimiento igual a 93.3%. La tasa de reconocimiento mínima fue mayor para MFCC con $Fm = 8kH$ que para LPC a la misma frecuencia de muestreo.

La figura 4.25 muestra la tasa de reconocimiento promedio por número de coeficientes MFCC con $Fm = 8kHz$. Se puede observar que la tasa de reconocimiento de los cinco conjuntos de coeficientes varió entre 98.64% y 99.66%, siendo 8 coeficientes MFCC la tasa de menor certeza y 11 coeficientes MFCC la de mayor tasa de reconocimiento. Las tasas de reconocimiento para el resto de coeficientes MFCC variaron entre 99.65% y 98.98%. Está gráfica muestra que el promedio de reconocimiento para los cinco conjuntos de coeficientes MFCC supera el 98.64% y que el rango de las tasas de reconocimiento promedio supera al rango de LPC con $Fm = 8kHz$.

Las palabras con la mayor tasa de reconocimiento para MFCC con $Fm = 8kHz$ fueron "Uno", "Dos", "Cinco", "Ocho", "Nueve", "Encender", "Apagar", "Colgar" y "Buscar" con 100%, la palabra con menor tasa de reconocimiento fue "Cuatro" con 87.3%, estas tasas de reconocimiento se obtuvieron del promedio de los cinco conjuntos de coeficientes MFCC de cada palabra. Nueve de las veinte palabras fueron reconocidas al 100% con los diferentes números de coeficientes MFCC establecidos (8-12), mientras que con la extracción de características con LPC a una $Fm = 8kHz$ ninguna palabra fue reconocida al 100% para los mismos números de coeficientes LPC.

El tiempo de entrenamiento para MFCC con $Fm = 8kHz$ fue menor para la configuración de 12 coeficientes LPC (2'01") y mayor para para 8 coeficientes MFCC (2'38"). Mientras que el tiempo de reconocimiento para MFCC con $Fm = 8kHz$ fue menor para la configuración de 12 coeficientes LPC (37") y mayor para 9 coeficientes MFCC (1'04"). Tanto el tiempo de entrenamiento como el de reconocimiento es menor en el caracterización con MFCC que para LPC, ambas con $Fm = 8kHz$.

Al comparar las tablas 4.1 y 4.2, y las figuras 4.24 y 4.25. Se puede observar que el método de extracción de características MFCC con $Fm = 8kHz$ tiene una tasa de reconocimiento mayor que para el método LPC con $Fm = 8kHz$. Además, los tiempos de procesamiento son más rápidos para la técnica MFCC.

En la tabla 4.3 se muestra la tasa de reconocimiento de voz de cada una de las veinte palabras con diferente número de coeficientes LPC a una $Fm = 16kHz$. En la parte inferior de la tabla se muestra la tasa máxima, mínima y promedio de cada número de coeficientes LPC, además del tiempo de entrenamiento y de reconocimiento.

Tabla 4.3 Tasa de reconocimiento de voz usando LPC con una $Fm = 16kHz$ y diferente número de coeficientes.

	No. de Coeficientes LPC para Fm=16kHz				
Palabra	**16**	**18**	**20**	**22**	**24**
Cero	90%	83.3%	50%	93.3%	80%
Uno	63.3%	80%	70%	70%	76.6%
Dos	100%	96.6%	100%	90%	86.6%
Tres	90%	60%	80%	73.3%	90%
Cuatro	80%	83.3%	90%	56.6%	66.6%
Cinco	86.6%	90%	93.3%	96.6%	93.3%
Seis	93.3%	83.3%	83.3%	90%	66.6%
Siete	96.6%	90%	90%	93.3%	90%
Ocho	83.3%	83.3%	83.3%	83.3%	76.6%
Nueve	73.3%	90%	83.3%	83.3%	80%
Encender	86.6%	86.6%	73.3%	86.6%	93.3%
Apagar	90%	86.6%	93.3%	90%	66.6%
Subir	93.3%	100%	73.3%	86.6%	70%
Bajar	96.6%	76.6%	93.3%	76.6%	53.3%
Volumen	83.3%	76.6%	86.6%	73.3%	76.6%
Llamar	73.3%	90%	80%	80%	33.3%
Colgar	86.6%	63.3%	43.3%	50%	50%
Enviar	90%	80%	76.6%	80%	76.6%
Mensaje	93.3%	80%	83.3%	73.3%	80%
Buscar	60%	90%	70%	93.3%	80%
Tasa de Reconocimiento Máxima	100%	100%	100%	93.3%	93.3%
Tasa de Reconocimiento Mínima	60%	60%	43.3%	50%	33.3%
Promedio de la Tasa de Reconocimiento	85.47%	83.47	79.81%	80.97%	74.3%
Tiempo de Entrenamiento	3’04”	3’08”	3’05”	3’19”	3’23”
Tiempo de Reconocimiento	40”	45”	45”	55”	57”

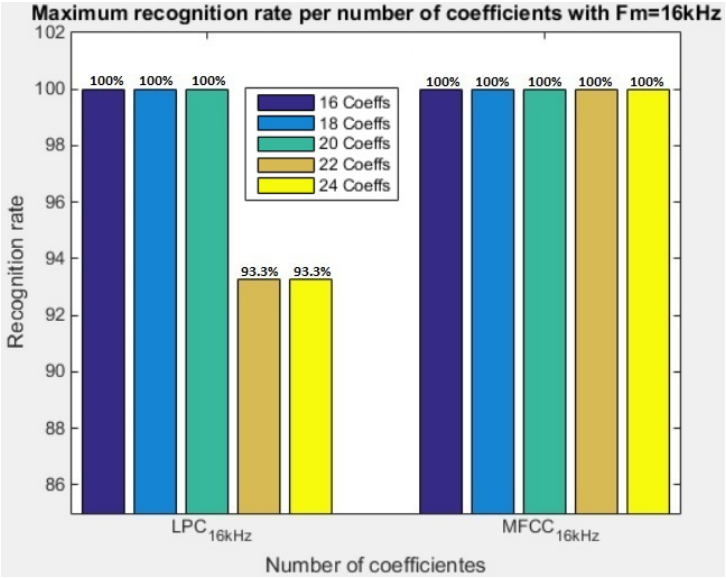


Figura 4.26 Tasa de reconocimiento máxima por número de coeficientes LPC y MFCC con $Fm = 16kH$.

La tasa de reconocimiento máxima por número de coeficientes LPC con $Fm = 16kHz$ fue del 100% para algunas de las palabras con 16, 18 y 20 coeficientes y de 93.3% para algunas palabras con 22 y 24 coeficientes característicos, como se muestra en la figura 4.26. Esto indica que a menor cantidad de coeficientes LPC con $Fm = 16kHz$ mayor tasa de reconocimiento.

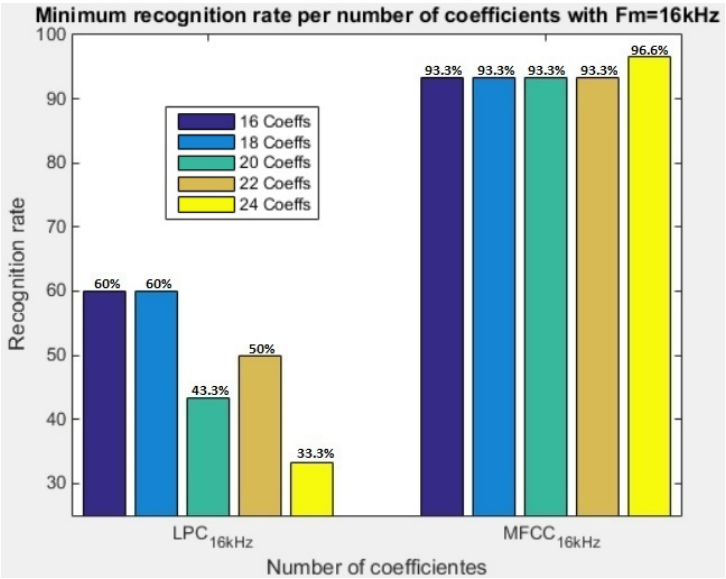


Figura 4.27 Tasa de reconocimiento mínima por número de coeficientes LPC y MFCC con $Fm = 16kH$.

La figura 4.27 muestra la tasa de reconocimiento mínima por número de coeficientes LPC con $Fm = 16kHz$. Se puede observar que fue menor para 24 coeficientes LPC con 33.3% de reconocimiento, la mayor tasa fue de 60% para 16 y 18 coeficientes LPC y para el resto de los conjuntos de coeficientes LPC la tasa de reconocimiento varió entre 43.3% y 50%. Esto demuestra que la extracción de características LPC con frecuencia de muestreo de $16kHz$ presenta una menor tasa de reconocimiento que para LPC con $Fm = 8kHz$.

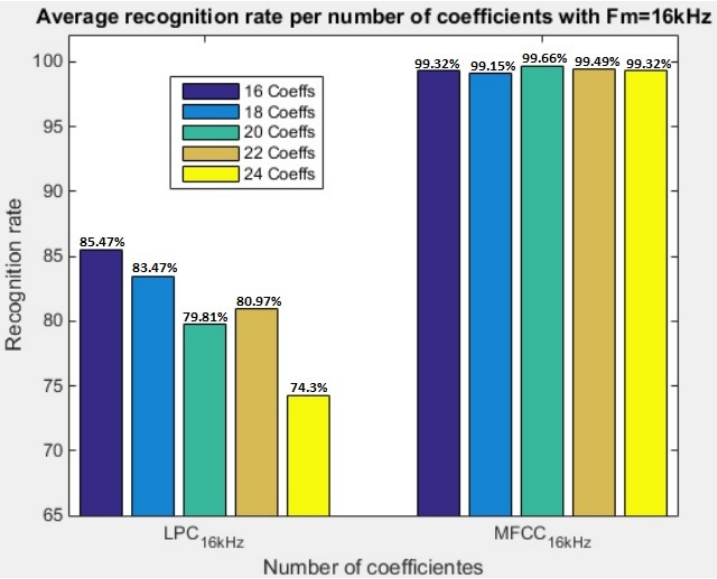


Figura 4.28 Tasa de reconocimiento promedio por número de coeficientes LPC y MFCC con $Fm = 16kH$.

La figura 4.28 muestra la tasa de reconocimiento promedio por número de coeficientes LPC con $Fm = 16kHz$. Se observa que la tasa varió entre 74.3% y 85.47%, siendo 24 coeficientes LPC la tasa de menor certeza y 16 coeficientes LPC la de mayor reconocimiento. Está gráfica muestra que el promedio de reconocimiento para los cinco conjuntos de coeficientes LPC supera el 74.3% y confirma que la tasa de reconocimiento es mayor para LPC con $Fm = 8kHz$.

El tiempo de entrenamiento para LPC con $Fm = 16kHz$ fue menor para 16 coeficientes LPC (3'04") y el mayor para 24 coeficientes LPC (3'23"). El tiempo de reconocimiento fue menor para 16 coeficientes LPC (40") y el mayor para 24 coeficientes LPC (57").

La palabra con la mayor tasa de reconocimiento para LPC con $Fm = 16kHz$ fue "Dos" con 94.64% y la palabra con menor tasa de reconocimiento fue "Colgar" con 58.64%. Estas tasas de reconocimiento se obtuvieron promediando los cinco conjuntos de coeficientes LPC con $Fm = 16kHz$ de cada palabra.

En la tabla 4.4 se muestra la tasa de reconocimiento de voz de cada una de las veinte palabras con diferente número de coeficientes MFCC a una $Fm = 16kHz$. En la parte inferior de la tabla se muestra la tasa máxima, mínima y promedio de cada número de coeficientes MFCC, además del tiempo de entrenamiento y de reconocimiento.

La tasa de reconocimiento máxima para MFCC con $Fm = 16kHz$ fue del 100% para cada conjunto de coeficientes (16-24) como se muestra en la figura 4.26. Indicando que al menos una de las palabras en cada conjunto de coeficientes MFCC tuvo el 100% de reconocimiento.

La figura 4.27 muestra la tasa de reconocimiento mínima por número de coeficientes MFCC con $Fm = 16kHz$. Se observa que la tasa de reconocimiento fue mayor para 24 coeficientes MFCC con 96.6% y para el resto de coeficientes MFCC la tasa fue 93.3%. Se obtivieron los mismos porcentajes para MFCC con $Fm = 8kHz$. Esto demuestra que la extracción de características MFCC con $Fm = 16kHz$ y con cierto número de coeficientes de alguna palabra, tiene por lo menos una tasa de reconocimiento del 93.3%. Las palabras ¨Cero¨ con 16 MFCC's y "Mensaje" con 18, 20 y 22 MFCC's a una $Fm = 16kH$ tuvieron la tasa mínima de 93.3%.

La figura 4.28 muestra la tasa de reconocimiento promedio por número de coeficientes MFCC con $Fm = 16kHz$. Se puede observar que el reconocimiento varió entre 99.15% y 99.66%, siendo 18 coeficientes MFCC la tasa de menor certeza y 20 coeficientes MFCC la de mayor tasa de reconocimiento. Las tasas de reconocimiento para el resto de coeficientes MFCC variaron entre 99.32% y 99.49%. Está gráfica muestra que el promedio de reconocimiento para los cinco conjuntos de coeficientes MFCC supera el 99.15%, mayor que MFCC con Fm=8kHz.

Tabla 4.4 Tasa de reconocimiento de voz usando MFCC con una $Fm = 16kHz$ y diferente número de coeficientes.

	No. de Coeficientes MFCC para Fm=16kHz				
Palabra	**16**	**18**	**20**	**22**	**24**
Cero	93.3%	96.6%	100%	100%	96.6%
Uno	100%	100%	100%	96.6%	96.6%
Dos	100%	100%	100%	100%	100%
Tres	100%	100%	100%	100%	100%
Cuatro	100%	100%	100%	100%	100%
Cinco	100%	100%	100%	100%	100%
Seis	100%	100%	100%	100%	100%
Siete	100%	96.6%	100%	100%	100%
Ocho	100%	96.6%	100%	100%	96.6%
Nueve	100%	100%	100%	100%	100%
Encender	100%	100%	100%	100%	100%
Apagar	96.6%	100%	100%	100%	100%
Subir	100%	100%	100%	100%	100%
Bajar	100%	100%	100%	100%	100%
Volumen	100%	100%	100%	100%	100%
Llamar	100%	100%	100%	100%	100%
Colgar	100%	100%	100%	100%	100%
Enviar	100%	100%	100%	100%	100%
Mensaje	96.6%	93.3%	93.3%	93.3%	96.6%
Buscar	100%	100%	100%	100%	100%
Tasa de Reconocimiento Máxima	100%	100%	100%	100%	100%
Tasa de Reconocimiento Mínima	93.3%	93.3%	93.3%	93.3%	96.6%
Promedio de la Tasa de Reconocimiento	99.32%	99.15	99.66%	99.49%	99.32%
Tiempo de Entrenamiento	2’28”	2’21”	2’27”	2’09”	2’09”
Tiempo de Reconocimiento	48”	46”	43”	45”	48”

La mayoría de las palabras caracterizadas con MFCC a una $Fm = 16kHz$ tuvieron el 100% de reconocimiento excepto "Cero", "Uno", "Siete", "Ocho", "Apagar" y "Mensaje", siendo esta última la de menor tasa de reconocimiento con 94.62%. Estas tasas de reconocimiento se obtuvieron del promedio de los cinco conjuntos de coeficientes MFCC de cada palabra con frecuencia de muestreo de $16kHz$.

El tiempo de entrenamiento para MFCC con $Fm = 16kHz$ fue menor para 22 y 24 coeficientes MFCC (2'09") y el mayor para 16 coeficientes MFCC (2'28"). El tiempo de reconocimiento fue menor para 20 coeficientes MFCC (43") y el mayor para 16 y 24 coeficientes MFCC (48").

A partir de las gráficas y tablas obtenidas en los resultados se puede observar que la técnica MFCC presentó una tasa de reconocimiento mayor a la de la técnica LPC para ambas frecuencias de muestreo. En cuanto al tiempo de procesamiento, también fue menor el tiempo de entrenamiento para la técnica MFCC que la de LPC en ambas frecuencias de muestreo. Mientras que el tiempo de reconocimiento fue relativamente igual para las cuatro configuraciones de extracción de características. De manera general la técnica MFCC con $Fm = 16kHz$, presentó la mayor tasa de reconocimiento de voz para el sistema de RAV que se diseñó.

Conclusiones

De la presente investigación se puede concluir que en un sistema de reconocimiento de voz cada una de las etapas que lo conforman es indispensable. El omitir alguna de ellas afecta directamente en la tasa de reconocimiento. Como ya se mencionó a lo largo del documento, está investigación se enfocó en la etapa de extracción de características, en la que las técnicas de Codificación Predictiva Lineal (LPC) y la de Coeficientes Cepstrales de Frecuencias Mel (MFCC) fueron evaluadas con diferentes frecuencias de muestreo y diferente número de coeficientes.

Los objetivos establecidos en este investigación se cumplieron satisfactoriamente:

- Se codificaron y se compararon las técnicas de extracción de características LPC y MFCC.
- Se obtuvo la tasa de reconocimiento de voz para ambas técnicas de extracción de características a una frecuencia de muestreo de $8kHz$ y haciendo variar los coeficientes de forma unitaria entre 8 a 12.
- Se obtuvo la tasa de reconocimiento de voz a una $Fm = 16kHz$ en la que los coeficientes fueron el doble que para la $Fm = 8kHz$, los cuales fueron variando de dos en dos entre 16 y 24. Esta selección de coeficientes se hizo para tener la misma distribución frecuencial en las bandas de $0 - 4kHz$ y que fuera una comparación objetiva de la existencia de una mejora o disminución en la tasa de reconocimiento al duplicar la frecuencia de muestreo.
- Se modelaron satisfactoriamente las matrices de coeficientes característicos tanto de LPC como MFCC mediante los modelos ocultos de Makov.

De los resultados obtenidos en esta investigación se puede concluir que la técnica de extracción de características MFCC tiene una tasa de reconocimiento de voz mayor a la de LPC bajo las condiciones en que se desarrolló la investigación, tales como; el ambiente de grabación del corpus de voz, las frecuencias de muestreo de las señales de voz y los números de coeficientes con los que se caracterizó a las palabras que conformaron el corpus de voz.

La extracción de características con MFCC tuvo resultados muy similares para las dos configuraciones de frecuencia de muestreo ($8y16kHz$). Caso contrario a la técnica LPC, en la cual, para una $Fm = 16kHz$ la tasa de reconocimiento estuvo muy por debajo de la obtenida con LPC a una $Fm = 8kHz$. De lo que se puede concluir, que a mayor frecuencia de muestreo y mayor número de coeficientes, la extracción de características con LPC presenta una menor tasa de reconocimiento.Por otro lado, la extracción de características con MFCC presta una mayor tasa de reconocimiento de voz a mayor frecuencia de muestreo y mayor numero de coeficientes característicos.

El tiempo de procesamiento en el método MFCC fue menor que en el método LPC para ambas frecuencias de muestreo. Es importante destacar que a mayor frecuencia de muestreo y mayor número de coeficientes el tiempo de procesionalmente es menor. Este es un factor importante a considerar en los sistemas de reconocimiento de voz en tiempo real.

Como trabajo futuro, se puede aumentar el número de locutores, el vocabulario y el corpus de voz. También se puede implementar un sistema de reconocimiento de voz con estas técnicas de extracción de características LPC y MFCC con frecuencias de muestreo de $8y16kHz$ y sus respectivas configuración de coeficientes, en alguna tarjeta digital o DSP y comparar si se obtienen los mismos resultados que los obtenidos en las simulaciones de esta investigación.

Apéndice A: Códigos de los filtros

Filtro pasa bajas para señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral para
%        Reconocimiento de Voz en Idioma Español
% Function: Low-pass filter 8kHz

function [voice_lpf_8] = LPF_8(voice,Fs,t)

% Values got from Filter Dising & Analysis tool (Matlab app)
G = [0.803204478848642; 0.690323717680132; 0.614593862083496; ...
     0.566239409020991; 0.539322213907318; 0.728477792361996; 1];
    % Scale values (Gain)

SOS = [1, 2, 1, 1, 1.38323978903067, 0.829578126363900;...
       1, 2, 1, 1, 1.18884201813157, 0.572452852588962; ...
       1, 2, 1, 1, 1.05842373457198, 0.399951713762007; ...
       1, 2, 1, 1, 0.975150041242042, 0.289807594841921; ...
       1, 2, 1, 1, 0.928794553603693, 0.228494302025578; ...
       1, 1, 0, 1, 0.456955584723992, 0];
    % SOS(second-order section)matrix

% Convert digital filter SOS data to transfer function form
[b_lpf_8,a_lpf_8] = sos2tf(SOS,G); % Returns the transfer function
% that describes a discrete-time system given by SOS in second-order
% section form with gain G

% Low-pass filter
voice_lpf_8 = filter(b_lpf_8,a_lpf_8,voice);
%Filters the voice signal

%% Plotting
% Low-pass filter voice signal plot
figure
subplot(2,1,1);plot(t,voice_lpf_8)
title('Voice signal filtered with a Low-pass filter');
xlabel('Time (sec)'); ylabel('Amplitude');

% STFT parameters (Short-Time Fourier Transfor)
Frm_width = 256;                    % # of samples per frame
Win_lngth = round(0.032*Fs);        % Window Length
```

```
Win_Hamm = hamming(Win_lngth);     % Hamming Window
Noverlap = floor(0.50*Win_lngth);  % Overlapping

% Spectrogram computed by the Matlab spectrogram function.
[S_lpf_8, F_lpf_8, T_lpf_8] = spectrogram(voice_lpf_8 + 1i*eps, ...
                                  Win_Hamm, Noverlap, Frm_width,Fs);
% Spectrogram plot
subplot(2,1,2)
t_epsilon = 0.01;
S_one_sided_lpf_8 = max(S_lpf_8(1:length(F_lpf_8)/2,:),t_epsilon);
pcolor(T_lpf_8,F_lpf_8(1:end/2),10*log10(abs(S_one_sided_lpf_8)));
shading interp; colormap('hot');
title('Spectrogram of the voice signal filtered with a Low-pass filter');
xlabel('Time (sec)'); ylabel('Frequency (Hz)');
end
```

Filtro pasa bajas para señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral para
%        Reconocimiento de Voz en Idioma Español
% Function: Low-pass filter 16kHz

function [voice_lpf_16] = LPF_16(voice,Fs,t)

% Values got from Filter Dising & Analysis tool (Matlab app)
G = [0.826357836779560; 0.763490243178792; 0.710703805171656; ...
     0.666471259402615; 0.629543207224940; 0.598906940414046; ...
     0.573748536516509; 0.553420359331216; 0.537414383538911; ...
     0.525341015476816; 0.516912825972202; 0.511932592757750; ...
     0.714342452574764; 1]; % Scale values (Gain)

SOS = [ 1, 2, 1, 1, 1.38842816857824, 0.917003178540006;...
        1, 2, 1, 1, 1.28279942766110, 0.771161545054072;...
        1, 2, 1, 1, 1.19410882150234, 0.648706399184283;...
        1, 2, 1, 1, 1.11979027597611, 0.546094761634352;...
        1, 2, 1, 1, 1.05774457909735, 0.460428249802410;...
        1, 2, 1, 1, 1.00627020089566, 0.389357560760528;...
        1, 2, 1, 1, 0.963999606858648, 0.330994539207388;...
        1, 2, 1, 1, 0.929844652958889, 0.283836784365974;...
        1, 2, 1, 1, 0.902951766286181, 0.246705767869463;...
        1, 2, 1, 1, 0.882666360181298, 0.218697701725965;...
        1, 2, 1, 1, 0.868505502502589, 0.199145801386220;...
```

```
        1, 2, 1, 1, 0.860137824756612, 0.187592546274388;...
        1, 1, 0, 1, 0.428684905149528, 0];
    % SOS(second-order section)matrix

% Convert digital filter SOS data to transfer function form
[b_lpf_16,a_lpf_16] = sos2tf(SOS,G); % Returns the transfer function
% that describes a discrete-time system given by SOS in second-order
% section form with gain G

% Low-pass filter
voice_lpf_16 = filter(b_lpf_16,a_lpf_16,voice);
%Filters the voice signal

%% Plotting
% Low-pass filter voice signal plot
figure
subplot(2,1,1);plot(t,voice_lpf_16)
title('Voice signal filtered with a Low-pass filter');
xlabel('Time (sec)'); ylabel('Amplitude');

% STFT parameters (Short-Time Fourier Transfor)
Frm_width = 512;                    % # of samples per frame
Win_lngth = round(0.032*Fs);        % Window Length
Win_Hamm = hamming(Win_lngth);      % Hamming Window
Noverlap = floor(0.50*Win_lngth);  % Overlapping

% Spectrogram computed by the Matlab spectrogram function.
[S_lpf_16, F_lpf_16, T_lpf_16] = spectrogram(voice_lpf_16 + 1i*eps, ...
                                 Win_Hamm, Noverlap, Frm_width,Fs);
% Spectrogram plot
subplot(2,1,2)
t_epsilon = 0.01;
S_one_sided_lpf_16 = max(S_lpf_16(1:length(F_lpf_16)/2,:),t_epsilon);
pcolor(T_lpf_16,F_lpf_16(1:end/2),10*log10(abs(S_one_sided_lpf_16)));
shading interp; colormap('hot');
title('Spectrogram of the voice signal filtered with a Low-pass filter');
xlabel('Time (sec)'); ylabel('Frequency (Hz)');
end
```

Filtro pasa altas para señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral para
```

```
%        Reconocimiento de Voz en Idioma Español
% Function: High-pass filter 8kHz

function [voice_hpf_8] = HPF_8(voice_lpf_8,Fs,t)

% Values got from Filter Dising & Analysis tool (Matlab app)
G = [0.931837731030737; 0.851216765530748; 0.907736999893535; 1];
   % Scale values (Gain)

SOS = [1, -2, 1, 1, -1.84442215416095, 0.882928769962000; ...
       1, -2, 1, 1, -1.68484598557895, 0.720021076544038; ...
       1, -1, 0, 1, -0.815473999787071, 0];
   % SOS(second-order section)matrix

% Convert digital filter SOS data to transfer function form
[b_hpf_8,a_hpf_8] = sos2tf(SOS,G); % Returns the transfer function
% that describes a discrete-time system given by SOS in second-order
% section form with gain G

% High-pass filter
voice_hpf_8 = filter(b_hpf_8,a_hpf_8,voice_lpf_8);
%Filters the voice signal

%% Plotting
% High-pass filter voice signal plot
figure
subplot(2,1,1); plot(t,voice_hpf_8)
title('Voice signal filtered with a High-pass filter');
xlabel('Time (sec)'); ylabel('Amplitude');

% STFT parameters (Short-Time Fourier Transfor)
Frm_width = 256;                    % # of samples per frame
Win_lngth = round(0.032*Fs);        % Window Length
Win_Hamm = hamming(Win_lngth);      % Hamming Window
Noverlap = floor(0.50*Win_lngth);  % Overlapping

% Spectrogram computed by the Matlab spectrogram function.
[S_hpf_8, F_hpf_8, T_hpf_8] = spectrogram(voice_hpf_8 + 1i*eps, ...
                                  Win_Hamm, Noverlap, Frm_width, Fs);
% Spectrogram ploting
subplot(2,1,2)
t_epsilon = 0.01;
S_one_sided_hpf_8 = max(S_hpf_8(1:length(F_hpf_8)/2,:),t_epsilon);
```

```
pcolor(T_hpf_8,F_hpf_8(1:end/2),10*log10(abs(S_one_sided_hpf_8)));
shading interp; colormap('hot');
title('Spectrogram of a voice signal filtered with a High-pass filter');
xlabel('Time (sec)'); ylabel('Frequency (Hz)');
end
```

Filtro pasa altas para señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral para
%        Reconocimiento de Voz en Idioma Español
% Function: High-pass filter 16kHz

function [voice_hpf_16] = HPF_16(voice_lpf_16,Fs,t)

% Values got from Filter Dising & Analysis tool (Matlab app)
G = [0.967131603004853; 0.921831031296358; 0.951644961116850; 1];
    % Scale values (Gain)

SOS = [1, -2, 1, 1, -1.92926919909240, 0.939257212927016; ...
       1, -2, 1, 1, -1.83890197561740, 0.848422149568037; ...
       1, -1, 0, 1, -0.903289922233700, 0];
    % SOS(second-order section)matrix

% Convert digital filter SOS data to transfer function form
[b_hpf_16,a_hpf_16] = sos2tf(SOS,G); % Returns the transfer function
% that describes a discrete-time system given by SOS in second-order
% section form with gain G

% High-pass filter
voice_hpf_16 = filter(b_hpf_16,a_hpf_16,voice_lpf_16);
%Filters the voice signal

%% Plotting
% High-pass filter voice signal plot
figure
subplot(2,1,1); plot(t,voice_hpf_16)
title('Voice signal filtered with a High-pass filter');
xlabel('Time (sec)'); ylabel('Amplitude');

% STFT parameters (Short-Time Fourier Transfor)
Frm_width = 512;                    % # of samples per frame
Win_lngth = round(0.032*Fs);        % Window Length
```

```
Win_Hamm = hamming(Win_lngth);     % Hamming Window
Noverlap = floor(0.50*Win_lngth);  % Overlapping

% Spectrogram computed by the Matlab spectrogram function.
[S_hpf_16, F_hpf_16, T_hpf_16] = spectrogram(voice_hpf_16 + 1i*eps, ...
                                  Win_Hamm, Noverlap, Frm_width, Fs);
% Spectrogram ploting
subplot(2,1,2)
t_epsilon = 0.01;
S_one_sided_hpf_16 = max(S_hpf_16(1:length(F_hpf_16)/2,:),t_epsilon);
pcolor(T_hpf_16,F_hpf_16(1:end/2),10*log10(abs(S_one_sided_hpf_16)));
shading interp; colormap('hot');
title('Spectrogram of a voice signal filtered with a High-pass filter');
xlabel('Time (sec)'); ylabel('Frequency (Hz)');
end
```

Apéndice B: Códigos de detección de voz

Código de detección de voz para señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Voice Activity Detection (beginning and end of the word) 8kHz

function [word_8] = vad_8(voice_hpf_8,Fs)

N = length(voice_hpf_8);          % # of samples of the signal
Frame_lngth = round(0.032*Fs);    % Frame Length

% Compute of the signal averaged energy
average_energy = sum(voice_hpf_8.*voice_hpf_8)/N;

% Loop to compute the energy of each frame
threshold = 0.0015; % Threshold
j = 1;            % # of iterations of the loop

for i = 1:Frame_lngth:N-Frame_lngth; % Frame index
    frame = voice_hpf_8(i:i+255);   % Frame by frame (256smpls)
    frame_energy = sum(frame.*frame)/Frame_lngth; % Frame energy
    % If the frame energy exceeds the product of the
    % average energy and the threshold...
    if  (frame_energy> threshold*average_energy);
        f(j) = i; %...then the frame index is increased by one...
    % F is a vector that saves de position (not the value) of the first
    % sample of the frame that exceeds the threshold
        j = j+1;  % ...and the iteration is increased by one too.
    end;
end

num_frames = length(f); % # of frames that exceed the threshold
frame_counter = 0;      % Frame counter
p = 0; % Index of decision of "beginning" and "end"

% Loop to define the beginning and end of the word
for i = 1:num_frames-1
    if frame_counter == 0 % If it's equal to zero...
       start = f(i);     % Could be the beginning of the word
    end
```

```
    dtns_btn_frms = f(i+1)-f(i); % compute the distance between frames

    if dtns_btn_frms<10000 % If distance is less than 10000 samples
       frame_counter = frame_counter+1; % frame counter is increased by one
    else % If distance is more than 10000 samples
         endd = f(i); % the frame index corresponds to the end of the word
         if frame_counter<10 % if the # of frames is less than 10...
            frame_counter = 0 ; % the Frame counter is equal to zero...
            start = 0; % the frame is not the beginning of the word...
            endd = 0;    % nor the end either
         else % if the # of frames is more than 10...
            p = 1; % the index are taken as the beginning and end index
         break;
         end
    end
end
if p == 0 % If the are no more sound elements of beginning and end (index)
   start = f(1) ; % the first value is the beginning
   endd = f(end); % and the last value is the end
end

word_8 = voice_hpf_8(1,start:endd);
% Segments the signal with the found indexes

%% Plotting
N = length(word_8);
t = 0:1/Fs:((N-1)/Fs);             % Time Vector
figure
subplot(2,1,1); plot(t,word_8); title('Word');
xlabel('Time (sec)'); ylabel('Amplitude');

% STFT parameters (Short-Time Fourier Transfor)
Frm_width = 256;                   % # of samples per frame
Win_lngth = round(0.032*Fs);       % Window Length
Win_Hamm = hamming(Win_lngth);     % Hamming Window
Noverlap = floor(0.50*Win_lngth);  % Overlapping

% Spectrogram computed by the Matlab spectrogram function.
[S_vad_8, F_vad_8, T_vad_8] = spectrogram(word_8 + 1i*eps, ...
                                Win_Hamm, Noverlap, Frm_width, Fs);

% Spectrogram plot
subplot(2,1,2)
```

```
t_epsilon = 0.01;
S_one_sided = max(S_vad_8(1:length(F_vad_8)/2,:),t_epsilon);
pcolor(T_vad_8,F_vad_8(1:end/2),10*log10(abs(S_one_sided)));
shading interp; colormap('hot');
title('Word spectrogram'); xlabel('Time (sec)'); ylabel('Frequency (Hz)');
end
```

Código de detección de voz para señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Voice Activity Detection (beginning and end of the word) 16kHz

function [word_16] = vad_16(voice_hpf_16,Fs)

N = length(voice_hpf_16);          % # of samples of the signal
Frame_lngth = round(0.032*Fs);     % Frame Length

% Compute of the signal averaged energy
average_energy = sum(voice_hpf_16.*voice_hpf_16)/N;

% Loop to compute the energy of each frame
threshold = 0.0015; % Threshold
j = 1;            % # of iterations of the loop

for i = 1:Frame_lngth:N-Frame_lngth; % Frame index
    frame = voice_hpf_16(i:i+511);   % Frame by frame (512smpls)
    frame_energy = sum(frame.*frame)/Frame_lngth; % Frame energy
    % If the frame energy exceeds the product of the
    % average energy and the threshold...
    if  (frame_energy> threshold*average_energy);
        f(j) = i; %...then the frame index is increased by one...
    % F is a vector that saves de position (not the value) of the first
    % sample of the frame that exceeds the threshold
        j = j+1;  % ...and the iteration is increased by one too.
    end;
end

num_frames = length(f); % # of frames that exceed the threshold
frame_counter = 0;      % Frame counter
p = 0; % Index of decision of "beginning" and "end"
```

```
% Loop to define the beginning and end of the word
for i = 1:num_frames-1
    if frame_counter == 0 % If it's equal to zero...
       start = f(i);     % Could be the beginning of the word
    end
    dtns_btn_frms = f(i+1)-f(i); % compute the distance between frames

    if dtns_btn_frms<10000 % If distance is less than 10000 samples
       frame_counter = frame_counter+1; % frame counter is increased by one
    else % If distance is more than 10000 samples
         endd = f(i); % the frame index corresponds to the end of the word
         if frame_counter<10 % if the # of frames is less than 10...
            frame_counter = 0 ; % the Frame counter is equal to zero...
            start = 0; % the frame is not the beginning of the word...
            endd = 0;    % nor the end either
         else % if the # of frames is more than 10...
            p = 1; % the index are taken as the beginning and end index
         break;
         end
    end
end
if p == 0 % If the are no more sound elements of beginning and end (index)
   start = f(1) ; % the first value is the beginning
   endd = f(end); % and the last value is the end
end

word_16 = voice_hpf_16(1,start:endd);
% Segments the signal with the found indexes

%% Plotting
N = length(word_16);
t = 0:1/Fs:((N-1)/Fs);             % Time Vector
figure
subplot(2,1,1); plot(t,word_16); title('Word');
xlabel('Time (sec)'); ylabel('Amplitude');

% STFT parameters (Short-Time Fourier Transfor)
Frm_width = 512;                   % # of samples per frame
Win_lngth = round(0.032*Fs);       % Window Length
Win_Hamm = hamming(Win_lngth);     % Hamming Window
Noverlap = floor(0.50*Win_lngth);  % Overlapping

% Spectrogram computed by the Matlab spectrogram function.
```

```
[S_vad_16, F_vad_16, T_vad_16] = spectrogram(word_16 + 1i*eps, ...
                                 Win_Hamm, Noverlap, Frm_width, Fs);

% Spectrogram plot
subplot(2,1,2)
t_epsilon = 0.01;
S_one_sided = max(S_vad_16(1:length(F_vad_16)/2,:),t_epsilon);
pcolor(T_vad_16,F_vad_16(1:end/2),10*log10(abs(S_one_sided)));
shading interp; colormap('hot');
title('Word spectrogram'); xlabel('Time (sec)'); ylabel('Frequency (Hz)');
end
```

Apéndice C: Códigos de pre-énfasis

Código de pre-énfasis para señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Pre-emphasis 8kHz

function [enfatiz_8] = pre_enf_8(word_8,Fs)

   N = length(word_8);          % # of samples of the signal
   enfatiz_8 = zeros(1,N-1);
   for i=2:1:N
       enfatiz_8(i)=word_8(i)-0.96875*word_8(i-1);
   end

%% Plotting
N = length(enfatiz_8);
t = 0:1/Fs:((N-1)/Fs);             % Time Vector
figure
subplot(2,1,1); plot(t,word_16); title('Emphasized word');
xlabel('Time (sec)'); ylabel('Amplitude');

% STFT parameters (Short-Time Fourier Transfor)
Frm_width = 256;                   % # of samples per frame
Win_lngth = round(0.032*Fs);       % Window Length
Win_Hamm = hamming(Win_lngth);     % Hamming Window
Noverlap = floor(0.50*Win_lngth);  % Overlapping

% Spectrogram computed by the Matlab spectrogram function.
[S_enf_8, F_enf_8, T_enf_8] = spectrogram(enfatiz_8 + 1i*eps, ...
                                  Win_Hamm, Noverlap, Frm_width, Fs);
% Spectrogram ploting
subplot(2,1,2)
t_epsilon = 0.01;
S_one_sided = max(S_enf_8(1:length(F_enf_8)/2,:),t_epsilon);
pcolor(T_enf_8,F_enf_8(1:end/2),10*log10(abs(S_one_sided)));
shading interp; colormap('hot');
title('Emphasized word spectrogram');
xlabel('Time (sec)'); ylabel('Frequency (Hz)');
 end
```

Código de pre-énfasis para señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Pre-emphasis 16kHz

function [enfatiz_16] = pre_enf_16(word_16,Fs)

   N = length(word_16);          % # of samples of the signal
   enfatiz_16 = zeros(1,N-1);
   for i=2:1:N
       enfatiz_16(i)=word_16(i)-0.96875*word_16(i-1);
   end

%% Plotting
N = length(enfatiz_16);
t = 0:1/Fs:((N-1)/Fs);             % Time Vector
figure
subplot(2,1,1); plot(t,word_16); title('Emphasized word');
xlabel('Time (sec)'); ylabel('Amplitude');

% STFT parameters (Short-Time Fourier Transfor)
Frm_width = 512;                   % # of samples per frame
Win_lngth = round(0.032*Fs);       % Window Length
Win_Hamm = hamming(Win_lngth);     % Hamming Window
Noverlap = floor(0.50*Win_lngth);  % Overlapping

% Spectrogram computed by the Matlab spectrogram function.
[S_enf_16, F_enf_16, T_enf_16] = spectrogram(enfatiz_16 + 1i*eps, ...
                                 Win_Hamm, Noverlap, Frm_width, Fs);
% Spectrogram ploting
subplot(2,1,2)
t_epsilon = 0.01;
S_one_sided = max(S_enf_16(1:length(F_enf_16)/2,:),t_epsilon);
pcolor(T_enf_16,F_enf_16(1:end/2),10*log10(abs(S_one_sided)));
shading interp; colormap('hot');
title('Emphasized word spectrogram');
xlabel('Time (sec)'); ylabel('Frequency (Hz)');
 end
```

Apéndice D: Códigos de segmentación y enventanado

Código de segmentación y enventanado para señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Windowing and segmentation 8kHz

function [Frms_mtrx] = seg_win_8(enfatiz_8)

% Frame features
Ns = length(enfatiz_8);              % # of samples
Frm_width = 256;                      % Frame width (samples per frame)
Frm_ovlp  = Frm_width/2;              % 50% overlap
Num_frms  = floor(Ns/Frm_ovlp)-1;     % # of frames
%Frm_dratn = Frm_width/Fs;             % frame duration (32 msec)

%%%% Hamming window %%%%
Ns_win = 0:1:Frm_width-1;             % # of samples per window
Hamm_win = 0.54-0.46*cos(2*pi*Ns_win/(Frm_width-1)); % Hamming window

% To prevent error allocating memory
Frms_mtrx = zeros(Num_frms,Frm_width); %zeros matrix to save the frames

for Frame_indx_v = 1:1:Num_frms,
    Frms_mtrx(Frame_indx_v,:) = ...
    enfatiz_8((Frame_indx_v-1)*Frm_ovlp +(1:Frm_width)).*Hamm_win;
end   % Saves each windowed frame in matrix row

%% Plotting
figure
subplot(2,1,1)
plot(Hamm_win)
title('Hamming Window')
xlabel('Number of samples per frame'); ylabel('Amplitud');
xlim([0 256]); ylim([0 1.2])

subplot(2,1,2)
plot(Frms_mtrx(2,:))
title('Windowed Frame')
xlabel('Number of samples'); ylabel('Amplitud'); xlim([0 256]);
```

```
end
```

Código de segmentación y enventanado para señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Windowing and segmentation 16kHz

function [Frms_mtrx] = seg_win_16(enfatiz_16)

% Frame features
Ns = length(enfatiz_16);              % # of samples
Frm_width = 512;                      % Frame width (samples per frame)
Frm_ovlp  = Frm_width/2;              % 50% overlap
Num_frms  = floor(Ns/Frm_ovlp)-1;     % # of frames
%Frm_dratn = Frm_width/Fs;             % frame duration (32 msec)

%%%% Hamming window %%%%
Ns_win = 0:1:Frm_width-1;             % # of samples per window
Hamm_win = 0.54-0.46*cos(2*pi*Ns_win/(Frm_width-1)); % Hamming window

% To prevent error allocating memory
Frms_mtrx = zeros(Num_frms,Frm_width); %zeros matrix to save the frames

for Frame_indx_v = 1:1:Num_frms,
    Frms_mtrx(Frame_indx_v,:) = ...
    enfatiz_16((Frame_indx_v-1)*Frm_ovlp +(1:Frm_width)).*Hamm_win;
end   % Saves each windowed frame in matrix row

%% Plotting
figure
subplot(2,1,1)
plot(Hamm_win)
title('Hamming Window')
xlabel('Number of samples per frame'); ylabel('Amplitud');
xlim([0 512]); ylim([0 1.2])

subplot(2,1,2)
plot(Frms_mtrx(2,:))
title('Windowed Frame')
xlabel('Number of samples'); ylabel('Amplitud'); xlim([0 512]);
end
```

Apéndice E: Códigos para obtener los coeficientes LPC

Código para obtener los coeficientes LPC de señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Feature Extraction with LPC 8kHz

function [Coefs_LPC] = Coeff_LPC_8(Frms_mtrx,p)
%%
[Num_frms,Frm_width] = size(Frms_mtrx);
N = Frm_width;          % # of samples per frame
r = zeros(1,p);         % Vector where the autocorrelation will be saved
Coef_mtrx_LPC = zeros(p+1,Num_frms); % Matrix where the coeficients will
% be saved (rows -> # of frams  cols -> # of coefficients)

for Frame_indx = 1:1:Num_frms,
    vec = Frms_mtrx(Frame_indx,:); % Camputes coefficients frame by frame

% Computes Autocorrelation
   pe = 0;              % Prediction Error
   for k = 0:p
   sum = 0;
   for j = 1:N-k
   sum = sum + vec(j).*vec(j+k); % Computes Autocorrelation
   end;
   if (k==0)
  pe = sum;      % Prediction Error equal to aurocorrelation
          else
  r(k) = sum;    % Saves autocorrelations in "r"
       end
   end

% check power in signal
   if (pe == 0)        % If the prediction error is equal to "0" ...
  % no signal
  pc = 1;         % ... the predicction coeficient is equal to "1"
  return;
   end;

% Compute predictor coefficients
```

```
pc = zeros(1,p+1);   % Vector where the prediction coeff. will be saved
pc(1) = 1;           % The first prediction coeff. is equal to 1

% for each coefficient in turn
   for k = 1:p

% find next coeff from pc[] and r[]
   sum = 0;
   for i = 1:k
   sum = sum - pc(1+k-i) * r(i);
       end
   pc(1+k) = sum/pe; % Compute de predicction coefficientes from r[](k)

% perform recursion on pc[]
   for i = 1:k/2
   pci  = pc(1+i) + pc(1+k) * pc(1+k-i);
   pcki = pc(1+k-i) + pc(1+k) * pc(1+i);
   pc(1+i) = pci;
   pc(1+k-i) = pcki;
   end;

% calculate residual error
   pe = pe * (1.0 - pc(1+k)*pc(1+k));
       if (pe <= 0)
% no power left in signal
          break;
       end
   end

   Coef_mtrx_LPC(:,Frame_indx) = pc; % Saves the coefficients in the matrix

end

Coefs_LPC = Coef_mtrx_LPC;

% figure
% mesh(Coefs_LPC)
% title('LPC Coefficients');
% xlabel('Number of windows'); ylabel('Number of Coeficientes');
% ylabel('Energy of the filter')

end
```

Código para obtener los coeficientes LPC de señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Feature Extraction with LPC 16kHz

function [Coefs_LPC] = Coeff_LPC_16(Frms_mtrx,p)
%%
[Num_frms,Frm_width] = size(Frms_mtrx);
N = Frm_width;          % # of samples per frame
r = zeros(1,p);         % Vector where the autocorrelation will be saved
Coef_mtrx_LPC = zeros(p+1,Num_frms); % Matrix where the coeficients will
% be saved (rows -> # of frams  cols -> # of coefficients)

for Frame_indx = 1:1:Num_frms,
    vec = Frms_mtrx(Frame_indx,:); % Camputes coefficients frame by frame

% Computes Autocorrelation
   pe = 0;              % Prediction Error
   for k = 0:p
   sum = 0;
   for j = 1:N-k
   sum = sum + vec(j).*vec(j+k); % Computes Autocorrelation
   end;
   if (k==0)
  pe = sum;      % Prediction Error equal to aurocorrelation
          else
  r(k) = sum;    % Saves autocorrelations in "r"
       end
   end

% check power in signal
   if (pe == 0)        % If the prediction error is equal to "0" ...
  % no signal
  pc = 1;         % ... the predicction coeficient is equal to "1"
  return;
   end;

% Compute predictor coefficients
pc = zeros(1,p+1);   % Vector where the prediction coeff. will be saved
pc(1) = 1;           % The first prediction coeff. is equal to 1

% for each coefficient in turn
```

```
   for k = 1:p

% find next coeff from pc[] and r[]
   sum = 0;
   for i = 1:k
   sum = sum - pc(1+k-i) * r(i);
       end
   pc(1+k) = sum/pe; % Compute de predicction coefficientes from r[](k)

% perform recursion on pc[]
   for i = 1:k/2
   pci  = pc(1+i) + pc(1+k) * pc(1+k-i);
   pcki = pc(1+k-i) + pc(1+k) * pc(1+i);
   pc(1+i) = pci;
   pc(1+k-i) = pcki;
   end;

% calculate residual error
   pe = pe * (1.0 - pc(1+k)*pc(1+k));
       if (pe <= 0)
% no power left in signal
      break;
       end
   end
   Coef_mtrx_LPC(:,Frame_indx) = pc; % Saves the coefficients in the matrix

end
Coefs_LPC = Coef_mtrx_LPC;

% figure
% mesh(Coefs_LPC)
% title('LPC Coefficients');
% xlabel('Number of windows'); ylabel('Number of Coeficientes');
% ylabel('Energy of the filter')

end
```

Apéndice F: Códigos para obtener los coeficientes MFCC

Código para obtener los coeficientes MFCC de señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Feature Extraction with MFCC 8kHz

function [Coefs_MFCC] = Coeff_MFCC_8(Fs,Frms_mtrx,num_coef)
%% 1st Block - FFT to get the power spectral density

[Num_frms,Frm_width] = size(Frms_mtrx);
N = Frm_width;                % # of samples per frame

DenSpc_mtrx = zeros(Num_frms,Frm_width/2);
% Matrix where the power spectrum of each window will be saved (1 per raw)

for Frame_indx = 1:1:Num_frms,
    vec = Frms_mtrx(Frame_indx,:);  % Analysis window from where the LPC
                                    % coefficients will be get
    % Fast Fourier Transformation (FFT)
    XX     = fft(vec,N);       % Fast Fourier Transform
    XXabs  = abs(XX(1:N/2));   % Magnitud Espectral X(Ns,Nf)
    XXabs2 = XXabs.^2;         % Power Spectrum |X(Ns,Nf)|^2

    DenSpc_mtrx(Frame_indx,:) = XXabs2; % power spectrum matrix
end

% Plotting
% figure
% frec = Fs*(0:127)/N;       % Frequency vector to plot
% plot(frec,DenSpc_mtrx(1,:))
% title('Power spectrum of a window |X(N_s,N_f)|^2')
% xlabel('Frequency (Hz)'); ylabel('Magnitude')
%
% figure
% imagesc(20*log(DenSpc_mtrx')) % espectrograma
% title('Spectrogram of the power spectrum window by window |X(N_s,N_f)|^2')
% ylabel('Number of samples per window'); xlabel('Number of windows')

%% 2nd Block - Filterbank
```

```
% Frequency features
low_freq  = 0;        % Low Frequency
high_freq = Fs/2;     % High Frequency

% Hertz to Mel-scale Limits
low_mel_freq =2595*log10(1+(low_freq/700));   % Low Frequency in Mel-Scale
high_mel_freq =2595*log10(1+(high_freq/700)); % High Frequency in Mel-Scale

% Creating the mel-scaled vector
Mel_vec = linspace(low_mel_freq,high_mel_freq,num_coef+2);

% Frequencies vector, computed from the Mel vector
freq_vec=700*((10.^(Mel_vec/2595))-1); % Separacion lineal de escala mel

% %Plotting
% figure
% plot(freq_vec, Mel_vec);
% title('Frequency to mel-frequency curve');
% xlabel('Frequency'); ylabel('Mel-frequency');

% Convert frequencies to nearest FFT bins
f= zeros(1,num_coef+2);
   for i = 1:num_coef+2
       f(i) = floor((256+1)*freq_vec(i)/Fs);
   end

   % Computes Filterbank
   Bank_mtrx = zeros(length(num_coef),N/2); % Filterbank Matrix
   for m = 2:1:num_coef+1
       for k = 1:N/2
           if k < f(m-1);
              Bank_mtrx(m-1:num_coef,k)=0;

           elseif k >= f(m-1) && k <= f(m)
                  Bank_mtrx(m-1:num_coef,k) = (k - f(m-1))/(f(m) - f(m-1));

           elseif k >= f(m) && k <= f(m+1)
                  Bank_mtrx(m-1:num_coef,k) = (f(m+1) - k)/(f(m+1) - f(m));

           else k > f(m+1);
                Bank_mtrx(m-1:num_coef,k)=0;
           end
```

```
        end
    end

%    %Plotting
%    figure
%    freq = (Fs/2)*linspace(0,1,N/2);
%    plot(freq,Bank_mtrx)
%    title('Filterbank'); xlabel('Frequency (Hz)'); ylabel('Amplitude')
%
%    figure
%    freq = (Fs/2)*linspace(0,1,N/2+1);
%    imagesc(freq,1:num_coef,Bank_mtrx)
%    title('Energy distribution of each filter')
%    xlabel('Frequency (Hz)'); ylabel('Filter')

    % Computes the energy of each filter
    [Num_frms,Frm_width] = size(Frms_mtrx);
    nrgy = zeros(num_coef,Num_frms);

    % Multiply each one of the filters per each one of the windows
    for d = 1:1:Num_frms   % Window loop
        for f = 1:1:num_coef   % Filter loop
            nrgy(f,d) = Bank_mtrx(f,:)*DenSpc_mtrx(d,:)';
        end
    end

%    % Plotting
%    figure
%    plot(nrgy); % Energy per filter
%    title('Energy of each filter in each window')
%    xlabel('Filter'); ylabel('Energy')
%    figure
%    imagesc(nrgy) % espectrogram % Energia por ventana
%    title('Energy of each filter in each window');
%    xlabel('Number of window'); ylabel('Number of Filter')

    %% 3th Block - Logarith
    log_filt = log(nrgy); % Computes the logarithm of each filter

%   % Plotting
%    figure
%    imagesc(log_filt) % espectrogram % Energia por ventana
%    title('Logarithmical energy of each filter in each window');
```

```
%    xlabel('Number of window'); ylabel('Number of Filter')

   %% 4th Block - Discrete Cosine Transform (DCT)

   Coefs_MFCC = dct(log_filt); %  Computes the DCT of each filter

%   % Plotting
% figure
% mesh(Coefs_MFCC)
% title('MFCC');
% xlabel('Number of windows'); ylabel('Number of Coeficientes');
% ylabel('Energy of the filter')
end
```

Código para obtener los coeficientes MFCC de señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Tesis: Comparación de Técnicas de Parametrización Espectral
%        para Reconocimiento de Voz en Idioma Español
% Function: Feature Extraction with MFCC 16kHz

function [Coefs_MFCC] = Coeff_MFCC_16(Fs,Frms_mtrx,num_coef)
%% 1st Block - FFT to get the power spectral density

[Num_frms,Frm_width] = size(Frms_mtrx);
N = Frm_width;                % # of samples per frame

DenSpc_mtrx = zeros(Num_frms,Frm_width/2);
% Matrix where the power spectrum of each window will be saved (1 per raw)

for Frame_indx = 1:1:Num_frms,
    vec = Frms_mtrx(Frame_indx,:);  % Analysis window from where the LPC
                                    % coefficients will be get
    % Fast Fourier Transformation (FFT)
    XX     = fft(vec,N);       % Fast Fourier Transform
    XXabs  = abs(XX(1:N/2));   % Magnitud Espectral X(Ns,Nf)
    XXabs2 = XXabs.^2;         % Power Spectrum |X(Ns,Nf)|^2

    DenSpc_mtrx(Frame_indx,:) = XXabs2; % power spectrum matrix
end

% Plotting
% figure
```

```
% frec = Fs*(0:255)/N;       % Frequency vector to plot
% plot(frec,DenSpc_mtrx(1,:))
% title('Power spectrum of a window |X(N_s,N_f)|^2')
% xlabel('Frequency (Hz)'); ylabel('Magnitude')
%
% figure
% imagesc(20*log(DenSpc_mtrx')) % espectrograma
% title('Spectrogram of the power spectrum window by window |X(N_s,N_f)|^2')
% ylabel('Number of samples per window'); xlabel('Number of windows')

%% 2nd Block - Filterbank

% Frequency features
low_freq  = 0;        % Low Frequency
high_freq = Fs/2;     % High Frequency

% Hertz to Mel-scale Limits
low_mel_freq =2595*log10(1+(low_freq/700));   % Low Frequency in Mel-Scale
high_mel_freq =2595*log10(1+(high_freq/700)); % High Frequency in Mel-Scale

% Creating the mel-scaled vector
Mel_vec = linspace(low_mel_freq,high_mel_freq,num_coef+2);

% Frequencies vector, computed from the Mel vector
freq_vec=700*((10.^(Mel_vec/2595))-1); % Separacion lineal de escala mel

% %Plotting
% figure
% plot(freq_vec, Mel_vec);
% title('Frequency to mel-frequency curve');
% xlabel('Frequency'); ylabel('Mel-frequency');

% Convert frequencies to nearest FFT bins
f= zeros(1,num_coef+2);
   for i = 1:num_coef+2
       f(i) = floor((512+1)*freq_vec(i)/Fs);
   end

   % Computes Filterbank
   Bank_mtrx = zeros(length(num_coef),N/2); % Filterbank Matrix
   for m = 2:1:num_coef+1
       for k = 1:N/2
           if k < f(m-1);
```

```
            Bank_mtrx(m-1:num_coef,k)=0;

        elseif k >= f(m-1) && k <= f(m)
               Bank_mtrx(m-1:num_coef,k) = (k - f(m-1))/(f(m) - f(m-1));

        elseif k >= f(m) && k <= f(m+1)
               Bank_mtrx(m-1:num_coef,k) = (f(m+1) - k)/(f(m+1) - f(m));

        else k > f(m+1);
             Bank_mtrx(m-1:num_coef,k)=0;
        end
    end
end

%   %Plotting
%   figure
%   freq = (Fs/2)*linspace(0,1,N/2);
%   plot(freq,Bank_mtrx)
%   title('Filterbank'); xlabel('Frequency (Hz)'); ylabel('Amplitude')
%
%   figure
%   freq = (Fs/2)*linspace(0,1,N/2+1);
%   imagesc(freq,1:num_coef,Bank_mtrx)
%   title('Energy distribution of each filter')
%   xlabel('Frequency (Hz)'); ylabel('Filter')

% Computes the energy of each filter
[Num_frms,Frm_width] = size(Frms_mtrx);
nrgy = zeros(num_coef,Num_frms);

% Multiply each one of the filters per each one of the windows
for d = 1:1:Num_frms   % Window loop
    for f = 1:1:num_coef   % Filter loop
        nrgy(f,d) = Bank_mtrx(f,:)*DenSpc_mtrx(d,:)';
    end
end

%   % Plotting
%   figure
%   plot(nrgy); % Energy per filter
%   title('Energy of each filter in each window')
%   xlabel('Filter'); ylabel('Energy')
%   figure
```

```
%    imagesc(nrgy) % espectrogram % Energia por ventana
%    title('Energy of each filter in each window');
%    xlabel('Number of window'); ylabel('Number of Filter')

   %% 3th Block - Logarith
   log_filt = log(nrgy); % Computes the logarithm of each filter

%   % Plotting
%    figure
%    imagesc(log_filt) % espectrogram % Energia por ventana
%    title('Logarithmical energy of each filter in each window');
%    xlabel('Number of window'); ylabel('Number of Filter')

   %% 4th Block - Discrete Cosine Transform (DCT)

   Coefs_MFCC = dct(log_filt); %  Computes the DCT of each filter

%   % Plotting
% figure
% mesh(Coefs_MFCC)
% title('MFCC');
% xlabel('Number of windows'); ylabel('Number of Coeficientes');
% ylabel('Energy of the filter')
end
```

Apéndice G: Códigos para el entrenamiento de los coeficientes LPC

Código para el entrenamiento de los coeficientes LPC de las señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Thesis: Comparación de Técnicas de Parametrización Espectral
%         para Reconocimiento de Voz en Idioma Español
% Automatic Speech Recognition System Training with LPC and fm = 8kHz

clc; clear all; close all;
tic
%% Speakers and Words Information
spkrs = {'Al';'An';'Ana';'Gus';'In';'JM';'Ka';'Lu';'Man';'Pa'};
words = {'Cero';'Uno';'Dos';'Tres';'Cuatro';'Cinco';'Seis';'Siete';...
         'Ocho';'Nueve';'Encender';'Apagar';'Subir';'Bajar';'Volumen';...
         'Llamar';'Colgar';'Enviar';'Mensaje';'Buscar'};

%% LPC training recordings
num_words = 20;  % # of words
num_spkrs = 10;  % # of speakers
num_it_ent = 7;  % # of training iterations (first 7 iterations)

for w = 1:num_words
    for s = 1:num_spkrs
        for r = 1:num_it_ent
%%%%%%%%%%%%%%% Data acquisition (Load Data) %%%%%%%%%%%%%%%
[voice,Fs,t] = load_audio_8(char(spkrs(s)),char(words(w)),r);

%%
%%%%%%%%%%%%%%% Pre-processing %%%%%%%%%%%%%%%

%%%%% Low-pass filter %%%%%
[voice_lpf_8] = LPF_8(voice,Fs,t);

%%%%% High-pass filter %%%%%
[voice_hpf_8] = HPF_8(voice_lpf_8,Fs,t);

%%%%% Voice activity detection (VAD) %%%%%
[word_8] = vad_8(voice_hpf_8,Fs);

%%%%% Pre-emphasis %%%%%
[enfatiz_8] = pre_enf_8(word_8,Fs);
```

```
%%%%% Segmentation and Windowing %%%%%
[Frms_mtrx] = seg_win_8(enfatiz_8);

%%
%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%

%%%%%% Linear Predicction Coding (LPC) %%%%%
p = 8;     % Prediction order (# of coefficients)
[Coefs_LPC] = Coeff_LPC_8(Frms_mtrx,p);

LPC_ent{(w-1)*num_spkrs*num_it_ent + (s-1)*num_it_ent + r} = Coefs_LPC;


        end
    end
%% Modelado
Q = 5;   % # of states
O = p+1; % # of observations (coefficients per window)
mix = 3; % # of gaussian mixtures (gaussian mixtures per state)
Train = LPC_ent((w-1)*num_spkrs*num_it_ent + 1 : (w)*num_spkrs*num_it_ent);
[LL, prior2, transmat2, mu2, Sigma2, mixmat2] = training(Train,Q,O,mix);
    eval(['save Modelos/LPC_8_8_2/' words{w} ...
            ' LL prior2 transmat2 mu2 Sigma2 mixmat2']);
end
toc
```

Código para el entrenamiento de los coeficientes LPC de las señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Thesis: Comparación de Técnicas de Parametrización Espectral
%         para Reconocimiento de Voz en Idioma Español
% Automatic Speech Recognition System Training with LPC and fm = 16kHz

clc; clear all; close all;
tic
%% Speakers and Words Information
spkrs = {'Al';'An';'Ana';'Gus';'In';'JM';'Ka';'Lu';'Man';'Pa'};
words = {'Cero';'Uno';'Dos';'Tres';'Cuatro';'Cinco';'Seis';'Siete';...
         'Ocho';'Nueve';'Encender';'Apagar';'Subir';'Bajar';'Volumen';...
         'Llamar';'Colgar';'Enviar';'Mensaje';'Buscar'};

%% LPC training recordings
```

```
num_words = 20;  % # of words
num_spkrs = 10;  % # of speakers
num_it_ent = 7;  % # of training iterations (first 7 iterations)

for w = 1:num_words
    for s = 1:num_spkrs
        for r = 1:num_it_ent
%%%%%%%%%%%%%%% Data acquisition (Load Data) %%%%%%%%%%%%%%%
[voice,Fs,t] = load_audio_16(char(spkrs(s)),char(words(w)),r);

%%
%%%%%%%%%%%%%%% Pre-processing %%%%%%%%%%%%%%%

%%%%% Low-pass filter %%%%%
[voice_lpf_16] = LPF_16(voice,Fs,t);

%%%%% High-pass filter %%%%%
[voice_hpf_16] = HPF_16(voice_lpf_16,Fs,t);

%%%%% Voice activity detection (VAD) %%%%%
[word_16] = vad_16(voice_hpf_16,Fs);

%%%%% Pre-emphasis %%%%%
[enfatiz_16] = pre_enf_16(word_16,Fs);

%%%%% Segmentation and Windowing %%%%%
[Frms_mtrx] = seg_win_16(enfatiz_16);

%%
%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%

%%%%%% Linear Predicction Coding (LPC) %%%%%
p = 16;     % Prediction order (# of coefficients)
[Coefs_LPC] = Coeff_LPC_16(Frms_mtrx,p);

LPC_ent{(w-1)*num_spkrs*num_it_ent + (s-1)*num_it_ent + r} = Coefs_LPC;


        end
    end
%% Modelado
Q = 5;   % # of states
O = p+1; % # of observations (coefficients per window)
```

```
mix = 3; % # of gaussian mixtures (gaussian mixtures per state)
Train = LPC_ent((w-1)*num_spkrs*num_it_ent + 1 : (w)*num_spkrs*num_it_ent);
[LL, prior2, transmat2, mu2, Sigma2, mixmat2] = training(Train,Q,O,mix);
    eval(['save Modelos/LPC_16_16_2/' words{w} ...
            ' LL prior2 transmat2 mu2 Sigma2 mixmat2']);
end
toc
```

Apéndice H: Códigos para el reconocimiento de los coeficientes LPC

Código para el reconocimiento de los coeficientes LPC de las señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Thesis: Comparación de Técnicas de Parametrización Espectral
%         para Reconocimiento de Voz en Idioma Español
% Automatic Speech Recognition System Test with LPC and fm = 8kHz

clc; clear all; close all;
tic
%% Speakers and Words Information
spkrs = {'Al';'An';'Ana';'Gus';'In';'JM';'Ka';'Lu';'Man';'Pa'};
words = {'Cero';'Uno';'Dos';'Tres';'Cuatro';'Cinco';'Seis';'Siete';...
         'Ocho';'Nueve';'Encender';'Apagar';'Subir';'Bajar';'Volumen';...
         'Llamar';'Colgar';'Enviar';'Mensaje';'Buscar'};

%% LPC test recordings
num_words = 20;  % # of words
num_spkrs = 10;  % # of speakers
num_it_prb = 3;      % # of test iterations (last 3 iterations)

for w = 1:num_words
    for s = 1:num_spkrs
        for r = 1:num_it_prb
%%%%%%%%%%%%%%%% Data acquisition (Load Data) %%%%%%%%%%%%%%%%
[voice,Fs,t] = load_audio_8(char(spkrs(s)),char(words(w)),7+r);

%%
%%%%%%%%%%%%%%%% Pre-processing %%%%%%%%%%%%%%%%

%%%%% Low-pass filter %%%%%
[voice_lpf_8] = LPF_8(voice,Fs,t);

%%%%% High-pass filter %%%%%
[voice_hpf_8] = HPF_8(voice_lpf_8,Fs,t);

%%%%% Voice activity detection (VAD) %%%%%
[word_8] = vad_8(voice_hpf_8,Fs);

%%%%% Pre-emphasis %%%%%
[enfatiz_8] = pre_enf_8(word_8,Fs);
```

```
%%%%% Segmentation and Windowing %%%%%
[Frms_mtrx] = seg_win_8(enfatiz_8);

%%
%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%

%%%%%% Linear Predicction Coding (LPC) %%%%%
p = 8;     % Prediction order (# of coefficients)
[Coefs_LPC] = Coeff_LPC_8(Frms_mtrx,p);

LPC_prb{(w-1)*num_spkrs*num_it_prb + (s-1)*num_it_prb + r} = Coefs_LPC;
        end
    end
%% Modelado

% Contadores
hit = 0; error = 0; ViterRes = zeros(1,20);
x = num_it_prb*num_spkrs; % Total number of possible hits

Test = LPC_prb((w-1)*num_spkrs*num_it_prb + 1 : (w)*num_spkrs*num_it_prb);

    for m = 1:num_spkrs*num_it_prb
        for n = 1:num_words
            eval(['load Modelos/LPC_8_8_2/' words{n}]);
            ViterRes(n) = mhmm_logprob(Test(m),prior2, transmat2,...
                mu2, Sigma2, mixmat2);
        end
        [tmp,Recog] = max(ViterRes);
        if Recog == w; hit = hit + 1; end;
    end
    rate = (hit / x)*100;
    fprintf('***--Speech recognition rate=%d --***\n',rate);
    hit = 0;
end
toc
```

Código para el reconocimiento de los coeficientes LPC de las señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Thesis: Comparación de Técnicas de Parametrización Espectral
%         para Reconocimiento de Voz en Idioma Español
% Automatic Speech Recognition System Test with LPC and fm = 16kHz
```

```
clc; clear all; close all;
tic
%% Speakers and Words Information
spkrs = {'Al';'An';'Ana';'Gus';'In';'JM';'Ka';'Lu';'Man';'Pa'};
words = {'Cero';'Uno';'Dos';'Tres';'Cuatro';'Cinco';'Seis';'Siete';...
         'Ocho';'Nueve';'Encender';'Apagar';'Subir';'Bajar';'Volumen';...
         'Llamar';'Colgar';'Enviar';'Mensaje';'Buscar'};

%% LPC test recordings
num_words = 20;  % # of words
num_spkrs = 10;  % # of speakers
num_it_prb = 3;      % # of test iterations (last 3 iterations)

for w = 1:num_words
    for s = 1:num_spkrs
        for r = 1:num_it_prb
%%%%%%%%%%%%%%% Data acquisition (Load Data) %%%%%%%%%%%%%%%
[voice,Fs,t] = load_audio_16(char(spkrs(s)),char(words(w)),7+r);

%%
%%%%%%%%%%%%%%% Pre-processing %%%%%%%%%%%%%%%

%%%%% Low-pass filter %%%%%
[voice_lpf_16] = LPF_16(voice,Fs,t);

%%%%% High-pass filter %%%%%
[voice_hpf_16] = HPF_16(voice_lpf_16,Fs,t);

%%%%% Voice activity detection (VAD) %%%%%
[word_16] = vad_16(voice_hpf_16,Fs);

%%%%% Pre-emphasis %%%%%
[enfatiz_16] = pre_enf_16(word_16,Fs);

%%%%% Segmentation and Windowing %%%%%
[Frms_mtrx] = seg_win_16(enfatiz_16);

%%
%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%

%%%%%% Linear Predicction Coding (LPC) %%%%%
p = 16;     % Prediction order (# of coefficients)
```

```
[Coefs_LPC] = Coeff_LPC_16(Frms_mtrx,p);

LPC_prb{(w-1)*num_spkrs*num_it_prb + (s-1)*num_it_prb + r} = Coefs_LPC;
        end
    end
%% Modelado

% Contadores
hit = 0; error = 0; ViterRes = zeros(1,20);
x = num_it_prb*num_spkrs; % Total number of possible hits

Test = LPC_prb((w-1)*num_spkrs*num_it_prb + 1 : (w)*num_spkrs*num_it_prb);

    for m = 1:num_spkrs*num_it_prb
        for n = 1:num_words
            eval(['load Modelos/LPC_16_16_2/' words{n}]);
            ViterRes(n) = mhmm_logprob(Test(m),prior2, transmat2,...
                mu2, Sigma2, mixmat2);
        end
        [tmp,Recog] = max(ViterRes);
        if Recog == w; hit = hit + 1; end;
    end
    rate = (hit / x)*100;
    fprintf('***--Speech recognition rate=%d --***\n',rate);
    hit = 0;
end
toc
```

Apéndice I: Códigos para el entrenamiento de los coeficientes MFCC

Código para el entrenamiento de los coeficientes MFCC de las señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Thesis: Comparación de Técnicas de Parametrización Espectral
%         para Reconocimiento de Voz en Idioma Español
% Automatic Speech Recognition System Training with MFCC and fm = 8kHz

clc; clear all; close all;
tic
%% Speakers and Words Information
spkrs = {'Al';'An';'Ana';'Gus';'In';'JM';'Ka';'Lu';'Man';'Pa'};
words = {'Cero';'Uno';'Dos';'Tres';'Cuatro';'Cinco';'Seis';'Siete';...
         'Ocho';'Nueve';'Encender';'Apagar';'Subir';'Bajar';'Volumen';...
         'Llamar';'Colgar';'Enviar';'Mensaje';'Buscar'};

%% LPC training recordings
num_words = 20;  % # of words
num_spkrs = 10;  % # of speakers
num_it_ent = 7;  % # of training iterations (first 7 iterations)

for w = 1:num_words
    for s = 1:num_spkrs
        for r = 1:num_it_ent
%%%%%%%%%%%%%%% Data acquisition (Load Data) %%%%%%%%%%%%%%%
[voice,Fs,t] = load_audio_8(char(spkrs(s)),char(words(w)),r);

%%
%%%%%%%%%%%%%%% Pre-processing %%%%%%%%%%%%%%%

%%%%% Low-pass filter %%%%%
[voice_lpf_8] = LPF_8(voice,Fs,t);

%%%%% High-pass filter %%%%%
[voice_hpf_8] = HPF_8(voice_lpf_8,Fs,t);

%%%%% Voice activity detection (VAD) %%%%%
[word_8] = vad_8(voice_hpf_8,Fs);

%%%%% Pre-emphasis %%%%%
[enfatiz_8] = pre_enf_8(word_8,Fs);
```

```
%%%%% Segmentation and Windowing %%%%%
[Frms_mtrx] = seg_win_8(enfatiz_8);

%%
%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%

%%%%%% Linear Predicction Coding (LPC) %%%%%
num_coef = 8;        % Prediction order (# of coefficients)
[Coefs_MFCC] = Coeff_MFCC_8(Fs,Frms_mtrx,num_coef);

MFCC_ent{(w-1)*num_spkrs*num_it_ent + (s-1)*num_it_ent + r} = Coefs_MFCC;

        end
    end
%% Modelado
Q = 5;           % # of states
O = num_coef;  % # of observations (coefficients per window)
mix = 3;         % # of gaussian mixtures (gaussian mixtures per state)
Train = MFCC_ent((w-1)*num_spkrs*num_it_ent + 1 : (w)*num_spkrs*num_it_ent);
[LL, prior2, transmat2, mu2, Sigma2, mixmat2] = training(Train,Q,O,mix);
    eval(['save Modelos/MFCC_8_8_2/' words{w} ...
            ' LL prior2 transmat2 mu2 Sigma2 mixmat2']);
end
toc
```

Código para el entrenamiento de los coeficientes MFCC de las señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Thesis: Comparación de Técnicas de Parametrización Espectral
%         para Reconocimiento de Voz en Idioma Español
% Automatic Speech Recognition System Training with MFCC and fm = 16kHz

clc; clear all; close all;
tic
%% Speakers and Words Information
spkrs = {'Al';'An';'Ana';'Gus';'In';'JM';'Ka';'Lu';'Man';'Pa'};
words = {'Cero';'Uno';'Dos';'Tres';'Cuatro';'Cinco';'Seis';'Siete';...
         'Ocho';'Nueve';'Encender';'Apagar';'Subir';'Bajar';'Volumen';...
         'Llamar';'Colgar';'Enviar';'Mensaje';'Buscar'};

%% LPC training recordings
num_words = 20;  % # of words
```

```
num_spkrs = 10;  % # of speakers
num_it_ent = 7;  % # of training iterations (first 7 iterations)

for w = 1:num_words
    for s = 1:num_spkrs
        for r = 1:num_it_ent
%%%%%%%%%%%%%%% Data acquisition (Load Data) %%%%%%%%%%%%%%%
[voice,Fs,t] = load_audio_16(char(spkrs(s)),char(words(w)),r);

%%
%%%%%%%%%%%%%%% Pre-processing %%%%%%%%%%%%%%%

%%%%% Low-pass filter %%%%%
[voice_lpf_16] = LPF_16(voice,Fs,t);

%%%%% High-pass filter %%%%%
[voice_hpf_16] = HPF_16(voice_lpf_16,Fs,t);

%%%%% Voice activity detection (VAD) %%%%%
[word_16] = vad_16(voice_hpf_16,Fs);

%%%%% Pre-emphasis %%%%%
[enfatiz_16] = pre_enf_16(word_16,Fs);

%%%%% Segmentation and Windowing %%%%%
[Frms_mtrx] = seg_win_16(enfatiz_16);

%%
%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%

%%%%%% Linear Predicction Coding (LPC) %%%%%
num_coef = 16;        % Prediction order (# of coefficients)
[Coefs_MFCC] = Coeff_MFCC_16(Fs,Frms_mtrx,num_coef);

MFCC_ent{(w-1)*num_spkrs*num_it_ent + (s-1)*num_it_ent + r} = Coefs_MFCC;

        end
    end
%% Modelado
Q = 5;           % # of states
O = num_coef;  % # of observations (coefficients per window)
mix = 3;         % # of gaussian mixtures (gaussian mixtures per state)
Train = MFCC_ent((w-1)*num_spkrs*num_it_ent + 1 : (w)*num_spkrs*num_it_ent);
```

```
[LL, prior2, transmat2, mu2, Sigma2, mixmat2] = training(Train,Q,O,mix);
    eval(['save Modelos/MFCC_16_16_2/' words{w} ...
            ' LL prior2 transmat2 mu2 Sigma2 mixmat2']);
end
toc
```

Apéndice J: Códigos para el reconocimiento de los coeficientes MFCC

Código para el reconocimiento de los coeficientes MFCC de las señales con $Fm = 8kHz$

```
% Manuel Alejandro Soto Murillo
% Thesis: Comparación de Técnicas de Parametrización Espectral
%         para Reconocimiento de Voz en Idioma Español
% Automatic Speech Recognition System Test with MFCC and fm = 8kHz

clc; clear all; close all;
tic
%% Speakers and Words Information
spkrs = {'Al';'An';'Ana';'Gus';'In';'JM';'Ka';'Lu';'Man';'Pa'};
words = {'Cero';'Uno';'Dos';'Tres';'Cuatro';'Cinco';'Seis';'Siete';...
         'Ocho';'Nueve';'Encender';'Apagar';'Subir';'Bajar';'Volumen';...
         'Llamar';'Colgar';'Enviar';'Mensaje';'Buscar'};

%% LPC test recordings
num_words = 20;  % # of words
num_spkrs = 10;  % # of speakers
num_it_prb = 3;  % # of test iterations (last 3 iterations)

for w = 1:num_words
    for s = 1:num_spkrs
        for r = 1:num_it_prb
%%%%%%%%%%%%%%% Data acquisition (Load Data) %%%%%%%%%%%%%%%
[voice,Fs,t] = load_audio_8(char(spkrs(s)),char(words(w)),7+r);

%%
%%%%%%%%%%%%%%% Pre-processing %%%%%%%%%%%%%%%

%%%%% Low-pass filter %%%%%
[voice_lpf_8] = LPF_8(voice,Fs,t);

%%%%% High-pass filter %%%%%
[voice_hpf_8] = HPF_8(voice_lpf_8,Fs,t);

%%%%% Voice activity detection (VAD) %%%%%
[word_8] = vad_8(voice_hpf_8,Fs);

%%%%% Pre-emphasis %%%%%
[enfatiz_8] = pre_enf_8(word_8,Fs);
```

```
%%%%% Segmentation and Windowing %%%%%
[Frms_mtrx] = seg_win_8(enfatiz_8);

%%
%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%

%%%%%% Linear Predicction Coding (LPC) %%%%%
num_coef = 8;        % Prediction order (# of coefficients)
[Coefs_MFCC] = Coeff_MFCC_8(Fs,Frms_mtrx,num_coef);

MFCC_prb{(w-1)*num_spkrs*num_it_prb + (s-1)*num_it_prb + r} = Coefs_MFCC;
        end
    end
%% Modelado

% Contadores
hit = 0; error = 0; ViterRes = zeros(1,20);
x = num_it_prb*num_spkrs; % Total number of possible hits

Test = MFCC_prb((w-1)*num_spkrs*num_it_prb + 1 : (w)*num_spkrs*num_it_prb);

    for m = 1:num_spkrs*num_it_prb
        for n = 1:num_words
            eval(['load Modelos/MFCC_8_8_2/' words{n}]);
            ViterRes(n) = mhmm_logprob(Test(m),prior2, transmat2,...
                mu2, Sigma2, mixmat2);
        end
        [tmp,Recog] = max(ViterRes);
        if Recog == w; hit = hit + 1; end;
    end
    rate = (hit / x)*100;
    fprintf('***--Speech recognition rate=%d --***\n',rate);
    hit = 0;
end
toc
```

Código para el reconocimiento de los coeficientes MFCC de las señales con $Fm = 16kHz$

```
% Manuel Alejandro Soto Murillo
% Thesis: Comparación de Técnicas de Parametrización Espectral
%         para Reconocimiento de Voz en Idioma Español
% Automatic Speech Recognition System Test with MFCC and fm = 16kHz
```

```
clc; clear all; close all;
tic
%% Speakers and Words Information
spkrs = {'Al';'An';'Ana';'Gus';'In';'JM';'Ka';'Lu';'Man';'Pa'};
words = {'Cero';'Uno';'Dos';'Tres';'Cuatro';'Cinco';'Seis';'Siete';...
         'Ocho';'Nueve';'Encender';'Apagar';'Subir';'Bajar';'Volumen';...
         'Llamar';'Colgar';'Enviar';'Mensaje';'Buscar'};

%% LPC test recordings
num_words = 20;  % # of words
num_spkrs = 10;  % # of speakers
num_it_prb = 3;  % # of test iterations (last 3 iterations)

for w = 1:num_words
    for s = 1:num_spkrs
        for r = 1:num_it_prb
%%%%%%%%%%%%%%% Data acquisition (Load Data) %%%%%%%%%%%%%%%
[voice,Fs,t] = load_audio_16(char(spkrs(s)),char(words(w)),7+r);

%%
%%%%%%%%%%%%%%% Pre-processing %%%%%%%%%%%%%%%

%%%%% Low-pass filter %%%%%
[voice_lpf_16] = LPF_16(voice,Fs,t);

%%%%% High-pass filter %%%%%
[voice_hpf_16] = HPF_16(voice_lpf_16,Fs,t);

%%%%% Voice activity detection (VAD) %%%%%
[word_16] = vad_16(voice_hpf_16,Fs);

%%%%% Pre-emphasis %%%%%
[enfatiz_16] = pre_enf_16(word_16,Fs);

%%%%% Segmentation and Windowing %%%%%
[Frms_mtrx] = seg_win_16(enfatiz_16);

%%
%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%

%%%%%% Linear Predicction Coding (LPC) %%%%%
num_coef = 16;        % Prediction order (# of coefficients)
```

```
[Coefs_MFCC] = Coeff_MFCC_16(Fs,Frms_mtrx,num_coef);

MFCC_prb{(w-1)*num_spkrs*num_it_prb + (s-1)*num_it_prb + r} = Coefs_MFCC;
        end
    end
%% Modelado

% Contadores
hit = 0; error = 0; ViterRes = zeros(1,20);
x = num_it_prb*num_spkrs; % Total number of possible hits

Test = MFCC_prb((w-1)*num_spkrs*num_it_prb + 1 : (w)*num_spkrs*num_it_prb);

    for m = 1:num_spkrs*num_it_prb
        for n = 1:num_words
            eval(['load Modelos/MFCC_16_16_2/' words{n}]);
            ViterRes(n) = mhmm_logprob(Test(m),prior2, transmat2,...
                mu2, Sigma2, mixmat2);
        end
        [tmp,Recog] = max(ViterRes);
        if Recog == w; hit = hit + 1; end;
    end
    rate = (hit / x)*100;
    fprintf('***--Speech recognition rate=%d --***\n',rate);
    hit = 0;
end
toc
```

Referencias

[1] Sadaoki Furui, *Digital Speech Processing, Synthesis, and Recognition*, Second edition, New York, Marcel Dekker, pp. 1-3, 2000.

[2] Amparo Varona Fernández, *Antecedentes y desarrollo de los sistemas actuales de reconocimiento automático del habla*, Universidad del País Vasco, Departamento de Electricidad y Electrónica, Bilbao, pp. 321-346, 1997.

[3] S. Adrian Poulton, *Microcomputer Speech Synthesis and Recognition*, First edition, London, Sigma Thecnical Press, 1983.

[4] John J. Ohala, *Christian Gottlieb Kratzenstein:Pioneer in Speech Synthesis*, ICPhS XVII, Hong Kong, pp 4, 2011.

[5] Bernd Pompino-Marschall, *Von Kempelen, Remarks on the history of articulatory-acoustic modelling*, ZAS Papers in Linguistics 40, pp 145-159, 2005.

[6] Sadaoki Furui, *50 years of progress in speech and speaker recognition*, Department of Computer Science, Tokyo Institute of Technology, pp 1-9, 2004.

[7] K. H. Davis, R. Biddulph, and S. Balashek, *Automatic recognition of spoken digits*, Journal of the Acoustical Society of America, Volume 24, Issue 6, pp 637-642, 1952.

[8] L.R. Rabiner, B.H. Juang *Fundamentals of speech recognition*, First edition, New Jersey, Prentice Hall, 1993.

[9] D. R. Fry *Theoretical aspects of mechanical speech recognition*, and P. Denes *The design and operation of the mechanical speech recognizer at University College London*, University College London, J. British Inst. Radio Engr., Volume 19, Issue 4, pp. 211-229, 1959.

[10] J. W. Forgie and C. D. Forgie *Results obtained from a vowel recognition computer program*, Journal of the Acoustical Society of America , Volume 31, Issue 11, pp. 1480-1489. 1959.

[11] J. Suzuki and K. Nakata *Recognition of Japanese vowels preliminary to the recognition of speech*, J. Radio Res. Lab, Volume 37, Issue 8, pp. 193-212, 1961.

[12] T. Sakai and S. Doshita *The phonetic typewriter, information processing*, Proc. IFIP Congress, pp. 445-450, 1962.

[13] K. Nagata, Y. Kato, and C. Chiba *Spoken digit recognizer for Japanese language*, NEC Res. Develop., pp. 6, 1963.

[14] T. B. Martin, *Speech recognition by feature abstraction techniques*, Technical Documentary Report No. AL TDR 64-176, Air Force Avionics Laboratory, 1964.

[15] T. K. Vintsyuk, *Speech discrimination by dynamic programming*, Kibernetika, Volume 4, Issue 2, pp. 81-88, 1968.

[16] H. Sakoe and S. Chiba, *Dynamic programming algorithm optimization for spoken word recognition*, IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-26 (1), pp. 43-49, 1978.

[17] F. Itakura, *Minimum prediction residual applied to speech recognition*, IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-23 (1), pp. 67-72, 1975.

[18] C. C. Tappert, et. al., *Automatic recognition of continuous speech utilizing dynamic segmentation, dual classification, sequential decoding and error recovery*, Rome Air Dev. Cen, Rome, NY, Tech. Report TR-71-146, 1971.

[19] L. R. Rabiner, et. al., *Speaker independent recognition of isolated words using clustering techniques*, IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-27, pp. 336-349, 1979.

[20] D. Klatt, *Review of the ARPA speech understanding project*, J.A.S.A., Volume 62, Issue 6, pp. 1324-1366, 1977.

[21] B. Lowerre, *The HARPY speech understanding system*, Trends in Speech Recognition, W. Lea, Ed., Speech Science Pub., pp. 576-586, 1990.

[22] F. Jelinek, *Continuous Speech Recognition by Statistical Methods*, Proc. of the IEEE, 64(4), 532-566, 1976.

[23] L.R. Bahl, F. Jelink, R.L. Mercer, *Maximum Likelihood Approach to Continuous Speech Recognition*, IEEE Transactions on Pattern Analysis and Machine Inteligence, Volume 5, Issue 2, 179-190, 1983.

[24] S. Furui, *Speaker independent isolated word recognition using dynamic features of speech spectrum*, IEEE Trans. Acoustics, Speech, Signal Processing, ASSP-34, pp. 52-59, 1986.

[25] F. Jelinek *The development of an experimental discrete dictation recognizer*, Proc. IEEE, Volume 73, Issue 11, pp. 1616-1624, 1985.

[26] R. P. Lippmann *An introduction to computing with neural nets*, IEEE ASSP Mag., Volume 4, Issue 2, pp. 4-22, 1987.

[27] A. Weibel, et. al., *Phoneme recognition using time-delay neural networks*, IEEE Trans. Acoustics, Speech, Signal Proc., Volume 37, pp. 393-404, 1989.

[28] B.-H. Juang and S. Furui, *Automatic recognition and understanding: A first step toward natural humanmachine communication*, Proc. IEEE, 88, 8, pp. 1142- 1165, 2000.

[29] C. J. Leggetter and P. C. Woodland, *Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models*, Computer Speech and Language, 9, pp. 171-185, 1995.

[30] A. P. Varga and R. K. Moore, *Hidden Markov model decomposition of speech and noise*, Proc. ICASSP, pp. 845-848, 1990.

[31] M. J. F. Gales and S. J. Young, *Robust Continuous Speech Recognition Using Parallel Model Combination*, IEEE Transactions on Speech and Audio Processing, Volume 4, Issue 5, 1996.

[32] H. Soltau, et. al., *The IBM 2004 conversational telephone system for rich transcription*, Proc. ICASSP, pp. 205-208, 2005.

[33] S. Furui, *Speech-to-Text and Speech-to-Speech Summarization of Spontaneous Speech*, IEEE Trans. Speech & Audio Processing, 12, 4, pp. 401-408, 2004.

[34] S. Furui, *Recent progress in corpus-based spontaneous speech recognition*, IEICE Trans. Inf. & Syst., E88-D, 3, pp. 366-375, 2005.

[35] T. Kawahara, et al., *Key-phrase detection and verification for flexible speech understanding*, IEEE Trans. Speech and Audio Proc., Volume 6, Issue 6, pp. 558-568, 1998.

[36] C. Lévy, G. Linares, P. Nocera, *Comparaison of Several Acoustic Modeling Techniques and Decoding Algorithms for Embedded Speech Recognition Systems*, Université dÁvignon et des Pays du Vaucluse, 2008.

[37] R.G.Leonard, *A database for speaker-independent digit recognition*, proceedings of ICASSP, San Diego, vol. 3, pp.42.11, 1984.

[38] R. Carré, R. Descout, M. Eskénazi, J. Mariani, M. Rossi, *The French language database: defining, planning, and recording a large database*, proceedings of ICASSP, San Diego, 1984.

[39] J. Méndez, J. Hernando, C. Nadeu, F. Vallveerdú, *Esquema Unificado de Parametrización de la Señal de Voz en Reconocimiento del Habla*, Simposium Nacional de la Unión Científica Internacional de Radio, Universidad de Valladolid, pp. 97-100, 1995.

[40] H. Hermansky, *Perceptual linear predictive (PLP) analysis of speech*, J. Acoust. Soc. Amer., vol. 87, pp. 1738-1752, 1990.

[41] M. Chetouani, M. Faundez-Zanuy, B. Gas, J.L. Zarader, *A New Nonlinear speaker parameterization algorithm for speaker identification*, ODYSSEY04 - The Speaker and Language Recognition Workshop, Toledo, Spain, 2004.

[42] J. Ortega-Garcia and al., *Ahumada: a large speech corpus in spanish for speaker identification and verification*, ICASSP, vol. 2, pp. 773-776, 1998.

[43] R. Modic, B. Lindberg, B. Petek *Comparative Wavelet and MFCC Speech Recognition Experiments on the Slovenian and English SpeechDat2*,ITRW on Non-Linear Speech Processing (NOLISP 03) Le Croisic, France, 2003.

[44] R. Sarikaya, B. L. Pellom, y J. H. Hansen, *Wavelet Packet Transform Features with Application to Speaker Identification*, NORSIG, pp.81-84, 1998.

[45] C. Duque y M. Morales, *Caracterización de voz empleando análisis tiempo-frecuencia aplicada al reconocimiento de emociones*, Tesis de grado para obtener el título de Ingeniero Electricista, Universidad Tecnológica de Pereira, Abril 2007

[46] C. Caballero, *Cómo educar la voz hablada y cantada*, 11va. edición, México, Editores Asociados Mexicanos (EDAMEX), pp. 257, 1998.

[47] R. Resnick, D. Halliday, *Física para estudiantes de Ciencias e Ingeniería. Parte 1*, 3era edición, México, John Wiley & Sons, Inc, 1992.

[48] M. N. Levy, B. M. Koeppen and B. A. Stanton, *Berne y Levy Fisiología*, Elsevier, pp.159-160, 2006.

[49] A. C. Guyton y J. E. Hall, *Tratado de fisiología médica*, Elsevier, pp.159-160, 2006. Elsevier, pp.662-698, 2006.

[50] A. Ardila, B. Bernal and M. Rosselli, *How Localized are Language Brain Areas? A Review of Brodmann Areas Involvement in Oral Language*, Archives of Clinical Neuropsychology, Volume 31, Issue 1, pp.112-122, 2015.

[51] A. Flores, *Reconocimiento de Palabras Aisladas en Castellano*, Inictel, Dirección de Investigación y Desarrollo, 1993.

[52] Health, U.S Department of Health & Human Services, National Institute on Deafness and Other Communication Disorders (NIDCD), *Taking Care of Your Voice*, Noviembre 2016, disponible en:
https://www.nidcd.nih.gov/health/taking-care-your-voice.

[53] Health, MED-EL, *Cómo Funciona la Audición*, Noviembre 2016, disponible en:
http://www.medel.com/esl/how-hearing-works/

[54] R. L. Klevans and R. D. Rodman, *Voice Recognition*, ARTECH HOUSE, INC., Norwood, USA, pp 1-9, 1997.

[55] R. Stuckless, *Developments in real-time speech-to-text communication for people with impaired hearing*, In M. Ross, Baltimore, MD, York Press, Communication access for people with hearing loss, pp.197-226, 1994.

[56] R.E. Gruhn et al., *Statistical Pronunciation Modeling for Non-Native Speech Processing*, Signals and Communication Technology, Springer-Verlag, Berlin Heidelberg, 2011.

[57] Speech, Music and Hearing, Kunglinga Tekniska Högskolan, *Snack v2.2.8 manual*, Diciembre 2016, disponible en:
http://www.speech.kth.se/snack/man/snack2.2/SoundProp.html

[58] Sundarrajan Rangachari, *Noise estimation algorithms for highly non-stationary environments*, University of Texas at Dallas, 2004.

[59] J. Xu, A. Ariyaeeinia, R. Sotudeh and Z. Ahmad, *Pre-processing Speech Signal in FPGAs*, University of Herrtsfordshire, UK, IEEE Xplore, 2005.

[60] J. Bernal, J. Bobadilla, P. Gómez Vilda, *Reconocimiento de voz y fonética acústica*, ALFAOMEGA GRUPO EDITOR, 2000.

[61] B. S. Atal, *The history of linear prediction*, IEEE Signal Process Mag., vol. 23, no. 2, pp. 154-161, Mar. 2006.

[62] L. Wang Z. Chen and F. Yin, *A Novel Hierarchical Decomposition Vector Quantization Method for High-Order LPC Parameters*, IEEE/ACM Transactions on Audio, Speech, and Language Processing, VOL. 23, No. 1, 2015.

[63] D. Reig, *Implementación de algoritmos para la extracción de patrones característicos en Sistemas de Reconocimiento De Voz en Matlab*, Universidad Politécnica de Valencia, Gandia, 2014.

[64] N. Wankhede and M. Shah, *Investigation on Optimum Parameters for LPC Based Vocal Tract Shape Estimation*, International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA), Bangalore, India, Oct. 10-11, 2013.

[65] S. Feraru and M. Zbancioc, *Emotion Recognition in Romanian Language using LPC Features*, The 4th IEEE International Conference on E-Health and Bioengineering - EHB, Grigore T. Popa University of Medicine and Pharmacy, Iasi, Romania, November 21-23, 2013

[66] F. Wang and W. Xu, *A Comparison of Algorithms for the Calculation of LPC Coefficients*, Communication University of China, Beijing, Information Science, Electronics and Electrical Engineering (ISEEE), International Conference on vol.3, pp. 26-28, 2014.

[67] S. Davis and P. Mermelstein, *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28 No. 4, pp. 357-366, 1980.

[68] S. Stevens, J. Volkmann and E. Newman, *A Scale for the Measurement of the Psychological Magnitude Pitch*, The Journal of the Acoustical Society of America, Vol. 8 No. 3, pp. 185-190, 1937.

[69] A. Majeed, H. Husain, S. Samad y T. Idbeaa, *Mel Frequency Cepstral Coefficients (MFCC) Feature Extraction Enhacement in the Application of Speech Recognition a Comparison Study*, Journal of Theoretical and Applied Information Technology, Vol.79. No.1, pp 38-56, 2015.

[70] G. Martínez y G. Aguilar, *Reconocimiento de voz basado en MFCC, SBC y Espectrogramas*, Ingenius. N.10, ISSN: 1390-650X, pp. 12-20, 2013.

[71] Mathematics and Computer Science, Practical Cryptography *Mel Frequency Cepstral Coefficient (MFCC) tutorial*, Enero 2017, disponible en: http://practicalcryptography.com/miscellaneous/machinelearning/ guidemelfrequency-cepstralcoefficientsmfccs/

[72] D. Albiñana *Implementación de algoritmos para la extracción de patrones característicos en Sistemas de Reconocimiento de Voz en Matlab*, Universidad Politécnica de Valencia, 2014.

[73] M. Bezoui, A. Elmoutaouakkil and A.Beni-hssane *Feature Extraction of some Quranic Recitation using Mel-Frequency Cepstral Coeficients (MFCC)*, 5th International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco, Sept. 29 - Oct.1, 2016.

[74] Mm Boussaa, Im Atouf, Mm Atibi and A. Bennis *Comparison of MFCC and DWT features extractors applied to PCG classification*, 11th International Conference on Intelligent Systems: Theories and Applications (SITA), Mohammedia, Morocco, Oct. 19-20, 2016.

[75] 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.

[76] K. Murphey, *HMM toolkit*, Massachusetts EE.UU., Noviembre 2017, disponible en: https://www.cs.ubc.ca/ murphyk/Software/HMM.zip