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

APLICACIÓN DE MODELOS OCULTOS DE MARKOV EN PRUEBAS DE DICCIÓN DE PALABRAS AISLADAS DEL IDIOMA ESPAÑOL

Tesis de Maestría

presentada como requisito parcial para obtener el grado de

MAESTRO EN CIENCIAS DE LA INGENIERÍA

ORIENTACIÓN: PROCESAMIENTO DE SEÑALES Y MECATRÓNICA

Presentada por:

Ángel David Pedroza Ramírez

Directores de tesis:

Dr. José Ismael de la Rosa Vargas y Dr. Efrén González Ramírez

UNIDAD ACADÉMICA DE INGENIERÍA ELÉCTRICA

Zacatecas, Zac., Agosto de 2015

RESUMEN

Gracias a la comunicación oral el ser humano ha logrado, entre otras cosas, la transmisión de ideas y conocimientos. Sin embargo, la buena comunicación, independientemente del idioma, debe ser clara, objetiva y expresiva con el fin de que lo que se quiere expresar sea lo que el oyente entienda.

Hace algunas décadas se pensaba que educar la voz era asunto de los comentaristas, locutores, reporteros, entre otros. Hoy en día, sin embargo, no se puede dejar este asunto para aquellas áreas antes mencionadas sino que cualquier persona necesita hablar un lenguaje correcto, claro y natural.

El reconocimiento de voz se basa en el estudio sobre el proceso del habla y la comunicación, y la forma en que este conocimiento puede ser aplicado como herramienta para diversas finalidades.

El enfoque de esta investigación es la aplicación de Modelos Ocultos de Markov con la finalidad de realizar pruebas de dicción en el idioma español de palabras aisladas. Para ello se propone el uso de tres diferentes métodos que son modificaciones en la aplicación de Modelos Ocultos de Markov los cuales son **Modelos Ocultos de Markov con DTW (MDTW)**, **Modelos Ocultos de Markov con DTW aproximado por izquierda y derecha (MID)** y **Modelos Ocultos de Markov con relleno de palabras (MRP)**.

Mediante esta investigación para reconocimiento del habla se hace posible el distinguir entre calidades de dicción y con una eficiencia de reconocimiento por encima del 90 $\%$ para cualquiera de los métodos propuestos.

ABSTRACT

By oral communication , human beings have achieved, among other things, the transmission of ideas and knowledge. However, good communication regardless of the language, should be clear, objective and expressive so that what is meant is what the listener understands.

A few decades ago it was thought that training the voice was matter of commentators, announcers, reporters, and others. Today, however, we can't leave this matter to those areas; instead, anyone needs to talk a correct, clear and natural language.

Speech recognition is based on the research of the process of speech and communication, and how this knowledge can be applied as a tool for different purposes.

The focus of this research is the use of Hidden Markov Models in order to perform tests of diction in the Spanish language on isolated words. To do so, we propose three different methods which are modifications in the application of Hidden Markov Models which are **Hidden Markov Models with DTW (MDTW)**, **Hidden Markov Models with DTW approximated by left and right (MID)** and **Hidden Markov Models filling words (MRP)**.

Through this research for speech recognition, becomes possible to distinguish between qualities of diction and a recognition efficiency above 90 $\%$ for any of the proposed methods.

A Dios principalmente, por acompañarme y guiarme en cada momento de mi vida.

A mi papá, por creer y apoyarme en cada uno de mis sueños.

A mi mamá, por ser mi mejor amiga y un ejemplo en todo aspecto de mi vida.

A mis hermanas, por ser un verdadero regalo del cielo.

A mis abuelos Raquel, Antonio, Luis y Celia, por haber sido una fuente de sabiduría inagotable en mi vida.

A mis compañeros de cubículo Kikin, Gustavo, Luis, Jorge y Carlitos, por esos torneos de realidad aumentada que me hicieron tan feliz.

A mis compañeros de la Maestría en Ciencias, por inspirarme a dar lo mejor de mi.

A mis maestros, por inculcarme el valor del esfuerzo y la dedicación.

A mis mejores amigos Mariano y Hanniel, por formar parte importante de mi vida.

A mis hermanos de Jornadas de la escuela Tiberiades, Grupo JSES y Pandillas, por enseñarme tantas cosas.

A una persona muy importante en mi vida, Andrea Natali.

A todos mis compañeros y amigos de la ciudad de Tijuana, por su gran hospitalidad y apoyo recibido en este trabajo de investigación.

A todas aquellas personas que ofrecen su vida al servicio de los demás y a pesar del cansancio, nunca dejan de sonreír.

Agradecimientos

Durante estos años son muchas las personas e instituciones que han participado en este trabajo y a quienes quiero expresar mi gratitud por el apoyo y la confianza que me han prestado de forma desinteresada.

En primer lugar quiero agradecer a la Maestría en Ciencias de la Ingeniería de la Universidad Autónoma de Zacatecas por el apoyo recibido durante el periodo en el que he desarrollado mi labor investigadora.

Agradezco a mi director de tesis: Dr. José Ismael de la Rosa Vargas por el apoyo recibido en el desarrollo de esta investigación.

Debo un especial reconocimiento a **CONACYT** por su apoyo al concederme la beca con clave:**297033**, la cual me han facilitado el sustento y movilidad necesario para desarrollar mi labor investigadora.

También me complace agradecer la acogida y el apoyo recibido en el Centro de Investigación en Tecnología Digital (CITEDI) de manera especial al Dr. Victor H. Diaz Ramirez por su valiosa colaboración en este trabajo de investigación.

Agradezco la valiosa colaboración de Angelica Rubí Martinez Martinez, Daniella Barajas Pérez, Ángel Francisco Velazquez y Ana Gabriela Avila Acosta, quienes conforman el equipo de locutores que accedieron a cooperar en los experimentos.

Se reconoce la participación del Ing. Ernesto García D. por su valiosa colaboración.

Gracias.

Contenido General

Pag.

Lista de figuras

Figura Pag.

Figura Pag.

Lista de tablas

Capítulo 1

Introducción.

Durante millones de años los seres vivos han utilizado la comunicación para diversas necesidades y de dicha forma transmitir información para lograr interactuar con el exterior.

El ser humano, desde sus inicios, ha tenido la necesidad de transmitir ideas, sentimientos y pensamiento dada la necesidad de vivir en sociedad. En un inicio la comunicación se hacía solo por medio de sonidos o de forma pictográfica, sin embargo, no fue hasta que se logró tener una comunicación con sonidos articulados que tomó la importancia que tiene hoy en día.

La información que se transmite a través de la voz posee características que pueden ser analizadas por medio de sistemas de reconocimiento de voz. Dicha rama de la ciencia se subdivide en reconocimiento del habla (RAH) y reconocimiento del locutor (RAL).

En un inicio, se hablaba sobre la necesidad de comercializar las tecnologías de reconocimiento de voz [1] y que ellas pudiesen ser aplicadas para poder facilitar la vida diaria. Hoy en día esto es ya una realidad y entre las aplicaciones donde se le puede encontrar están las siguientes:

- Telefonía,
- Seguridad,
- Manos libres,
- Computación,
- Ayuda para discapacidades físicas y Rehabilitación,

- Dictado o traducción.

En un inicio los sistemas de reconocimiento de voz se enfocaron en la producción sintética de voz, como fue el caso de la máquina creada por Von Kempelen [2], el cual, mediante un arreglo mecánico, producía sonidos parecidos a los producidos por el ser humano, sin embargo este sistema requería de horas de entrenamiento. Este novedoso pero burdo invento daría más tarde lugar a los sintetizadores musicales y a los órganos que utilizaban tarjetas perforadas para producir ciertas melodías.

Poco a poco los sistemas fueron haciéndose más sofisticados hasta que años más tarde Thaddeus Cahill descubrió que cualquier sonido podía ser sintetizado como una sumatoria de pequeñas ondas sinusoidales (principio de superposición) el cual revolucionó el concepto sobre la síntesis de música. Diversos inventos fueron creados, como el caso del *telharmonium* el cual utilizaba este principio de adición de ondas.

Sin embargo, el primer invento enfocado al reconocimiento de voz fue un juguete llamado Radio Rex [3] manufacturado en 1926. Este funcionaba mediante un mecanismo de activación por comando de voz. Al ser nombrada la palabra *REX*, el mecanismo hacía que la energía contenida en la palabra disparara el dispositivo y liberara al perro. Sin embargo este dispositivo comercial tenía la desventaja que cualquier palabra con la misma energía que la palabra "Rex", hacía que el sistema funcionara, es decir, liberara al perro.

No fue sino hasta 1950 cuando los laboratorios Bell crearon el primer reconocedor más sofisticado hasta esa fecha; logrando reconocer dígitos de un solo hablante. Sin embargo aún era un poco burdo para las necesidades tecnológicas.

El ámbito del reconocimiento de voz siguió desarrollándose y en 1960 aparecen tres técnicas que fueron la base para el reconocimiento de voz [4]:

1. Transformada Rápida de Fourier (FFT): Forma eficiente de la transformada de Fourier, interpretada como un banco de filtros.

2. Análisis Cepstral: Utilizado como una aproximación espectral.

3. Codificación Lineal Predictiva (LPC): Modelos autoregresivos para representar la generación de voz.

Adicionalmente se crearon nuevos métodos de aproximaciones:

- Alineamiento temporal dinámico (DTW por sus siglas en inglés): Esquema de optimización, utilizado como método de normalización (puede ser utilizado para corregir variaciones por diferentes pronunciaciones de la misma palabra con diferente duración).
- Modelos ocultos de Markov: Modela una secuencia observada como la generada por una secuencia desconocida de variables.

Específicamente, los Modelos Ocultos de Markov (MOM o HMM por sus siglas en inglés) fueron desarrollados por primera vez por un grupo de IBM en 1970 los cuales, en primera instancia, se utilizaron para el reconocimiento continuo del habla [29].

Para 1976 se incorpora información semántica y de sintaxis con el fin de lograr más allá del reconocimiento del habla, la comprensión del lenguaje. Para este fin, se incorporaron y desarrollaron investigaciones con redes neuronales, DTW y HMM.

Los Modelos Ocultos de Markov tuvieron gran impacto hasta mediados de 1980, sin embargo, hoy en dia algunos sistemas comerciales aún los utilizan [5].

A finales de 1980 se utilizó información acústico-fonética para desarrollar reglas de clasificación para sonidos del habla. Para ese mismo año se utilizaron redes neuronales para delimitar secciones de frontera (dentro de la señal de voz, ¿qué parte de la señal posee información y cuál no?). Luego de una serie de avances en materia de bases de entrenamiento y subsistemas de extracción de información de la señal de voz, en 1998 algunos sistemas podían reconocer cerca de 60,000 palabras de vocabulario en tiempo real con menos del 10% de error.

Hoy en día, y luego de un proceso que como se explicó con anterioridad comenzó con invenciones burdas hasta llegar a los complejos sistemas híbridos, el reconocimiento de voz es un área de investigación multidisciplinaria que tiene muchísimas aplicaciones aún por seguir explorando.

Entre los principales sistemas existentes (ya sea en su fase experimental o comercial) se tienen [15]:

- LINCOLN: Aportación en cuanto al modelado de la voz rápida, con emisión y tensión.

- PHILIPS: Sistema pionero en reconocimiento rápido de habla continua.
- CSELT: Sistema de búsqueda rápida basado en un descifrado fonético.
- SPHINX-II: Sistema pionero independiente del locutor enfocado en grandes vocabularios.
- BYBLOS:Sistema independiente del locutor con algoritmo de búsqueda rápida.

La comunicación oral, por otro lado, es una de las herramientas de expresión y comunicación más importantes que en el pasado dio paso hacia la evolución y que en el mundo actual, forma parte de una necesidad básica. Es distinguible de la comunicación sonora de los animales por poseer dos elementos principales: ser simbólica y convencional.

El proceso de comunicación oral (entre dos personas) está formado por tres elementos principales: Locutor, medio y oyente. El locutor, mediante una serie de procesos de coordinación de órganos, articula las palabras (previamente procesadas por el cerebro) para formar un mensaje. Dicho mensaje se transmite por un medio (generalmente aire) para llegar al oyente (receptor) quien lo decodifica y cierra el ciclo o cadena del habla.

La buena comunicación debe ser clara, objetiva y expresiva ([6]) con el fin de que lo que se quiere expresar sea lo que el escucha (o receptor) entienda. El reto de la correcta articulación de palabras, y por tanto de una correcta dicción, está presente en casi cualquier momento de la vida diaria. Hace algunas décadas se pensaba que educar la voz era asunto de los comentaristas, locutores, reporteros, entre otros. Hoy en día, sin embargo, no se puede dejar este asunto para aquellas áreas antes mencionadas sino que cualquier persona necesita hablar un lenguaje correcto, claro y natural. Es por ello que, como se verá mas adelante en esta misma sección, la realización de pruebas para entrenar la dicción de palabras conforman el objetivo central de esta investigación.

Por otro lado, el conjunto de fonemas (cada uno de los sonidos simples del lenguaje hablado) en combinaciones diferentes forman las palabras propias de un idioma y la base de la comunicación oral. De tal forma que cada idioma posee un conjunto propio de fonemas siendo, por ejemplo, para el español 24, para el inglés cerca de 42 y para el francés casi 50 fonemas.

El sonido o conjunto de sonidos que se articulan de una forma clara y limpia para expresar cierta idea constituyen a la correcta dicción. Una buena dicción da por resultado un tono de voz natural y sin modificaciones propias del origen del hablante o la región donde se hable. De tal forma que, sea o no nativo en algún idioma, el hablante debe de tener una pronunciación estándar mostrando así dominio y naturalidad en el idioma.

Según la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO), se distingue entre 8 principales lenguas habladas en la mitad de la población a nivel mundial (en millones de habitantes)[7]:

1. chino: 1,200,
2. inglés: 478,
3. hindi: 437,
4. español: 392,
5. ruso: 284,
6. árabe: 225,
7. portugués: 184,
8. francés: 125.

En específico el idioma español es un idioma hablado por cerca de 392 millones de personas por lo que una aplicación para realizar pruebas de dicción en el idioma lograría pertinencia para las personas que intentan familiarizarse o desean hablarlo de mejor manera.

Existen diversos métodos para lograr el reconocimiento de voz, sin embargo la señal de voz es una señal de tipo aleatorio por lo que, la utilización de un modelo estadístico resulta más que pertinente. Por dicha razón, los Modelos Ocultos de Markov (HMM) son ideales para este tipo de aplicaciones en reconocimiento de voz.

Basado en lo anterior, como hipótesis se plantea que es posible la aplicación en software para reconocimiento del habla, con la finalidad de realizar pruebas de dicción en un idioma y

con una eficiencia de reconocimiento por encima del 90%; esto significa que la palabra emitida se reconocerá de manera confiable; todo ello con la finalidad de retroalimentar al hablante para probar su dicción.

El Objetivo General es por tanto, desarrollar una aplicación en software para reconocimiento del habla con el fin de apoyar la mejora de dicción de palabras aisladas en el idioma español. Para alcanzar este objetivo es necesario:

- Analizar esquemas recientes para el desarrollo de sistemas eficientes de reconocimiento de voz.
- Utilizar Modelos Ocultos de Markov y adaptarlos para la realización de pruebas de dicción.
- Completar las etapas de entrenamiento y prueba del sistema.

La estructura de la tesis es como sigue: en el capitulo 2 se plantean las bases sobre el proceso de producción de voz. El capitulo 3 trata sobre los fundamentos y principales técnicas para llevar a cabo del reconocimiento de voz .El capitulo 4 se enfoca en los sistemas de reconocimiento propuestos y, mediante el capítulo 5, se describe a detalle el proceso llevado a cabo en cada fase descrita. El capítulo 6 explica las pruebas y resultados obtenidos de la investigación realizada. Finalmente en el capitulo 7 se brindan las conclusiones.

Capítulo 2

Teoría fundamental sobre la producción de voz.

El reconocimiento de voz es una rama de la investigación multidisciplinaria y que ha logrado dar solución a diversas problemáticas del mundo real que van desde la ayuda a personas con deficiencias hasta los complejos sistemas de navegación con tecnología GPS. Específicamente el reconocimiento de voz es de nuestro especial interés dado que mediante este podemos lograr, una vez extraída la información por algún método, reconocer palabras con algún margen de error para después utilizarlo en algún proceso (como es el caso de la realización de pruebas de dicción). Con el fin de lograr entender mejor este proceso, en este capítulo se plantean las bases sobre la producción de voz.

La comunicación oral es entendida como la transferencia de información de una persona a otra por medio de la señal acústica llamada voz. La señal de voz consiste en variaciones de presión generadas por el tracto vocal del locutor. Dichas variaciones se propagan como ondas a través del aire y alcanzan el oído de los escuchas quienes las procesan y convierten en un mensaje tal como se muestra en la figura 2.1.

2.1 El sonido.

El sonido [6] es una vibración (ó movimiento ondulatorio) que se transmite en el aire (generalmente), hasta alcanzar el oído de quien lo percibe. Dicha vibración tiene cierta amplitud,

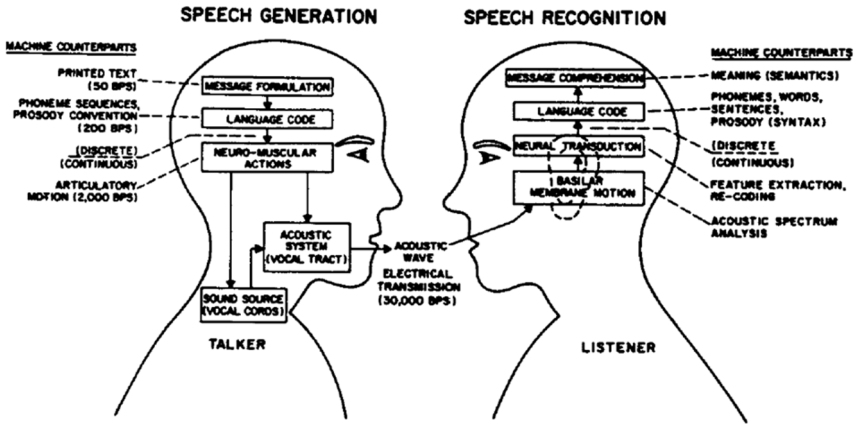


Figura 2.1 Comunicación oral [8].

frecuencia, duración y forma. Las características de la vibración caracterizan al sonido producido.

Siendo así, la frecuencia corresponde al tono del sonido (también llamado altura). Las frecuencias que son altas son denominadas "frecuencias en tono agudo" y las frecuencias bajas se denominan "frecuencias en tono grave". Es interesante mencionar que la cualidad de percepción la proporciona el tono mientras que el dolor auditivo experimentado en la audición lo da la energía y no, por tanto, el tono que se produce.

Al respecto, el sonido se puede clasificar de la siguiente manera:

- Infrasonidos: Frecuencias menores a 20 Hz.
- Percepción humana: Entre 20 y 20,000 Hz.
- Ultrasonidos: Frecuencias mayores a 20,000 Hz.

La amplitud se corresponde directamente a la intensidad del sonido (volumen) clasificado como "alto" o "bajo". Esta característica se refiere a la distancia entre el extremo superior e inferior de la onda de sonido. Dicho parámetro está relacionado con la intensidad y por tanto con la energía.

La duración de la vibración es también la duración del sonido. Por otro lado, las vibraciones parásitas permiten identificar a la fuente de sonido (a cada una corresponden diversos

armónicos) dado que son producidas por diversas formas de vibración. Esa cualidad es llamada en la teoría del sonido como timbre. Así, el sonido de una flauta y un violín con la misma intensidad, tono y duración se diferencian por el timbre. Lo anterior se puede resumir como se muestra en la figura 2.2.

Características de la **vibración.**		Características del **sonido**
Frecuencia	produce	Tono
Amplitud	"	Intensidad
Duración	"	Duración
Forma	"	Timbre

Figura 2.2 Correspondencia vibración a sonido [6].

Las vibraciones que puede captar el ser humano como sonido están limitadas por la frecuencia. Inferiores a 30 ciclos por segundo se percibe como movimiento mientras que, si sobrepasan los 18,000 ciclos por segundo rebasa las posibilidades de audición.

Si la amplitud es muy leve no se produce sensación auditiva pero si es muy grande, puede dañar de forma permanente el órgano de la audición. En cuanto a la duración, si es menor a 100/10000 de segundo solo se percibe como un "click" no identificable. Si la forma de vibración y frecuencia permanecen regulares se percibe como sonido; si alguno de los dos es irregular, el resultado es un ruido.

2.2 Órganos de Fonación.

La voz humana es conocida como el flujo de ondas (de tipo sonoras ó silencios) que se propaga por medio del aire (presión de moléculas). El proceso de producción de voz conlleva la combinación de órganos, huesos, músculos y algunos sistemas funcionales. La combinación de estos elementos en conjunto con la postura, la respiración y el estado emocional influyen en dicho proceso de producción [24]. Este proceso se resume de la siguiente forma: Previa inhalación de aire en los pulmones, la voz se produce cuando el aire es exhalado; al expandirse el diafragma, los órganos de respiración proporcionan un flujo de aire a los órganos de la

fonación, y es ahí donde el sonido adquiere sus características primarias en donde las cuerdas vocales juegan un rol importante. Luego, se agregan otras características impuestas por los órganos articulatorios, brindando al final una señal acústico-fonética.

El ser humano posee un sistema productor de sonido con ciertas características anatómicas (ver figura 2.3). En el sistema fonador interviene no solo la laringe, sino también los pulmones para proporcionar el aire y los modificadores de sonido (resonadores faringo-buco-nasales). Es por ello que, la calidad de sonidos producidos están directamente determinados por la posición de la lengua, los labios y la mandíbula.

La fonación funciona mediante dos sistemas: Sistemas directos y sistemas indirectos. Los sistemas directos están compuestos por el sistema respiratorio, órganos articulatorios, órganos de resonancia y el aparato fonador. Por otro lado, los sistemas indirectos están conformados por el sistema muscular, el aparato auditivo y el sistema óseo.

La fuente de aire en el sistema de producción de voz se da gracias a los pulmones en la espiración. Según la cantidad de aire que se expulse por los pulmones es la intensidad que se tiene en la voz; mientras que gracias al diafragma se hace posible el variar su duración.

Específicamente, el aparato fonador es el encargado de reproducir el sonido y está conformado por la laringe y las cuerdas vocales.

La laringe, además de proteger las vias respiratorias, es una caja sonora encargada de la fonación (donde se origina el tono fundamental de la voz). Entre las funciones mas importantes de la laringe se encuentran la respiratoria (permite entrada y salida del aire al respirar), esfinteriana (permite realizar esfuerzos), deglutatoria (impide el paso de comida a los pulmones) y de fonación (produce la voz gracias a las cuerdas vocales) [24].

Las cuerdas vocales se encuentran dentro de la laringe y son membranas de tejido de la laringe. Estas producen el sonido mediante la vibración al paso del aire y la aproximación o separación entre si producen voz o silencio respectivamente. Su forma, aunado a las de la garganta, nariz y boca, es decir, las cavidades resonantes, determinan el sonido de la voz en cada persona.

El tono de voz se relaciona con la frecuencia a la que vibran las cuerdas vocales. Específicamente, en los adultos masculinos se presenta una frecuencia baja (120 ciclos por segundo)

por lo que poseen una voz mas grabe. En el caso de las mujeres (250 ciclos por segundo) se presenta una voz aguda; en los niños y jóvenes es aún mas aguda (400 ciclos por segundo). Es necesario aclarar que, aunado a esto, la longitud, tensión y masa de las cuerdas vocales influyen en el tono de voz. Por ejemplo, las cuerdas vocales más gruesas vibran mas despacio por lo que producen un sonido mas grave.

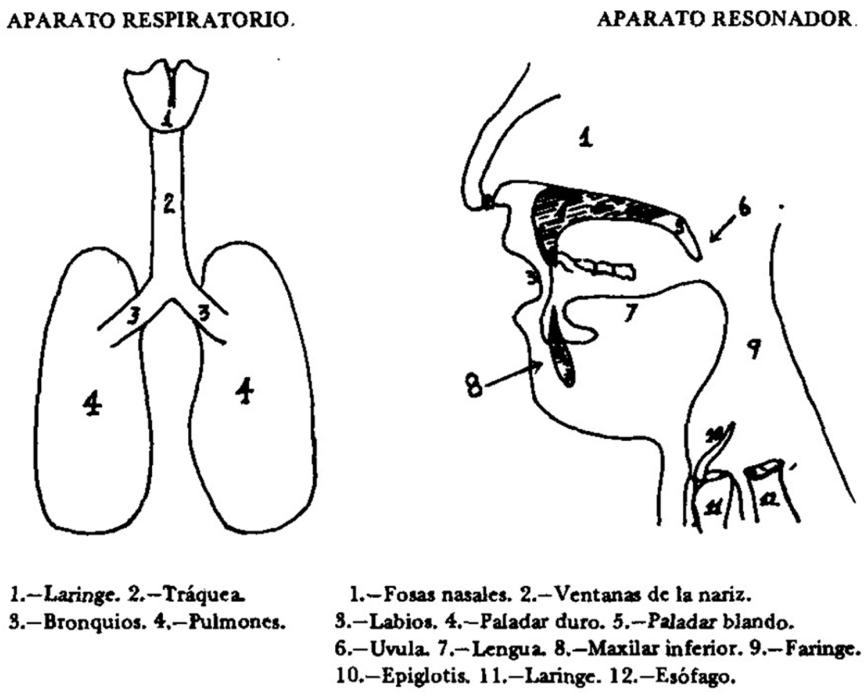


Figura 2.3 Sistema Fonador [6].

En relación al aspecto psicológico y la voz, este parámetro es notorio en la respiración. La respiración varia en un ritmo particular en proporción al estado de ánimo (acelerado ante la excitación y limitado ante la tristeza).

2.3 Fonemas.

En cuanto a la forma en que se clasifica cada sonido, cada lengua o lenguaje posee un conjunto propio de sonidos que el tracto vocal permite producir con naturalidad. A las unidades sonoras propias de un idioma se les llama “fonemas” (representados entre diagonales) y varían

según las características impuestas por cada idioma. La combinación entre fonemas genera sílabas y a su vez, un conjunto de sílabas forma una palabra.

Una de las características más importantes de los fonemas es que, a un enunciado hablado, se le puede descomponer en estas unidades sonoras y si se cambia o sustituye por otro se cambia el contenido lingüístico. Se debe distinguir entre fonemas y letras ya que los fonemas son elementos sonoros de un idioma; mientras que las letras son las representaciones gráficas de los fonemas. En algunas ocasiones un mismo fonema es representado de diversas formas como en el caso de la representación de k, c y q para el mismo sonido.

Las frecuencias formantes permiten clasificar los diferentes sonidos o fonemas. Dichas formantes dependen de la dimensión y forma del tracto vocal (cada configuración provee unas formantes particulares [10]).

En cuestiones de modelado solo es necesario tomar en cuenta las formantes F1, F2 y F3. Las dos primeras formantes las determina la posición de la lengua; cuanto mas baja esté , F1 posee una frecuencia más alta. Para el caso de F2, su frecuencia es mayor cuanto mas hacia adelante esté la lengua.

Por otro lado, la frecuencia fundamental (pitch) o F0 corresponde a la frecuencia de oscilación de las cuerdas vocales en los sonidos sonoros. Si la frecuencia fundamental es mayor que la de la frecuencia de las formantes se hace difícil de distinguir y por ende difícil al reconocer.

Los fonemas, de modo general, se organizan en varios grupos [8]:

- Las vocales orales,
- Las vocales nasales,
- Las semi-vocales,
- Las consonantes nasales,
- Los fricativos (vocalizados y no vocalizados),
- Los plosivos (vocalizados y no vocalizados),

- Las liquidas.

Cabe mencionar que la primera subdivisión, sin embargo, se hace con respecto al modo de excitación del tracto vocal y a su estabilidad; es decir, la separación entre las vocales y consonantes.

Para el caso del idioma español la subdivisión de los fonemas es la siguiente:

- Timbres Básicos,
- Sonidos auxiliares.
- Ataques:
  - Fuertes,
  - Suaves,
  - Sonidos Mixtos.

Timbres Básicos: Producidas con la intervención de las cuerdas vocales. Se denominan como timbres básicos debido a que son variaciones del sonido producido naturalmente por las cuerdas vocales y la columna de aire que se produce mediante la respiración. En esta clasificación se encuentran las vocales.

Para el caso del idioma español las vocales son; /a/, /e/, /i/, /o/ y /u/. De entre ellas la que se produce con naturalidad es la /a/. En cualquier otra, la posición de la lengua se modifica y por ello se crea otra clasificación [20]:

- Vocales Palatales: La lengua se expone gradualmente y se eleva hacia el paladar. En esta clasificación se encuentran las vocales /e/ e /i/.
- Vocales velares: La lengua se contrae y se eleva al velo del paladar. En esta clasificación se encuentran las vocales /o/ y /u/.

De modo general, las vocales se pueden clasificar de acuerdo al grado de abertura de la boca (o tracto vocal), posición de la lengua y grado de sonoridad (ver figura 2.4) . La correcta

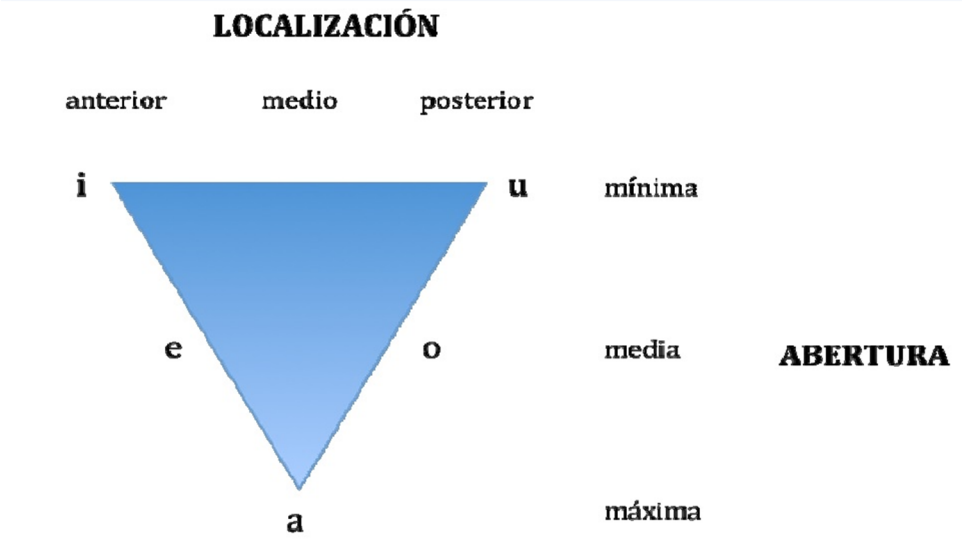


Figura 2.4 Triángulo vocálico del español (HELWAG).

producción de dichos fonemas (sin esfuerzo y con naturalidad) es el material de estudio en la impostación de la voz.

Sonidos auxiliares: Se producen por el aparato resonador sin intervención de las cuerdas vocales. La vibración no se produce en la laringe sino en el paladar blando, la lengua, etc. Corresponden a las consonantes sonoras: /m/, /n/, /l/, /s/, /j/, /r/.

Ataques: No son propiamente un sonido dado que no posee una vibración periódica. Son modos de iniciar un timbre básico o un sonido auxiliar y representan las distintas formas de la oclusión que impide al aire fluir. Una vez liberada, la columna de aire vibra produciendo un timbre básico o un sonido auxiliar; en ese momento el ataque deja de existir. Su naturaleza es momentánea y requiere de su identificación perfecta.

Pueden ser suaves o fuertes según la brusquedad de la liberación de aire. Ellos son: /b/, /p/, /d/, /t/, /k/. Los sonidos mixtos, como caso particular de ataques, se refiere a la sucesión de sonidos auxiliares /l/ o /n/ y el timbre básico /i/, los cuales producen los fonemas /ll/ y /ñ/; o las combinaciones del ataque /k/ con el sonido auxiliar /s/ produciendo /x/.

La calidad de la dicción se fundamenta en la correcta articulación de los ataques.

Algo que se debe tomar en cuenta para poder hablar, es respetar las reglas que posee cada idioma para formar combinaciones de fonemas. Por ejemplo, para el caso del castellano, los

sonidos básicos y los sonidos auxiliares son combinables con cualquier fonema. Los ataques casi nunca son combinables entre si y no pueden producirse aislados; requieren de la combinación con un timbre básico o un sonido auxiliar.

2.4 Dicción y Fonación.

Para poder poseer una buena fonación es indispensable poseer las siguientes características:

1. Suficiente: Mediante el dominio de la respiración.

2. Clara: Correcta producción de cada uno de los sonidos que forman el idioma (aislada o en combinaciones); es decir, con dicción.

3. Expresiva: En entonación, ritmo, intensidad y timbrado para poder captar no solo el significado propio de la palabra pronunciada sino el significado explícito dado por la entonación, velocidad y pausas.

Existen dos ramas de estudio muy importantes dentro de la teoría de producción de voz: Impostación y Dicción. En lo que se refiere al termino conocido como "impostación de la voz" se refiere al aprovechamiento de la espiración (expulsión de aire por los pulmones) para producir sonido con el máximo rendimiento y el minino esfuerzo en la garganta. Tal como lo indica la teoría, se necesita que el aire expulsado por los pulmones sea totalmente puesto en vibración por las cuerdas vocales; y éstas a su vez vibren por la acción del aire y la tensión propia que poseen; y una vez cumplidas estas condiciones, el aire se aproveche en el aparato resonador de forma conveniente. Sin embargo, algunos de los hábitos adquiridos hacen que la capacidad fonadora presente obstáculos en algunas de las etapas descritas.

En lo referente a la dicción, que es el tema central de nuetra investigación, se refiere a la correcta producción de los fonemas y la combinación con los timbres básicos. En otras palabras, es la forma de emplear palabras de forma correcta y acertada en el idioma al que pertenecen sin poner especial interés en lo que estas quieren expresar. Para lograr tener una dicción excelente es fundamental poseer una pronunciación correcta y matizar los sonidos respetando las pausas que sean necesarias.

La dicción buena o mala no tiene que ver con el significado que se desea transmitir ni con aquello que se pretende expresar. Cuando se posee una manera clara de hablar, se consigue un acento correcto y resulta fácil entender a quien se expresa, se habla de una dicción clara o limpia.

Contrario a la buena dicción se encuentran los vicios del lenguaje o defectos en el habla. Sobre este punto, es necesario aclarar que la dicción no está precisamente vinculada al entendimiento; algunas palabras que no poseen buena dicción aún así son reconocibles pero no es tan sencillo de entender como aquellas que poseen buena dicción. Ejemplo de ellas son el utilizar "veniste" en vez de "viniste"; o "andabanos" en vez de "andábamos".

Se puede lograr una dicción correcta y por ende un habla clara, pronunciando cada palabra y sílaba de forma adecuada con el fin de incrementar la memoria muscular en base a repeticiones. Para mejorar la dicción es necesario también tomar en cuenta la velocidad a la que son emitidas las palabras por lo que, se recomienda una velocidad pausada con el fin de prestar atención a cada parte de la estructura del mensaje oral que se desea transmitir; la rapidez excesiva puede generar problemas en la comunicación. Esto no debe ser un sinónimo de monotonía y aburrimiento sino una mejoría en la voz y el ritmo.

En conclusión, para poseer una buena dicción es necesario pronunciar correctamente, acentuar adecuadamente y poseer buena velocidad. Es por ello que, es necesario el desarrollo de un algoritmo capaz de tomar en cuenta estas tres características y en base a ello determinar si se tiene una excelente, buena o mala dicción.

Capítulo 3

Fundamentos sobre reconocimiento de voz.

Como ya se mencionó, el propósito de la voz es la comunicación. La señal de voz, según la teoría de la información, puede ser representada en términos del contenido de información que posee. Dicha señal posee información lingüística en el rango de 200 a 8000 Hz. En un esquema general, la manipulación de una señal de voz se lleva a cabo cuando se convierte de una señal acústico-fonética en una señal eléctrica para después transformarla en una señal discreta que luego será tratada matemáticamente.

La finalidad del tratamiento de señales implica el uso de sistemas digitales con el fin de llevar a cabo operaciones de procesamiento que tengan bajo costo y mejoren el desempeño. Los avances tecnológicos en Hardware han logrado que el tratamiento de señales se convierta en una herramienta multidisciplinaria que engloba, entre otras, las siguientes áreas:

- Teoría de la información y comunicaciones,
- Computación,
- Lingüística,
- Fisiología,
- etc.

El reconocimiento de voz se basa en el conocimiento sobre el proceso del habla y la estructura que conlleva el lenguaje en el ser humano. El fin último del reconocimiento de voz

es hacer que las computadoras logren un nivel suficiente para expresar y comprender la comunicación como lo hace en su naturaleza el ser humano (reconocimiento del habla espontanea [23]).

Entre las ventajas de utilizar el reconocimiento de voz para la comunicación usuario máquina por sobre la comunicación tradicional se tienen [25]:

- Permite tener las manos libres al mismo tiempo que se da una instrucción.
- Permite la movilidad gracias al uso de micrófonos.
- Permite el acceso remoto al identificar cada comando.

Desde esta perspectiva los sistemas de reconocimiento de voz buscan poseer algunas características importantes tales como robustez (buen desempeño en diversos entornos), flexibilidad (extendible a diferentes vocabularios), de gran vocabulario (un léxico superior a 1,000 palabras), entre otros.

Los mayores avances en materia de reconocimiento de voz se desarrollaron en los años 70´s, sin embargo desde finales de los años 40´s principios de los 50´s comenzaron los trabajos enfocados en esta área de estudio. Tal es el caso del *fonetógrafo* desarrollado por los laboratorios Bell en el año de 1952 el cual transcribía la voz en unidades fonéticas. El *fonetógrafo III* (desarrollado en 1960) lograba transcribir letras aisladas de un solo hablante con un proceso de entrenamiento bastante exhaustivo. Para el año de 1956 se desarrollo el primer reconocedor de vocales para múltiples usuarios. Todos estos desarrollos se basaron en el diseño en hardware y de componente, sin embargo, no fue sino hasta los años 60´s cuando se desarrollaron los primeros trabajos en Reconocimiento Automático de Voz en sistemas informáticos [15].

Actualmente las empresas más importantes en desarrollo de aplicaciones en sistemas de reconocimiento de voz son [25]:

- Philips,
- Lernout & Hauspie,
- Sensory Circuits,

- Dragon Systems,
- Speechworks,
- Vocalis,
- Dialogic,
- Novell,
- Microsoft,
- NEC,
- Siemens,
- Intel.

El reconocimiento automático de voz (RAV) se divide en dos áreas [10] enfocadas cada una en una tarea específica:

1. Reconocimiento automático del habla (RAH).
2. Reconocimiento automático de locutor (RAL), el cual se subdivide a su vez en:
    - Identificación automática de locutor (IAL).
    - Verificación automática del locutor (VAL).

Los sistemas de reconocimiento se pueden, además, clasificar como dependientes e independientes del locutor. Para el caso de los sistemas dependientes del vocabulario (cooperativo), el mensaje que se identifica o articula, es una palabra o mensaje fijo (frase pre-establecida); mientras que para el caso en que son independientes del vocabulario (no cooperativo), se posee la libertad para articular cualquier mensaje para ser reconocido. Debe aclararse que, para este último caso, la biblioteca es mucho más extensa en relación al vocabulario que el locutor posee por lo que se puede decir que tiene la "libertad" para pronunciar cualquier mensaje. La eficiencia de los sistemas dependientes es mayor que la de los sistemas independientes.

Otro de los parámetros a identificar y definir es el tipo de reconocimiento que se llevará a cabo: Voz aislada ó continua. Para el primer caso cuando se habla de voz aislada, se trata de la identificación de un elemento (ya sea fonema, sílaba ó palabra) visto desde la perspectiva de una unidad. Por otro lado, la voz continua se corresponde con la forma natural del habla visto desde la perspectiva de una estructura unificada de elementos individuales. En el caso presente el objetivo es la identificación de palabras aisladas por lo que cada uno de los procesos, hasta llegar al reconocimiento, será aplicado a solo una palabra por vez.

El estudio de reconocimiento de voz se basa en tres principios:

1. La información de la señal de voz se puede representar por el espectro en amplitud a corto plazo de la forma de onda de la voz. Esta es la base para la construcción de patrones.

2. El contenido de la voz se puede expresar en forma escrita (una secuencia de símbolos fonéticos o caracteres del alfabeto).

3. Es un proceso cognoscitivo, es decir, la comprensión está ligada a la gramática, semántica y estructura del lenguaje.

La señal de voz puede considerarse como cuasi-estacionaria (varía lentamente) si se analiza en periodos de tiempo relativamente cortos (entre 5 y 100 mseg), mientras que para periodos más largos (mayores a 0.2 segundos) se dice que la señal es no estacionaria. El analizar la señal de voz en periodos cortos (ventanas de análisis) se utiliza para preservar la resolución en frecuencia y reducir los efectos de las variaciones temporales y, reduciendo a si mismo, la no estacionariedad.

El análisis de voz se puede realizar en base a dos métodos clásicos:

1. Análisis puramente temporal: Se basa en la forma de onda. Los modelos basados en la forma de onda van desde el análisis por predicción lineal hasta el uso de modelos AR, MA, ARMA, etc.

2. Una representación espectral: De aquí la necesidad de analizar la señal en tiempos cortos para poder aplicar la Transformada de Fourier de Corta duración (STFT por sus siglas en inglés), la cual solo se aplica a señales periódicas. Esta representación se enfoca

en conocer el contenido de frecuencias y la distribución de la energía en el espectro. La transformada wavelet (sustitución de ventanas por ondículas) también es una herramienta para el modelado en frecuencia.

Para poder lograr el tratamiento de la señal de voz se debe cumplir con la condición fundamental de preservar el contenido del mensaje y que la representación se haga de forma conveniente.

Dentro del procesamiento de señales, la señal de voz de forma discreta puede ser representada de la siguiente forma:

1. Representaciones en cuanto a su forma de Onda (enfocado a preservar la forma de Onda).
2. Representaciones paramétricas (enfocado a la caracterización del sistema o modelo de producción de la señal de voz):
   (a) Parámetros de excitación.
   (b) Parámetros del conducto vocal.

Algunas de las características a tomar en cuenta para elegir la forma de representación son los costos, la flexibilidad de la representación, la calidad de la señal, entre otros; sin embargo, es necesario tomar en cuenta la consideración más importante, es decir, la aplicación final.

La dificultad que se presenta para poder llevar a cabo reconocimiento de voz (en cualquiera de sus áreas) es la variabilidad presente en las señales de voz, es decir, las variaciones que se presentan en cada locutor, las condiciones acústicas, entre otras. Sin embargo dentro de estas variaciones se debe determinar cuáles de ellas son relevantes y cuáles no, es decir, para cada aplicación determinar los parámetros a tomar en cuenta y los que se pueden despreciar.

Existen cuatro enfoques principales que se pueden utilizar para realizar el reconocimiento del habla [11]:

1. Contraste de patrones:
   - Supone al habla como una secuencia de palabras cada uno con un patrón o conjunto de patrones.

- Se compara la unidad de habla entrante con los patrones de referencia almacenados.
- Utilizado en reconocimiento de palabras aisladas o conectadas.
- Esta técnica no permite generalizar los patrones por lo que es necesario crear una bilbioteca para cada hablante lo cual reduce su nivel de aplicación en tareas de reconocimiento avanzadas.

2. Sistemas basados en el conocimiento:

   - Emula y aplica los conocimientos sobre el habla en tareas de reconocimiento que utiliza el ser humano.
   - Usa técnicas con reglas y sistemas expertos, desde el nivel acústico-fonético hasta niveles más complejos.

3. Modelos estocásticos:

   - Hace uso de modelos estocásticos (Modelos Ocultos de Markov) en vez de modelos determinísticos.
   - Utilizado comúnmente para el reconocimiento de palabras continuas.

4. Modelos neuronales o conexionistas:

   - No posee tantas restricciones como los modelos estocásticos.
   - Sus tiempos de entrenamiento son razonables.
   - Se han utilizado también como parte de fusiones entre varios enfoques.

Es necesario aclarar que el uso de cada una o combinación entre ellas dependerá de cual es el objetivo que el reconocimiento pretende alcanzar. A continuación se describen los métodos principales utilizados en el reconocimiento de voz.

3.1 Métodos de análisis para reconocimiento del habla.

Para llevar a cabo el reconocimiento de voz es necesario obtener los parámetros que representen la información espectral contenida en la señal de voz. Para ello, y como se analizará

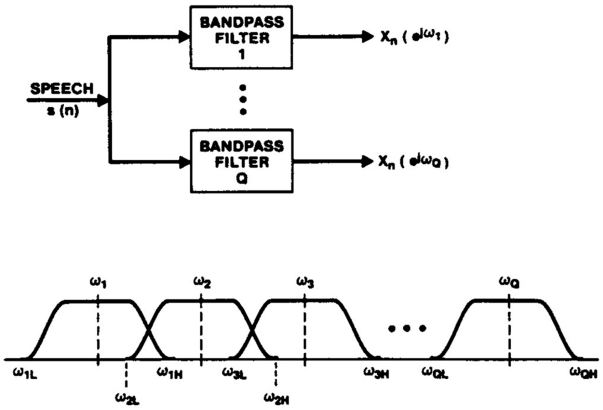


Figura 3.1 Modelo por Banco de Filtros [8].

con mas detalle posteriormente, es necesario aplicar un procesamiento matemático con el fin de adecuar la señal para poder ser analizada posteriormente. Una vez aplicado este "pre-procesamiento" la señal de voz se decodifica para extraer los parámetros propios de la señal. A continuación se presentan algunos de los principales métodos de análisis que se aplican en el reconocimiento de voz.

3.1.1 Bancos de Filtros.

Es uno de los modelos clásicos perteneciente a los métodos de análisis espectral. En el, la señal de voz es filtrada a través de bancos de filtros pasa-banda agrupados en rangos de frecuencia con la finalidad de medir y analizar la energía de la señal de voz en esas frecuencias de interés. Cada filtro puede traslaparse (como se muestra en la figura 3.1) y la salida será $X_n(\mathrm{e}^{jw_i})$; en donde w_i es la frecuencia central de cada filtro expresada como w_i=$\frac{2\pi f_i}{f_m}$ (en radianes).

La salida de la señal al pasar por los bancos de filtros se muestra en la siguiente ecuación:

$$s_i(n) = \sum_{M=0}^{M_{i-1}} h_i(M)s(n-M), \tag{3.1}$$

donde h_i(M) es la respuesta al impulso del i-ésimo filtro pasa banda que tiene duración de M_i muestras. Para cumplir el propósito del banco de filtros, cada señal se pasa a través de una no-linealidad (como si fuera un rectificador de media u onda completa). Dicha no-linealidad

introduce un corrimiento en la señal pasa banda a pasa bajas lo cual crea imágenes de la señal en altas frecuencias. Finalmente se aplica un filtro pasa bajas para eliminar dichas imágenes y poder obtener finalmente una estimación de la energía en cada una de las Q bandas.

Los bancos de filtros pueden implementarse de dos formas diferentes:

- Respuesta Infinita al Impulso (IIR).
- Respuesta Finita al Impulso (FIR).

De entre ellos, los filtros FIR tienen un tiempo de cálculo mayor que los IIR, sin embargo con ellos son más fáciles de obtener características de frecuencias reales.

Otra alternativa es usar bancos de filtros no uniformes los cuales, se diseñan de acuerdo a algún criterio de espaciamiento.

Entre las ventajas que posee el utilizar bancos de filtros se encuentra que con pocos parámetros se puede representar la envolvente espectral aunado al poder tener resoluciones de frecuencias diferentes para cada filtro (esta última característica es utilizada en simulación de procesos auditivos).

3.1.2 Codificación Lineal Predictiva (LPC).

El análisis por predicción lineal o modelo LPC ha brindado buenos resultados en reconocimiento y procesamiento de voz desde hace mucho tiempo. Su principal característica es que con esta técnica se modela el sistema traqueal que produce la voz humana. El análisis LPC considera que la señal que se produce no ocurre de forma aleatoria por lo que existe una relación entre cada una de las muestras de voz. De aquí que se considera que para una muestra de voz obtenida en un tiempo n, $S(n)$ puede ser aproximada utilizando una combinación lineal de p muestras anteriores en el tiempo (ecuación (3.2)).

$$S(n) \approx a_1 S(n-1) + a_2 S(n-2) + \cdots + a_p S(n-p). \tag{3.2}$$

Los coeficientes $a_1, a_2, \ldots, a_p$ son los coeficientes a estimar que se suponen constantes sobre la secuencia analizada. Estos parámetros cambian en función de los parámetros de excitación. Gracias a esta relación entre muestras se puede reducir la cantidad de información y por ende el tiempo de procesamiento.

Cada uno de estos coeficientes pueden ser calculados mediante los siguientes métodos:

1. Autocorrelación.

2. Covarianza.

A diferencia del método de covarianza, el uso del método de autocorrelación asegura la estabilidad del filtro LPC. Los coeficientes generados por la Autocorrelación pueden ser también calculados mediante la ecuación (3.3) .

$$R_a(i) = \sum_{k=0}^{p-i} a_k a_{k+i}, \text{ en donde } 1 \leq i \leq p. \tag{3.3}$$

Entre mas coeficientes se calculen se tiene una $S(n)$ mas fidedigna con respecto de la original; pero dado que este método también es de codificación, por lo general se toman en cuenta entre 10 y 15 parámetros solamente. Sin embargo, la elección del número de coeficientes se relaciona directamente con los recursos computacionales, el tiempo, la exactitud espectral y la aplicación.

El análisis LPC sirve como un modelo para aproximar la envolvente espectral (con pocos recursos computacionales y una buena representación en tiempo y frecuencia) del tracto vocal, lo cual es importante si se considera que la señal contiene regiones cuasi-estacionarias y vocalizadas para los cuales dicho análisis provee una buena aproximación.

Este método realiza un análisis espectral de voz por bloques con un modelo (o filtro) "todos polos". Esto significa que la representación espectral resultante $X_n(e^{jw})$ está condicionada a ser de la forma $\frac{\gamma}{A(e^{jw})}$, en donde $A(e^{jw})$ es un polinomio de orden *p* en el plano *z*.

La salida del análisis LPC brinda un bloque o vector de coeficientes, los cuales especifican paramétricamente el espectro de un modelo "todos polos" que se ajusta al espectro de la señal en el periodo de tiempo para el cual se lleva a cabo el análisis.

Las principales razones por las que se utiliza el análisis LPC sobre otros métodos son:

1. Este análisis permite distinguir una separación entre la fuente (señal de excitación) y el tracto vocal (función de transferencia del tracto vocal).

2. Analíticamente manipulable para su implementación sencilla en software y hardware.

3. Esta técnica puede ser ligada al análisis cepstral.

El análisis LPC permite además, analizar y obtener casi en tiempo real los vectores de coeficientes, ya que el modelo es estable y lineal con los sonidos vocalizados. Un aspecto a tomar en cuenta es que lo importante es obtener las formantes, poder distinguir entre fonemas y tratar de conservar la señal si se quiere reconstruir. La ventaja de utilizar este método es que a partir del análisis LPC, se puede realizar una conversion espectro logarítmica, es decir, transformar los parámetros LPC en parámetros cepstrales.

3.1.3 Análisis Cepstral

La señal de voz se puede analizar por medio de características espectrales por dos razones fundamentales [25]:

- La señal es cuasi estacionaria (en periodos de tiempo corto) y puede ser aproximada por medio de la sumatoria de ondas senoidales.

- La información en la percepción de la voz por medio del oído humano incluye información espectral.

La envolvente espectral cambia lentamente en función de la frecuencia y refleja no solo las características resonantes (o antiresonantes), sino también algunas otras características como la radiación del sonido en labios y nariz.

Con base en estas características, una técnica de análisis comúnmente utilizada es el cepstrum (caso particular de una transformación homomórfica) lo cual es una buena técnica de deconvolución (separación entre la función de excitación y la respuesta del tracto vocal). El cepstrum es definido como la transformada inversa de la amplitud del espectro logarítmico en corto tiempo $|X(w)|$ (transformada inversa del espectro).

La ventaja principal del uso de los coeficientes cepstrales es que aprovechan los beneficios del análisis LPC pero eliminando ruido y perturbación propiamente contenida en las tramas de información.

Para ejemplificar el análisis cepstral, se supone una señal de voz $S(n)$, la cual es el resultado de la convolución entre la señal de excitación $u(n)$ con la respuesta del tracto vocal $h(n)$, es decir:

$$S(n) = u(n) * h(n). \tag{3.4}$$

Si se desea separar las componentes de la ecuación (3.4), es necesario que las señales que la componen posean características frecuenciales muy diferentes (tal y como ocurre en la señal de voz). Si se reconsideran las propiedades de la señal de voz $S(n)$ para que no varíen en un intervalo de tiempo determinado mediante el análisis por ventanas $w(n)$, la señal $x(n)$ quedaría expresada mediante la ecuación:

$$x(n) = w(n)s(n), \text{ en donde } 0 \leq n \leq N-1. \tag{3.5}$$

En base a la ecuación (3.4), la ecuación (3.5) quedaría expresada como:

$$x(n) = w(n)[u(n) * h(n)], \text{ en donde } 0 \leq n \leq N-1. \tag{3.6}$$

En base a investigaciones, la ecuación (3.6) puede ser expresada también como:

$$x(n) \approx [u(n)w(n)] * h(n). \tag{3.7}$$

Finalmente la ecuación (3.7) refleja que la señal u(n) varía mas lentamente que $h(n)$.

Específicamente para obtener el cepstrum $c(n)$ de la señal $x(n)$, es necesario realizar una transformada discreta de Fourier (TDF), aplicar un logaritmo de la magnitud y finalmente una transformación discreta inversa (TDIF), es decir, cumplir con las siguientes ecuaciones [10]:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}Kn}, \text{ en donde } 0 \leq k \leq N-1. \tag{3.8}$$

$$c(n) = \frac{1}{N}\sum_{n=0}^{N-1} \log |X(k)| e^{j\frac{2\pi}{N}Kn}, \text{ en donde } 0 \leq n \leq N-1. \tag{3.9}$$

En la ecuación (3.9), $c(n)$ representa la combinación de las componentes de $x(n)$, ya no por una convolución sino en forma de una suma. Es importante aclarar que no es posible obtener la secuencia $x(n)$ a partir de los parámetros $c(n)$.

Una de las ventajas del análisis LPC es que a partir de los coeficientes de predicción lineal α_k, es posible obtener los coeficientes cepstrales c_i. Para ello se utiliza la siguiente ecuación recursiva:

$$c_i = -\alpha_i - \sum_{k=1}^{i-1}(1 - \frac{k}{i}) \cdot \alpha_k \cdot c_{i-k}, \text{ en donde } 1 \leq i \leq p. \tag{3.10}$$

Dentro del análisis cepstral se tiene alguna nomenclatura equivalente al análisis frecuencial, dada por los siguientes términos:

- frecuency=quefrency,
- spectrum=cepstrum,
- phase=shape,
- magnitude=gamnitude,
- filtering=liftering,
- harmonic=rahmonic,
- period=repiod.

3.1.4 Frecuencias de Mel (MFCC)

Uno de los métodos para la extracción de características más utilizado comercialmente son los Coeficientes Cesptrales en Escala de Mel (MFCC)[25]. Este método utiliza la transformada de Fourier para obtener las frecuencias de la señal.

El objetivo es obtener características de la señal basadas en criterios perceptuales (percepción del sistema auditivo en el ser humano). Es definida como el cepstrum de una señal enventanada en el tiempo que ha sido derivada de la aplicación de una STFT, pero con una escala de frecuencia no lineal la cual se aproxima al comportamiento del sistema auditivo humano.

Para la obtención de los coeficientes Mels se utiliza la transformada de Fourier (para la representación espectral de la señal) y el diseño de banco de filtros (para seleccionar las bandas de frecuencia de la señal en análisis).

La idea básica es calcular la energía que aporta a cada banda de frecuencia la señal que se analiza y calcular en términos de un coeficiente para cada valor de energía en banda de frecuencia (coeficiente cepstral).

3.2 Técnicas de reconocimiento de Voz.

Existen diversas técnicas para llevar a cabo el reconocimiento de voz. Algunas de ellas son enfocadas al reconocimiento de palabras aisladas y otras más son comúnmente utilizadas en reconocimiento de habla continua. Sin embargo es necesario tomar en cuenta que la elección de ellas dependerá del objetivo final que se pretende alcanzar.

A continuación se describen algunas de las técnicas principales para llevar a cabo el reconocimiento de voz.

3.2.1 Cuantificación vectorial (CV).

El resultado de utilizar ya sea bancos de filtros, análisis LPC, etc., son vectores característicos que representan las características espectrales variantes en el tiempo de acuerdo a la señal de voz. Denotaremos a los vectores espectrales como X_i, para $i = 1, 2, 3, \ldots, L$, en donde típicamente cada vector es de una dimensión *p* (o al menos *p*):

$$
\begin{aligned}
X_1 &= A_0^1, A_1^1, A_2^1, \ldots, A_p^1, \\
X_2 &= A_0^2, A_1^2, A_2^2, \ldots, A_p^2, \\
&\vdots, \\
X_L &= A_0^L, A_1^L, A_2^L, \ldots, A_p^L.
\end{aligned} \tag{3.11}
$$

La cuantificación vectorial [30] es la generalización de la cuantificación escalar y en ella la cantidad total de información es tratada y almacenada en vectores los cuales representarán al total de información. Dicha técnica fue ampliamente aplicada y estudiada en los años 80´s especialmente en la rama de la comunicación. Andrés Buzo es uno de los mexicanos que ha logrado mayor popularidad en esta área de estudio.

La razón por la que se denomina "codificación" queda justificada desde el punto de vista de la tasa de transmisión de la información en comparación con una señal sin codificar. Por ejemplo, para una señal muestreada a 10 KHZ con amplitudes de voz de 16 bits, la señal de voz tiene una transmisión de 160,000 bits por segundo lo cual impacta directamente con el espacio requerido para almacenar dicha señal sin comprimir. Por otro lado comparándolo con el análisis LPC, si consideramos vectores de dimensión $p = 10$, y obtenemos 100 vectores espectrales por segundo y cada componente espectral está representado por 16 bits, entonces el almacenamiento requerido seria de 16,000 bits por segundo. Es decir, este proceso tiene una tasa de compresión de 10 a 1. Debemos tomar en cuenta que para una p pequeña existe mayor error y, para el caso contrario, cuando p sea grande se acerca más a la señal original pero se disminuye la compresión.

Dicha representación espectral ideal es impráctica por la variabilidad en las propiedades espectrales por lo que es necesario tomar secuencias de duración grande para tener un modelado con mayor precisión. Sin embargo, se debe recalcar que independientemente de la técnica utilizada todos los sistemas de reconocimiento tienen error dado que están basadas en aproximaciones.

Esta técnica es una representación parcialmente eficiente de la información espectral de la señal de voz; gracias a esto se puede llevar a cabo una codificación de la señal de voz aplicable

en codificación fuente para tareas de telecomunicaciones y reconocimiento. La codificación fuente tiene como fin el lograr la mínima distorsión para una tasa de bits de transmisión predeterminada. En cuanto a las tareas de reconocimiento, ha sido aplicada en sistemas independientes y dependientes del vocabulario con tasas de error bajas.

La Cuantificación Vectorial, en el proceso de codificación, inicia con la segmentación de la señal de tal forma que se crean vectores de características (compresión de datos) de la señal. Estos a su vez son comparados con el libro de códigos (vectores almacenados previamente) que contiene los vectores representantes (centroides o palabras código con direcciones o etiquetas). Para el proceso de transmisión, la dirección del libro de código mas similar al vector de la señal, es transmitida al receptor donde se utiliza el mismo libro de código para poder extraer la información. De esta forma se reconstruye una aproximación de la señal original. En el caso en el cual los vectores y libros de código sean de dimension y tamaño grande respectivamente, la búsqueda debe de ser completa y exhaustiva por lo que se requiere de un procesamiento que conlleva muchas operaciones por segundo. Algunas de las alternativas de solución a este problema es el utilizar vectores LPC para la etapa de entrenamiento o el realizar la búsqueda por árboles, por multietapas, por mallas o alguno de los métodos de codificación rápida Tiempo-calidad.

Entre las ventajas de la representación por Cuantificación Vectorial se encuentran el que se reduce el almacenamiento para el análisis de la información, el tiempo de cálculo para determinar los vectores de análisis espectral y que los cálculos de similitud son reducidos a solo una tabla de estimación. Por otro lado, entre las desventajas se encuentra el que se crea una distorsión espectral inherente en la representación del análisis de cada vector.

El proceso que se lleva a cabo mediante la CV considera que se dispone de una secuencia $X_1, X_2, ..., X_L$ donde X_l es el l-ésimo vector de K componentes; dichos vectores se obtienen al segmentar la señal en tramas temporales y realizar el procedimiento de extracción de parámetros de cada una de las tramas de la señal. El cuantificador vectorial de N entradas y k dimensiones que es una función $q(\cdot)$, asigna a cada vector de entrada $X_l = [X_0, X_1, ..., X_{k-1}]$ un vector $\hat{X}_n = q(X_l)$ de un conjunto finito:

$$\hat{A} = \hat{X}_n; \text{ en donde } n = 1, 2, ..., N. \tag{3.12}$$

donde $\hat{A}$ es el libro de códigos y $\hat{X}_n$ son los centroides (el tamaño del libro de código es el equivalente al número de niveles de cuantización). El cuantificador queda completamente descrito por el conjunto $\hat{A}$ y la partición $S = S_n; n = 1, 2, ..., N$ (forma del subconjunto ó particiones), como en el ejemplo de la figura 3.2.

Para poder saber la eficiencia de un cuantificador es necesario conocer la distorsión promedio producida por dicho cuantificador (ver ecuación (3.13)). Se habla de un cuantificador óptimo cuando la distorsión promedio es minima al cuantificar la secuencias de vectores:

$$D = \frac{1}{L} \sum_{l=1}^{L} d(X_l, q(x - l)), \tag{3.13}$$

en donde *D* es la distorsión promedio y $d(\cdot)$ es una medida de distorsión o distancia espectral.

Un algoritmo de clasificación comúnmente utilizado es el denominado *Algoritmo de K-medias* [4]. El algoritmo comienza asignando un numero deseado de particiones o subgrupos; cada uno representado por un punto centroide en el espacio.

Los centroides son colocados aleatoriamente para después ser comparados con cada punto de la muestra. Luego, cada punto es evaluado y asignado al centroide mas cercano. Luego de que todos los puntos han sido clasificados por este proceso, los centroides son recalculados como la posición media de todos los puntos que se le asignaron.

El proceso continua hasta que los centroides no se muevan, es decir, que los puntos sigan perteneciendo a la región del mismo centroide a través de cada iteración.

3.2.2 Alineación dinámica en el tiempo (DTW).

Existen varios factores que se deben tomar en cuenta en el reconocimiento de voz. Entre ellos, la velocidad a la que son emitidas las muestras de voz (ya sean frases o palabras aisladas). Esto es debido a que varían de locutor a locutor (inter-locutor) y aún entre repeticiones de palabras de un mismo locutor (intra-locutor). Dicho fenómeno tiene una repercusión directa

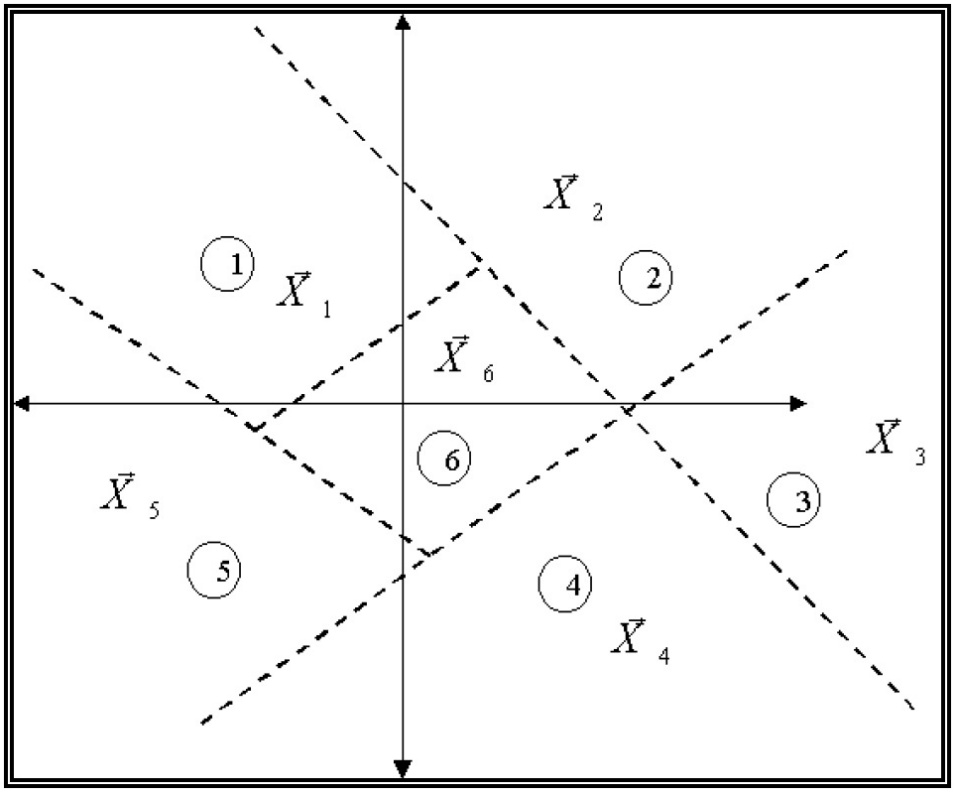


Figura 3.2 Partición de un espacio en 2 dimensiones para llevar a cabo la CV.

cuando se trata del reconocimiento de voz dado que, los eventos acústicos y por tanto los parámetros extraídos, poseen muchas variaciones (temporales, de escala, etc) lo cual dificulta la comparación.

Este efecto hace difícil la comparación en reconocimiento de patrones "punto a punto" dado que, considera que las variaciones de velocidad entre muestras no existen (son lineales) entre repeticiones. Debido a esta consideración, este método posee un alto nivel de error ya que, en la realidad, el comportamiento entre muestras es no lineal.

En base a este comportamiento se hace necesario hacer un alineamiento acústico en el dominio del tiempo, con el fin de minimizar las variaciones presentes en cada muestra de voz. El método más utilizado en los primeros reconocedores para resolver el problema anteriormente descrito fue el de Alineación Dinámica en el Tiempo (DTW)[10].

Entre las aplicaciones donde se ha utilizado el Alineamiento Dinámico en el tiempo se encuentran:

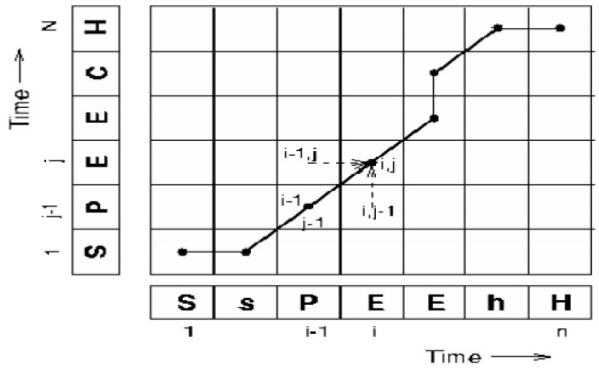


Figura 3.3 Ejemplo de alineamiento entre dos muestras de voz [22].

- Robótica,
- Medicina,
- Reconocimiento de rostros,
- Procesamiento de datos.

Básicamente se trata de tomar ciertas muestras y encontrar el empate con la muestra de voz respecto de la cual se está haciendo la alineación. La idea es entonces, encontrar la mejor alineación de muestras con el fin de causar el mínimo desajuste. En la figura 3.3 se muestra el alineamiento de dos muestras de voz en las cuales, una es la versión distorsionada de otra. En ella se puede apreciar el mejor alineamiento que propone el mejor ajuste entre ambas muestras.

Para entender mejor este proceso, dadas dos muestras de voz:

$$X = X_1, X_2, \ldots, X_N, \tag{3.14}$$

$$Y = Y_1, Y_2, \ldots, Y_M, \tag{3.15}$$

de un mismo o diferente locutor, donde los subíndices N y M (que pueden o no ser iguales debido a la variabilidad de velocidad), se considera una función que relacione la distorsión entre ambas muestras (función de distorsión espectral de tiempo corto). Para este caso, está denotada como se muestra en la siguiente ecuación:

$$D(X, Y) = \sum_{n=1}^{N} d(n, m). \tag{3.16}$$

En la ecuación (3.16), $d(n, m)$ es la distorsión de tiempo, mientras que n y m deben satisfacer la relación:

$$m = \frac{M}{N}n. \tag{3.17}$$

Otra de las técnicas comúnmente utilizadas es la de minimización local no restringida a los límites (MLNL), la cual busca una función de alineamiento entre contornos y condiciones más realistas de alineamiento entre X e Y. La idea básica es definir un contorno guía (ubicado en el eje horizontal del alineamiento) y un contorno esclavo (ubicado en el eje vertical de alineamiento) y tratar de cumplir con la función de alineamiento; sin embargo en la implementación de este método es necesario definir y delimitar un área de alineamiento para limitar el número de rutas que se pueden tomar para llevar a cabo esta tarea. Para encontrar la ruta óptima, y debido a que el contorno guía ha de ser mas corto que el contorno esclavo, se utiliza una medida de distorsión. Este método además, permite obtener la zona donde se tiene información acústica y aquella donde existen silencios.

La función de alineamiento entre n y m estará definida como:

$$m = w(n). \tag{3.18}$$

Otro de los métodos alternativos para llevar a cabo el alineamiento es el denominado "algoritmo de programación dinámica" (optimización dinámica) el cual permite obtener la ruta óptima sin tener que obtener todas las rutas posibles.

Existen algunas restricciones que se aplican para poder reducir el tiempo de cálculo de la ruta óptima las cuales son [22]:

1. Cada trama debe ser alineada sin excepción.

2. Los caminos de coincidencia no pueden ir hacia atrás en el tiempo.

3. Las distancias locales se suman para encontrar la distancia global del proceso.

Ha sido utilizado junto con parámetros cepstrales para tareas de reconocimiento dependiente del vocabulario y se fundamenta en el hecho de que la ruta óptima para el alineamiento

se puede obtener si se minimiza localmente la distancia puntual. Es decir, la distancia acumulada desde cualquiera de los puntos de la ruta óptima hasta el inicio o fin, permite obtener la ruta de alineamiento óptimo.

La distancia acumulada sobre cualquier punto se puede obtener mediante la siguiente ecuación recursiva:

$$D(n,m) = \begin{cases} d(n,m) + \min[D(n-1,m-1), \\ \qquad D(n-1,m), D(n,m-1)]. \end{cases} \tag{3.19}$$

En donde:

$$D(1,1) = d(1,1). \tag{3.20}$$

El método descrito anteriormente, y como ya se mencionó, en forma general es conocido como "Programación Dinámica", pero al ser aplicado en plantillas para el reconocimiento del habla se conoce como "Alineación Dinámica en el Tiempo " [22].

3.2.3 Modelos Ocultos de Markov (HMM).

La aproximación de patrones no modela estadísticamente a las señales de voz por lo que posee ciertas restricciones. Por otro lado las técnicas estadísticas han sido aplicadas en el agrupamiento para crear patrones de referencia. Mediante ello se mejora la clasificación de patrones y se realiza una simplificación dado que se utilizan multiples señales de referencia y se caracterizan mejor las variaciones de diferentes pronunciaciones.

Una de las herramientas que permiten caracterizar estadísticamente las propiedades de un patron o una señal de voz son los Modelos Ocultos de Markov(MOM ó HMM)[14]. Los HMM son considerados una extensión de las Cadenas Ocultas de Markov (caso continuo o discreto de las Cadenas de Markov) y fueron desarrollados por primera vez por un grupo de IBM en 1970. Para 1976 se incorporaron y desarrollaron investigaciones con redes neuronales, DTW y HMM para incorporar información semántica y de sintaxis con el fin de lograr más allá del reconocimiento del habla la comprensión del lenguaje. Los Modelos Ocultos de Markov tuvieron gran impacto hasta mediados de 1980, siendo implementados hasta la fecha en algunos

sistemas comerciales (Sistema de reconocimiento de voz de la marca Apple llamado Siri como lo indica [5]).

El fundamento de esta técnica se basa en considerar que la señal de voz es un proceso estocástico paramétrico y que dichos parámetros pueden ser estimados y definidos de manera precisa. Por esto los Modelos Ocultos de Markov son una técnica que provee una forma de reconocimiento fiable que puede ser aplicable además en varias areas.

Es necesario entender un Modelo Evidente de Markov para entender un Modelo Oculto de Markov [10]. Se considera a un espacio de estados como un número finito de estados posibles $S_1, S_2, \ldots, S_N$ en una sucesión de pruebas, luego en cada una de las pruebas ocurre un cambio de estado asociado con una probabilidad para dicho estado. Finalmente la idea es encontrar la probabilidad de pasar a un estado en cierto instate dado que se encontraba en algún estado.

En los Modelos Evidentes de Markov a cada estado corresponde un evento determinísticamente observable y por tanto la salida en cualquier estado no es aleatorio. Por otro lado, en Modelos Ocultos de Markov la observación es una función probabilística del estado, es decir, es un proceso doblemente empotrado; un proceso estocástico que no es observable directamente (se encuentra oculto), es decir, observable solo a través de otro conjunto de procesos estocásticos que producen una secuencia de observación.

Para aplicar un HMM y en general para las tareas de reconocimiento, es necesario tomar en cuenta dos etapas: entrenamiento y prueba. En la etapa denominada *entrenamiento* se crean los modelos de referencia mediante la extracción de parámentos de cada repetición utilizados para caracterizar a cada modelo (ver figura 3.4). Por otro lado, la etapa de reconocimiento propiamente dicha, se lleva a cabo en la fase de *prueba* en la cual, se realiza la comparación entre el modelo entrenado y la muestra a identificar. Esto consiste en obtener el valor del logaritmo de la probabilidad de observar una secuencia de símbolos contenida en el vector de símbolos para el modelo Markoviano entrenado previamente. En otras palabras, la probabilidad de que la muestra (repetición) a identificar haya sido generada por ese modelo en específico.

Un modelo oculto de Markov esta caracterizado por [29]:

- El **número de estados en un modelo** (N). Los estados individuales se denotan con la letra q en el tiempo t.

- El **número de símbolos distintos de observación en cada estado** (M). Al conjunto de observaciones se le denota como $O = o_1, o_2, o_3, \ldots, o_M$, mientras que a los símbolos individuales se les denota como $V = v_1, v_2, v_3, \ldots, v_M$.

- La **matriz de probabilidad de transición de estados** ($A = [a_{ij}]$). La suma de los elementos de fila de la matriz es la unidad, es decir, misma probabilidad de pasar de un estado a otro. Expresada como:

$$a_{ij} = P[q_{t+1} = j / q_t = i], \tag{3.21}$$

$$1 \leq i \leq N, \tag{3.22}$$

$$1 \leq j \leq N. \tag{3.23}$$

- La **distribución de probabilidad de los símbolos en el estado** ($B = [b_j(k)]$), es decir:

$$b_j(k) = P[O_t = V_k / q_t = j], \tag{3.24}$$

$$1 \leq k \leq M. \tag{3.25}$$

- La **distribución de probabilidad inicial de estados** ($\pi = [\pi_i]$). Probabilidad de iniciar en un estado cualquiera:

$$\pi_i = P[q_i = i], \tag{3.26}$$

$$1 \leq i \leq N. \tag{3.27}$$

Con estos valores, el HMM puede generar una secuencia de observaciones:

$$O = O_1, O_2, \ldots, O_T. \tag{3.28}$$

De manera compacta, un HMM se puede expresar de manera completa como:

$$\lambda = (A, B, \pi). \tag{3.29}$$

En la ecuación (3.29), λ es el nombre de un HMM particular ó palabra específica.

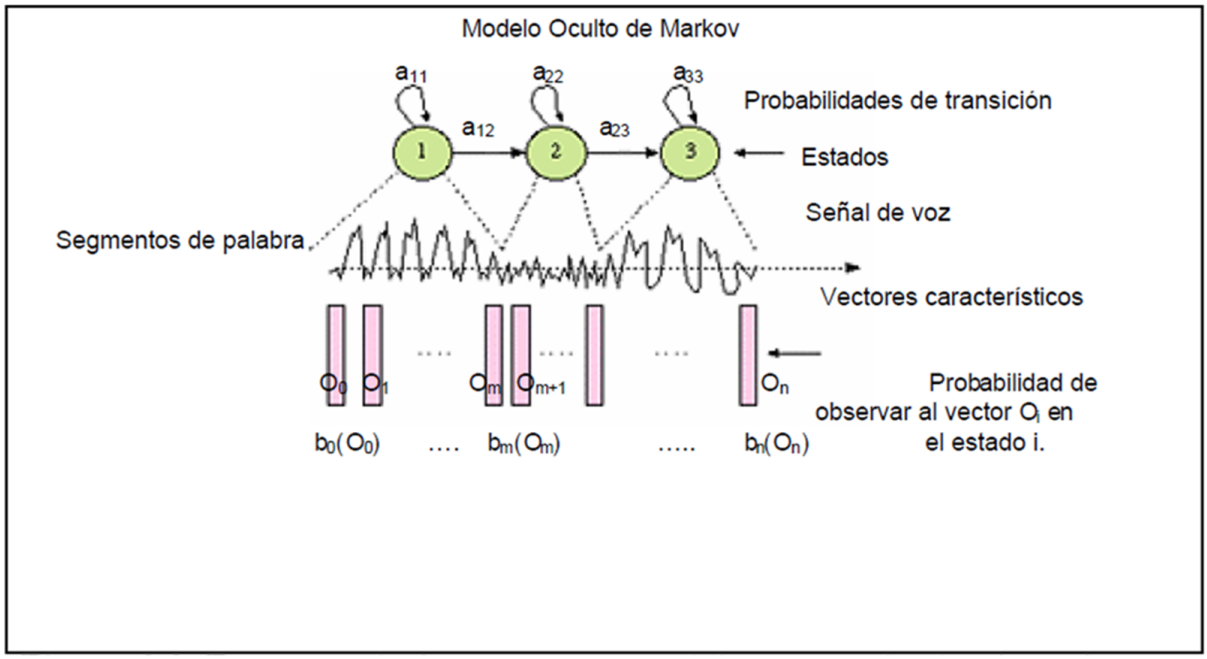


Figura 3.4 Extracción de características para conformar un HMM [15].

Existen, dentro de la teoría de Modelos Ocultos de Markov, tres problemas básicos que se deben resolver:

Problema 1 (Evaluación): Dada la secuencia de observaciones $O = O_1, O_2, \ldots, O_T$ y un modelo $\lambda = (A, B, \pi)$: ¿Cómo calcular eficientemente la $P_r(O, \lambda)$, es decir, la probabilidad de producir una secuencia de observaciones a partir del modelo (elegir el mejor modelo que se ajuste a la señal)? La resolución de este problema se aplica en reconocimiento de voz aislada.

Su utilidad consiste en lo siguiente, de entre la variabilidad de modelos, conocer cual es aquel más probable que genere una secuencia de símbolos. De entre la variabilidad de métodos, y debido a la complejidad de sumar todas las posibles secuencias de estados, se opta por utilizar el algoritmo "**hacia adelante (forward)**". Primeramente, es necesario definir la probabilidad

de observar una cadena de símbolos hasta el tiempo t (observación parcial) con un estado q_i en el instante t para un modelo.

$$\alpha_t(i) = Pr(O_1, O_2, \ldots, O_t, X_t = q_i|\lambda) \tag{3.30}$$

La probabilidad inicial de avanzar es la probabilidad conjunta del estado respecto de la observación (ecuación (3.33)), es decir, se calcula la probabilidad de que el primer estado genere el primer símbolo de la observación. Después, se calculan las probabilidades de que se observe el evento $O_1, O_2, ..., O_T$.

$$\alpha_1(i) = \pi_i b_i(O_i), \text{ en donde } 1 \leq i \leq N. \tag{3.31}$$

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i)a_{ij}]b_j(O_{t+1}), \text{ en donde } 1 \leq t \leq T, 1 \leq j \leq N. \tag{3.32}$$

Mediante la ecuación (3.32) , se calcula la probabilidad de que se llegue, en el tiempo $t+1$, al estado q_i una vez realizadas las observaciones de los estados anteriores a este tiempo y el actual.

Una vez calculadas todas las probabilidades de la ecuación anterior, el resultado final se obtiene mediante la ecuación:

$$P_r(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{3.33}$$

Gracias al cálculo de esta probabilidad, para cada modelo, se puede llevar a cabo el reconocimiento.

Problema 2 (Decodificación): Dada la secuencia de observaciones $O = O_1, O_2, \ldots, O_T$ y un modelo $\lambda = (A, B, \pi)$: ¿Cómo se puede elegir la secuencia de estados correspondiente $Q = q_1, q_2, \ldots, q_T$ que es óptima en algún sentido, es decir, la secuencia que "explica" mejor las observaciones? La resolución de este problema se aplica en voz continua.

El algoritmo que se encarga de resolver este problema es el "**algoritmo de Viterbi**". Mediante este algoritmo se puede identificar el tipo de modelo que es más adecuado para cada

aplicación; es decir, permite analizar de mejor manera un modelo y de esta forma adecuarlo al objetivo en el que se pretende utilizar.

Con el fin de encontrar, a partir de la observación, la secuencia de estados $X_1X_2X_3\cdots X_T$ que la definan, es necesario encontrar (sobre un camino dado en el tiempo) la probabilidad mas alta con la observación parcial y terminando en el último estado (q_i):

$$\delta_t(i) = \max_{X_1X_2X_3\cdots X_{t-1}} Pr(X_1X_2X_3\cdots X_T = i, O_1, O_2, \ldots, O_t|\lambda), \tag{3.34}$$

$$\delta_{t+1}(j) = [\max \delta_t(i)a_{ij}] * b_j(O_t + 1). \tag{3.35}$$

De las ecuaciones anteriores, los valores obtenidos se almacenan en un vector (ψ). El proceso que continua se define como:

Inicialización:

$$\delta_t(i) = \pi_i b_j(O_1), \text{ en donde } 1 \leq i \leq N; \tag{3.36}$$

$$\psi_1 = 0. \tag{3.37}$$

Recursión:

$$\delta_t(j) = \max_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}] * b_j(O_t), \text{ en donde } 2 \leq t \leq T, i \leq j \leq N; \tag{3.38}$$

$$\psi_t(j) = \max_{1\leq i\leq N}[\delta_{t-1}(i)]a_{ij}, \text{ en donde } 2 \leq t \leq T, i \leq i \leq N. \tag{3.39}$$

Finalización:

$$p^* = \max_{1\leq i\leq N}[\delta_t(i)], \tag{3.40}$$

$$X_t^* = arg \max_{1\leq i\leq N}[\delta_t(i)]. \tag{3.41}$$

$$X_t^* = \psi_{t+1}X_{t+1}^*, \text{ en donde } t = T-1, T-2, ..., 1. \tag{3.42}$$

La secuencia de estados final se encuentra por medio de la ecuación (3.42).

Problema 3 (Aprendizaje): ¿Cómo se pueden ajustar los parámetros λ del modelo para maximizar $P_r(O, \lambda)$?. La resolución de este problema determina de manera automática los parámetros para el entrenamiento de manera correcta.

El método utilizado para la resolución de este problema es el "**algoritmo de Baum-Welch**" [21]. En otras palabras, para cada una de las secuencia de observaciones $O = O_1, O_2, \ldots, O_T$, el algoritmo Baum-Welch permite obtener los parámetros λ de un modelo que maximizan la probabilidad de dicha secuencia $Pr(O_j)$. Es necesario definir la probabilidad ($E_t(i, j)$) de estar en un estado (q_i) en un tiempo y otro estado (q_j) en un tiempo posterior dada una cadena de símbolos para un modelo en específico (ver ecuación 3.43).

$$E_t(i, j) = Pr(X_t = q_i, X_t + 1 = q_j | O, \lambda). \tag{3.43}$$

Si $\gamma_t(i)$ se define como la probabilidad de estar en un estado (q_i) en un instante de tiempo dadas las secuencias de observación y el modelo, entonces:

$$\gamma_t(i) = \sum_{t=1}^{T-1} E_t(i, j). \tag{3.44}$$

La ecuación anterior indica el número de transiciones necesarias para llegar desde un estado en un tiempo a uno posterior, es decir, desde (q_i) a (q_j) en base a su probabilidad.

En base a lo anterior se caracteriza a un modelo, es decir:

- La frecuencia del estado en el instante de tiempo $t = 1$, es decir π_i (γ_1(i)).
- El número de transiciones esperadas:

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

- El número de veces que el modelo está en el estado *j*:

$$b_j(k) = \frac{\sum_{t=1:O_t:O_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Por medio del cálculo iterativo de estos parámetros es posible mejorar la probabilidad de observar la secuencia de observaciones en un modelo.

Capítulo 4

Sistemas de reconocimiento propuestos.

El objetivo básico del reconocimiento de voz es que este pueda ser aplicado a sistemas mas complejos para desempeñar una tarea de forma mas sencilla. Debido a esto, y como se mencionó en el capítulo anterior, la forma en como se lleva a cabo el reconocimiento depende del objetivo que este pretenda alcanzar.

Los sistemas de reconocimiento de voz utilizados en la actualidad pueden variar en estructuras tan diversas como el diseñador lo desee, sin embargo, de entre la variabilidad existente, algunos son los procesos fundamentales que deben de poseer para poder ser considerados como sistemas de reconocimiento. Al respecto, el proceso de reconocimiento, independiente de la técnica utilizada, se puede definir como se muestra en la figura 4.1 .

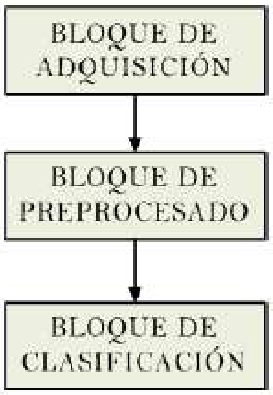


Figura 4.1 Esquema general utilizado en el reconocimiento de voz.

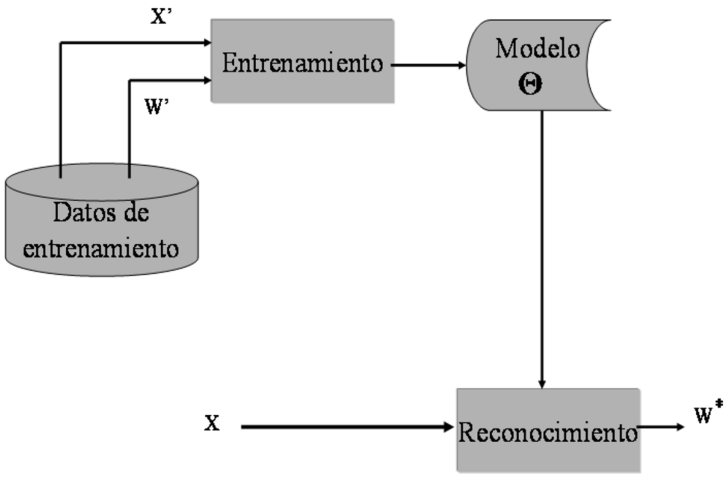


Figura 4.2 Esquema de funcionamiento para reconocimiento con Modelos Ocultos de Markov.

Por otro lado, para el caso específico cuando se utilizan Modelos Ocultos de Markov, las etapas necesarias para llevar a cabo el reconocimiento de voz se resumen en la figura 4.2. De este proceso se pueden diferenciar dos etapas fundamentales:

1. **Etapa de Entrenamiento:** Se crean los modelos de referencia con respecto al corpus de voces correspondiente al reconocimiento. El modelo se puede crear tomando un gran número de referencias (repeticiones) de una palabra en específico (entre mayor sea la cantidad de información con la que se entrene un modelo en especifico mas adecuado será el modelo resultante). Este proceso se puede observar en el ser humano considerando su edad temprana (proceso de aprendizaje).

2. **Etapa de Prueba:** En esta etapa se lleva a cabo el reconocimiento en base a los modelos creados y la palabra a reconocer. En base al desempeño que se obtenga en esta tarea se puede calificar la eficiencia del reconocedor. El proceso que se lleva a cabo es probar la muestra de voz a reconocer, con cuál modelo se relaciona mejor y que, por ende, será la palabra reconocida.

En las siguientes secciones se describe la estructura de reconocimiento así como los métodos propuestos en el presente trabajo de investigación para realizar las pruebas de dicción.

4.1 Estructura de reconocimiento.

De entre la variabilidad de estructuras utilizadas en reconocimiento de voz, la que se propone retomar para el trabajo presente consta de las siguientes partes principales:

- Adquisición,
- Pre-procesamiento,
- Extracción de características,
- Entrenamiento y prueba.

A continuación se describe de forma detallada cada una de estas etapas.

4.1.1 Adquisición.

La primera etapa dentro del diseño de un sistema de reconocimiento consiste en definir y delimitar el *corpus de voces*. El *corpus de voces* no es mas que el conjunto de archivos de voz en forma digital que, junto con sus etiquetas (¿a qué palabra corresponde cada archivo?), permite llevar a cabo las fases de entrenamiento del sistema. Desde una perspectiva más amplia, según las palabras que conformen el corpus de voces serán las palabras que se podrán reconocer (constituyendo bases de datos que van de pequeñas a muy grandes).

En esta etapa se define si se realizará el reconocimiento mono o multi locutor (uno o varios locutores realizan las repeticiones del corpus de voz) y si es dependiente o independiente del vocabulario (según la cantidad de palabras que se puedan reconocer). La efectividad de utilizar un corpus de voces en conjunto con los Modelos de Markov es que, una vez extraídas sus características para la creación del modelo, ya no es necesario utilizar la totalidad de archivos para la etapa de reconocimiento, sino solamente el modelo entrenado y la etiqueta correspondiente (en caso de que sea más de una palabra a reconocer).

Entre la configuración que se puede llevar a cabo dentro del proceso de adquisición se encuentra el tipo de canal (monoaural o estéreo), tamaño de la palabra del sistema digital (4, 8 o 16 bits) y la frecuencia de muestreo (11025, 22050 o 44100 Hz) así como el formato en

que se manejarán las repeticiones de voz (archivos tipo .wav, .mp3, etc.) [15]. Es necesario aclarar que las condiciones de captura de las muestras de voz que conforman el corpus de voces están directamente ligadas con el modelado que se realiza y por tanto con la eficiencia de reconocimiento. Factores tales como ruido ambiental, tipo de micrófono a utilizar, condiciones acústicas inherentes del lugar de grabación y la distancia al micrófono (entre otras), son factores a tomar en cuenta antes de realizar las grabaciones de voz (ver figura 4.3).

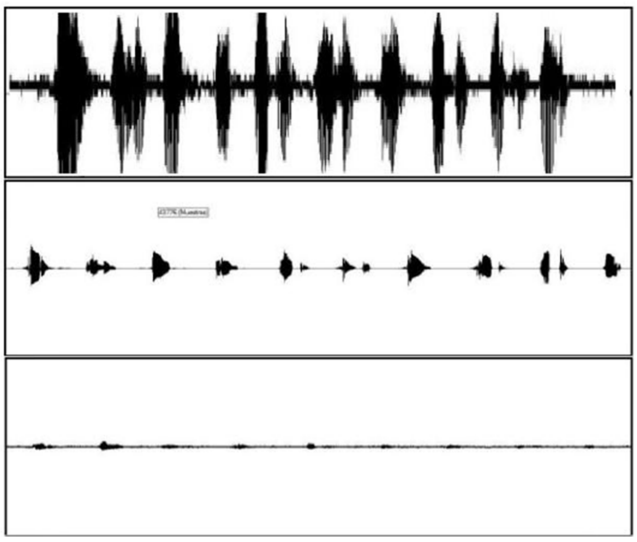


Figura 4.3 Grabaciones con amplitud: saturada (mucho ruido y poca separación entre palabras), media (adecuada) y baja (volumen bajo)[25].

4.1.2 Pre-procesamiento.

Una vez que se han definido cuales son las palabras a reconocer y con las cuales se ha de conformar el corpus de voces, es necesario aplicar un procesamiento previo a la extracción de la información contenida en la señal de voz. Este proceso "normaliza" las características temporales de las muestras de voz almacenadas o adquiridas directamente del micrófono en tiempo real.

Las etapas del pre-procesamiento propuestas son:

- Aplicación de filtro pasa voz,

- Recorte de la señal de voz,
- Pre-énfasis,
- Normalización,
- Segmentación,
- Ventaneo.

Aplicación de filtro "**pasa voz**". Es necesario aplicar como primera etapa de pre-procesamiento un filtro capaz de separar y dejar pasar solo las frecuencias de la voz humana y eliminar el ruido que pudiera ser captado durante el proceso de adquisición por el micrófono; este proceso se logra aplicando un filtrado digital pasa bajas (ya que la frecuencia de la voz humana está comprendida entre 200 y 8000 Hz).

En el diseño del filtro es necesario tomar en cuenta la elección de dos parámetros principales que condicionan el sistema (ver figura 4.4):

- Orden del filtro: La elección de este parámetro relaciona al tiempo de cómputo y la pendiente en la zona de transición.
- Especificaciones en frecuencia: Elección de la frecuencia de corte y banda de rechazo.

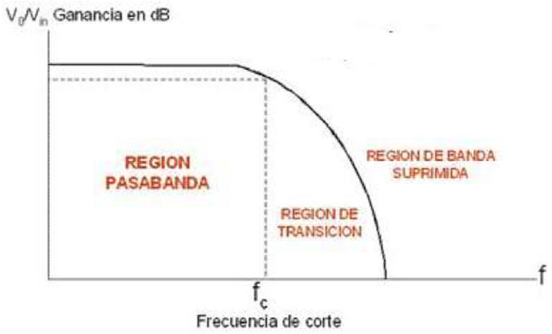


Figura 4.4 Representación gráfica de los parámetros principales de un filtro pasa bajas.

Recorte de la señal de voz (detector de voz). Un elemento importante a tomar en cuenta es que la señal de voz que se procesa posee fragmentos con información de voz y algunas

otras con silencios. Es necesario, por tanto, el uso de un algoritmo capaz de detectar, de la señal de voz, aquella parte que corresponde a los fragmentos de voz que se desean analizar. Es necesario considerar todo esto con la finalidad de no perder tiempo de cálculo en tramas que no contienen información y así evitar errores en el análisis de la señal.

Para el caso del reconocimiento de voz de palabras aisladas, el algoritmo de detección de voz solo ha de distinguir entre la información de voz, las pausas propias al formar las palabras y aquellas que corresponde a los silencios antes y después de la misma; sin embargo, para el caso del reconocimiento de palabras continuas, se hace necesario reconocer además, las pausas o silencios correspondientes a la separación entre palabras al formar una oración.

Evitar las fallas en la detección de inicio y fin es muy importante en los sistemas de reconocimiento de voz dado que, si se detecta erróneamente como ruido una parte de la señal de voz (como lo es el caso de confundir el espectro de los fonemas fricativos con el de la respiración) o viceversa, puede causar una falla total en el proceso del reconocimiento. De entre estos problemas, algunos otros que se pueden encontrar en la detección de voz se encuentran:

- La detección de fonemas nasales al final de una palabra (poseen baja energía),
- Resonancia en los micrófonos después de pronunciar una vocal,
- Ruido confundible con voz,
- Silencios entre palabras (confundibles con falso principio y fin de las mismas).

En este proceso, el cálculo de la energía en corto tiempo y la energía promedio poseen un papel fundamental (ecuaciones 4.1 y 4.2). Mediante ellas y el tiempo de detección energética consecutiva entre muestras, es posible diferenciar las partes vocalizadas de las no vocalizadas (en base a un umbral).

$$E_n = \sum_{k=1}^{W_n} |X[k]|^2 W[n-k], \tag{4.1}$$

$$E_{avg} = \frac{1}{N} \sum_{k=1}^{N} |X[k]|^2. \tag{4.2}$$

La elección del valor de umbral es muy importante dado que dependiendo de su elección será la efectividad del detector de voz (rechaza o acepta una ventana de energía como información vocalizada).

Pre-énfasis. Existe una caída de frecuencia de las altas frecuencias de la voz al salir de los labios; el filtro de pre-énfasis se utiliza para enfatizar las frecuencias formantes (enfatiza bajas frecuencias y disminuye altas frecuencias) y por tanto no perderlas en el proceso posterior de segmentación. Este algoritmo se utiliza en audio para dar brillantez al sonido; mientras que en el caso de reconocimiento de voz se utiliza para reducir los efectos de redondeo de los algoritmos para esas frecuencias. A este proceso también se le conoce como "Ecualización".

Básicamente se trata de un filtro digital pasa altas de primer orden descrito por la función de transferencia:

$$H(z) = 1 - az^{-1}. \tag{4.3}$$

La utilidad de este filtro es que remueve las componentes de corriente directa de la señal (si el valor es muy grande domina el estimado del espectro de la señal) y que aplana (suaviza) espectralmente la señal de voz.

Normalización. Dado que la señal de voz es una señal irregular aún entre repeticiones de un mismo locutor, se busca un proceso que pueda hacer que la señal de voz posea amplitudes no tan pequeñas, pero no tan grandes que puedan llevar a la saturación. La normalización ajusta la amplitud y hace homogéneas las muestras de voz; responde a una necesidad de la conservación de la potencia, es decir, que la potencia de la señal de entrada (antes de la segmentación) sea aproximadamente igual a la potencia de la señal de salida (después del ventaneo). Luego del proceso de normalización, las muestras quedan escaladas de tal forma que el valor más grande que pueden tomar es la unidad.

Segmentación y Ventaneo. La señal de voz es una señal de tipo aleatoria, sin embargo, para poder analizarla y extraer sus características, es necesario subdividir la señal en tramas temporales. A la señal de voz se le podrá considerar como "estacionaria" en intervalos de tiempo cortos (entre 20ms y 45ms) por lo que una segmentación de la señal en dicho intervalo resulta conveniente. Es necesario además, llevar a cabo durante la segmentación, un traslape

Tabla 4.1 Filtros comunmente utilizados en reconocimiento de voz

Ventana Rectangular	$a(n) = 1$ en el intervalo temporal del bloque $(0 \leq n \leq L-1)$ $a(n) = 0$ en el resto de la señal.
Ventana de Hanning	$a(n) = 0.5 - 0.5 * \cos(\frac{2\pi n}{L})$ en $(0 \leq n \leq L-1)$ $a(n) = 0$ en el resto de la señal.
Ventana de Hamming	$a(n) = 0.5 - 0.46 * \cos(\frac{2\pi n}{L})$ en $(0 \leq n \leq L-1)$ $a(n) = 0$ en el resto de la señal.

entre muestras (usualmente de 1/2 o 1/3 del tamaño de la trama); de manera que exista una transición suave entre muestras y no una discontinuidad entre tramas consecutivas y pérdida de información.

Una vez aplicada la segmentación, se procede a realizar el enventanado de cada trama obtenida. El proceso para llevar a cabo el enventanado queda definido mediante la siguiente ecuación:

$$z(n) = x(n)a(n); \tag{4.4}$$

donde $z(n)$ es la trama resultante, $x(n)$ es la trama y $a(n)$ es la ventana utilizada.

La ventana por la que se multiplica cada una de las tramas elimina los problemas causados por las variaciones rápidas en los extremos de cada una (el ancho de cada ventana es del mismo tamaño que el ancho de cada trama). En otras palabras, se pondera la ventana de tal forma que los pesos de las muestras de los extremos de las tramas sean menores que los pesos de las muestras centrales donde se ubica la información más importante de la trama. En la tabla 4.1 se muestran las ventanas más utilizadas en procesamiento de señales y sus características [15].

La ventana más utilizada en reconocimiento de voz es la ventana Hamming (que es un caso especial de una ventana Hanning). Dicha ventana queda representada gráficamente como se muestra en la figura 4.5.

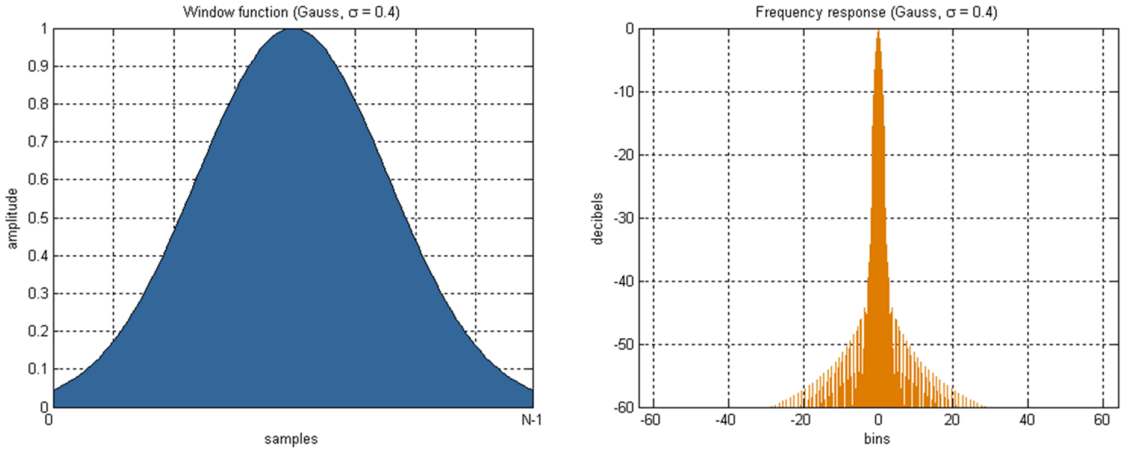


Figura 4.5 Ventana Hamming.

4.1.3 Extracción de características.

Una vez que la señal de voz ha sido adecuada en el pre-procesamiento, la etapa siguiente corresponde a la extracción de la información contenida en la señal de voz. Esto se realiza con el fin de obtener una secuencia de vectores o parámetros que describan la información de interés contenida en la señal. Es necesario aclarar que entre más parámetros posean estos vectores, más difícil se vuelve su tratamiento e implementación.

A este proceso se le conoce como "parametrización" o "caracterización" y consiste en obtener las características propias o distintivas de la señal de voz en análisis con el fin de poder realizar de manera más sencilla las comparaciones necesarias para llevar a cabo el reconocimiento.

El método utilizado para llevar a cabo este proceso depende de la aplicación en que se busca aplicar el reconocimiento. Si bien es cierto que las técnicas de LPC, LPC-Cepstrum, Cepstrum, etre otros, han sido los métodos más populares para la extracción de características; para el caso específico en que se requiere un reconocimiento que tome en cuenta condiciones tales como la pronunciación o claridad con que se menciona una palabra, el método de cálculo de energía por trama, además de ser una de las técnicas mas sencillas en implementación, ha demostrado ser la más adecuada [20].

Una vez obtenidos los vectores de energía, es necesario procesar estos vectores mediante una cuantificación. La cuantificación consiste en asignar a cada valor de energía obtenido, un

valor respecto de un tabulador en niveles; es decir, asignar cada valor de energía a un valor discreto de un conjunto de valores preestablecidos. El proceso que se lleva a cabo, a grandes rasgos, consiste en dividir la amplitud máxima de la señal en niveles o casillas a la cual le corresponde un rango. Así por ejemplo, si una casilla asignada con el nivel de 2 admite valores entre 0.2 y 0.3 y el valor a cuantificar es de 0.25, este valor se cuantifica con el valor de 2 dado que es la casilla que le corresponde. De aquí la necesidad de definir el "paso" entre escalones al momento de cuantificar la señal. El tamaño del escalón (paso) se puede definir en base a la siguiente ecuación:

$$Escalon = \frac{Vmax. - Vmin.}{simbolos}. \tag{4.5}$$

donde $Escalon$ es el valor del paso, $Vmax.$ es el valor máximo encontrado en la señal de voz, $Vmin.$ es el valor mínimo encontrado en la señal de voz y $simbolos$ es el número de valores discretos posibles en los que se puede cuantificar la señal. Previo a este proceso, se propone utilizar una normalización con el fin de pre-establecer los valores en el rango de 0 a 1.

Debido a que algunas de las variaciones no se pueden diferenciar (para el caso del ejemplo anterior, cuando una muestra tiene el valor de 0.25 y 0.9 se cuantifica con el mismo valor), el concepto de "error de cuantificación" ha de ser tomado en cuenta en el diseño del cuantificador (ver figura 4.6).

En base a si el "paso" del escalón es fijo o variable, se puede decir que los cuantificadores pueden ser lineales o no lineales respectivamente. Para el caso de la selección del número de símbolos (escalones) a utilizar, generalmente se selecciona un número potencia de 2.

La critica principal al uso de este tipo de cuantificación es que no se hace ninguna suposición de la señal en análisis; sin embargo, es el "menos costoso" en materia de implementación. Mencionar que, aunque no es uno de los métodos mas elaborados, los Modelos Ocultos de Markov se fundamentan en el uso de "estados" los cuales, mediante estas técnicas, se pueden obtener directamente.

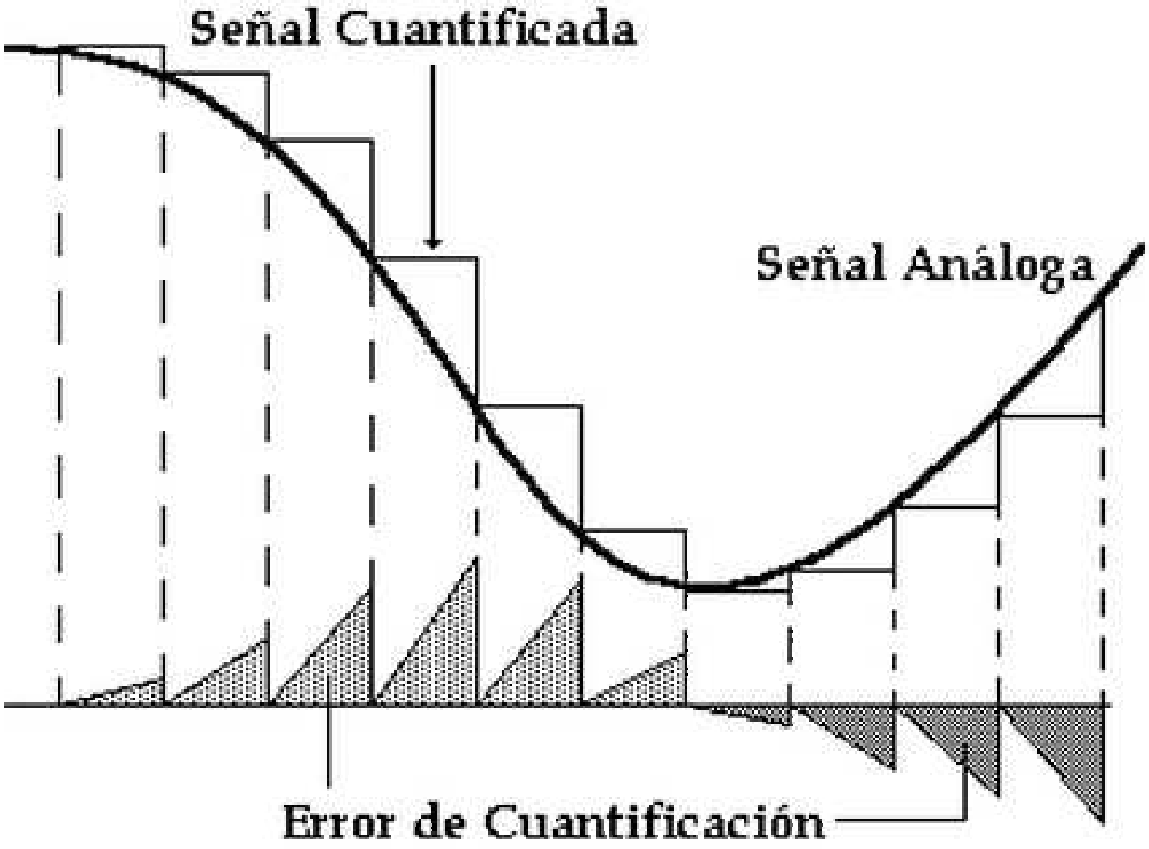


Figura 4.6 Descripción gráfica de la cuantificación.

4.1.4 Entrenamiento y prueba del sistema.

El proceso de reconocimiento, como se vio en las secciones anteriores, se fundamenta en dos procesos: entrenamiento y prueba.

Entrenamiento. Una vez cuantificada la señal de voz, se procede a su respectivo almacenamiento y se repite el proceso para cada repetición que conforma el "corpus de voces" (creación de la matriz de entrenamiento). El proceso de entrenamiento se realiza previo a la etapa de prueba con el fin de crear los modelos con los que se ha de realizar el reconocimiento.

Para llevar a cabo el proceso de entrenamiento por Modelos Ocultos de Markov es necesario obtener los siguientes parámetros:

1. Matriz de entrenamiento.

2. Matriz de probabilidad de transición.

3. Matriz de probabilidad de generación de símbolos.

4. Matriz de probabilidad de inicio.

Con estos valores, y mediante el uso del algoritmo de Baum-Welch (descrito en el capítulo anterior), se obtienen las matrices entrenadas; estas representan el Modelo de Markov respectivo. Al respecto, es necesario repetir este proceso para cada palabra del vocabulario con el fin de obtener un modelo para cada una de ellas.

Cada uno de los modelos se almacena, con su respectiva etiqueta, para utilizarlos en el proceso de prueba.

Prueba. Una vez obtenidos todos y cada uno de los modelos para las palabras, el proceso continúa con la etapa de prueba, es decir, la etapa propia para llevar a cabo el reconocimiento. La etapa de prueba inicia cuando se requiere identificar una nueva repetición (perteneciente al vocabulario que se puede reconocer) y su estructura es la siguiente:

- Adquisición,
- Pre-procesamiento,
- Extracción de características,
- Reconocimiento.

Estas etapas se realizan de la misma manera como se llevó a cabo en la etapa de entrenamiento excepto la etapa de reconocimiento. En esta etapa, junto con los parámetros que describen a cada modelo en específico, se calcula el valor de probabilidad de que la señal de voz a identificar (ahora expresada mediante una secuencia de símbolos) haya sido generada por alguno de los modelos entrenados. Esto se lleva a cabo mediante el uso del algoritmo hacia adelante (forward) descrito en el capítulo anterior. Una vez que se calcula esta probabilidad, se busca aquella que posea el menor valor y que, por lo tanto, representa la palabra de vocabulario que ha reconocido el sistema.

4.2 Métodos de reconocimiento.

En la sección anterior se describió el proceso que se llevará a cabo para la realización del sistema reconocedor, sin embargo, se proponen algunas variantes a este proceso con el fin de utilizar aquel o aquellos que sean mejores para realizar las pruebas de dicción.

Una vez que se han analizado algunas de las metodologías utilizadas para el reconocimiento de voz, como en [12], [13], [14], [15], [16], [17], [19] entre otros, se proponen varios algoritmos mediante los cuales se hace posible el diferenciar según la calidad de la dicción.

Para la investigación presente, se propone el uso de Modelos Ocultos de Markov Discretos junto con algunas modificaciones las cuales son:

- Modelos Ocultos de Markov con DTW,
- Modelos Ocultos de Markov con DTW aproximado por izquierda y derecha,
- Modelos Ocultos de Markov con relleno de palabras.

Como se describió anteriormente, los Modelos Ocultos de Markov son de gran utilidad cuando se trabaja con información estocástica como lo es en el caso de la voz. Dada la capacidad que posee dicha herramienta para trabajar con estados, es que se pretende buscar un método capaz de generar dichos estados para ser posteriormente tratados. Sin embargo, si las muestras de voz que producen los estados con los que trabajan los Modelos Ocultos varían en cuanto a su velocidad y duración es necesario utilizar una técnica que pueda disminuir el efecto que produce esta variación; es por ello que hemos considerado las tres variaciones descritas en los tres puntos anteriores.

4.2.1 Modelos Ocultos de Markov con DTW (MDTW).

La técnica de DTW, como se mencionó anteriormente, ha sido de gran utilidad para poder disminuir el efecto causado por las variaciones de velocidad en las emisiones de las palabras y por tanto una de las herramientas más utilizadas en el reconocimiento de voz en base a patrones.

Existen algunas investigaciones respecto de modelos híbridos entre DTW y HMM que han demostrado obtener buenos porcentajes en tareas de reconocimiento de voz [22].

Por esta razón es que se propone alinear temporalmente los eventos acústicos, es decir, cada repetición de las palabras y en base a este ajuste continuar con el proceso de reconocimiento hasta llegar a completar la etapa de prueba. La figura 4.7 muestra el diagrama de flujo correspondiente.

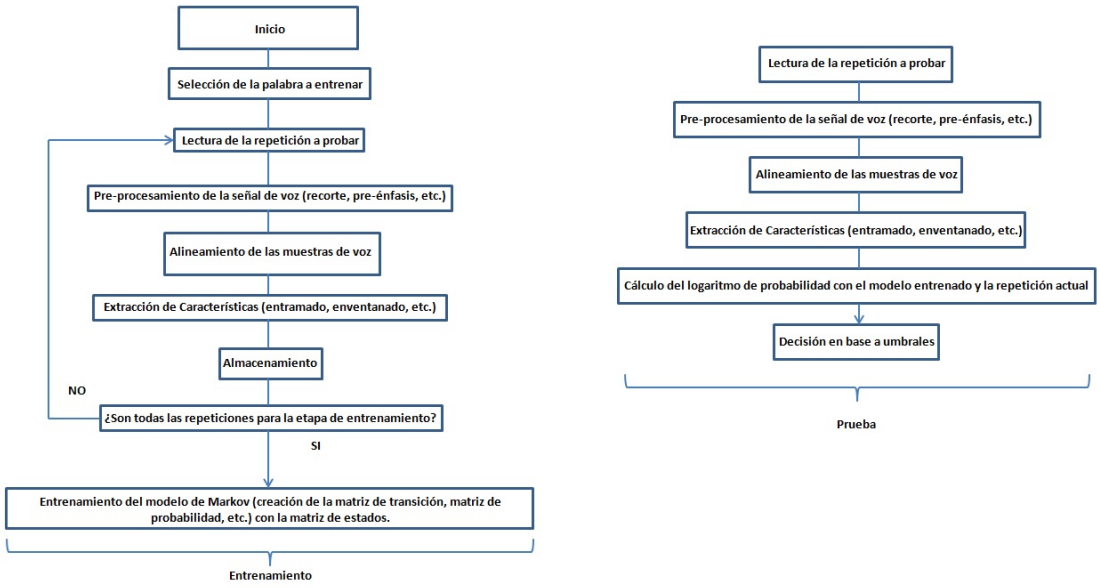


Figura 4.7 Diagrama de flujo para MDTW.

4.2.2 Modelos Ocultos de Markov con DTW aproximado por izquierda y derecha (MID).

Este método, al igual que en el caso anterior (MDTW), usa como fundamento la técnica de DTW junto con los Modelos Ocultos de Markov, sin embargo considera que la señal de voz puede ser dividida en dos unidades como el caso utilizado en palabras continuas [18] (este método es utilizado especialmente para el caso de similitudes de unidades fonéticas entre palabras). Sin embargo este método, pretende dividir a la señal de voz (repetición de la palabra) en dos unidades previamente alineadas por DTW, para luego crear con cada una un Modelo Oculto de Markov. El diagrama de flujo de ésta técnica se muestra en la figura 4.8.

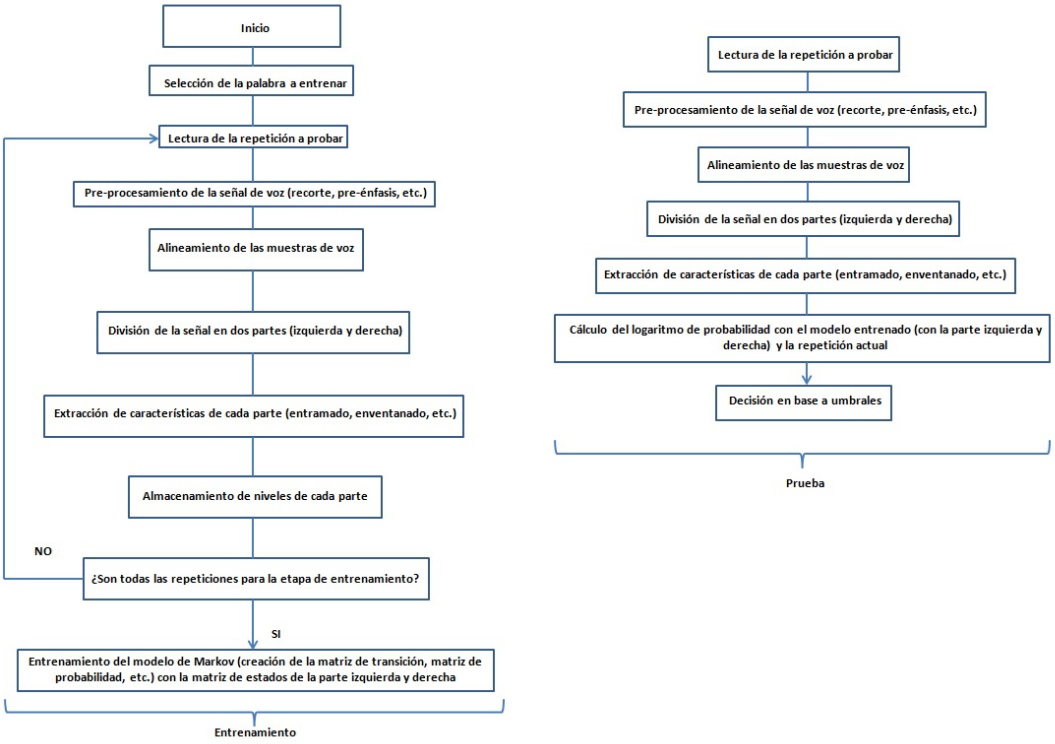


Figura 4.8 Diagrama de flujo para MID.

4.2.3 Modelos Ocultos de Markov con relleno de palabras (MRP).

Las técnicas anteriores usan como fundamento el alineamiento de muestras, sin embargo, dicho proceso modifica la información presente en cada una debido al costo por alineación. Una forma de poder crear la matriz de repeticiones es ajustando cada repetición al tamaño más grande de las muestras, es decir, rellenando la señal al tamaño más grande presente en las repeticiones de entrenamiento.

El relleno se hace en base al ajuste en la cuantificación, es decir, si una muestra de voz cuantificada es más pequeña que las muestras cuantificadas anteriormente, entonces se rellena dicha muestra al tamaño de la muestras anteriores. En el caso particular cuando las muestras cuantificadas de voz sean del mismo tamaño, entonces la muestra se almacena tal cual, sin modificar. En el último caso en el cual la muestra de voz es más grande que las muestras

almacenadas hasta ese momento, se rellenan todas las anteriores hasta alcanzar el tamaño de la muestra. El diagrama de flujo referente a esta aproximación se muestra en la figura 4.9.

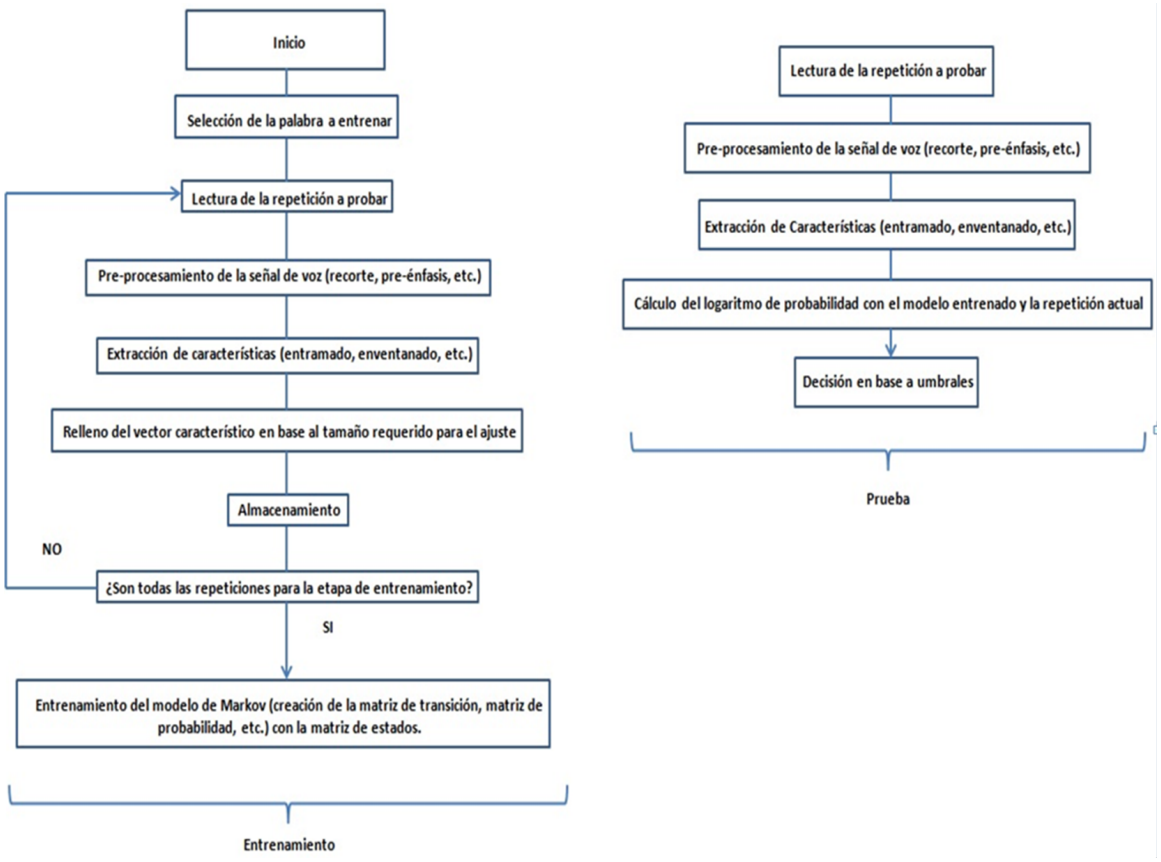


Figura 4.9 Diagrama de flujo para MRP.

Capítulo 5

Desarrollo de los sistemas de reconocimiento para dicción.

Como se describió en el capítulo anterior, para poder llevar a cabo el reconocimiento de voz, es necesario tratar matemáticamente la voz (haciendo uso de un sistema digital) con el fin de poder analizarla de la mejor manera posible y en base a dicho análisis, crear los patrones necesarios para llevar a cabo el reconocimiento.

En la presente investigación se optó por utilizar el software MATLAB en su versión 2011 dada su estructura para el desarrollo de programas con un amplio rango de funciones para la implementación en reconocimiento de voz.

Con respecto a la estructura anteriormente propuesta, se utiliza la siguiente configuración para poder, como se describirá en el capítulo siguiente, realizar las prueba de dicción del idioma español.

5.1 Adquisición.

5.1.1 Definición del corpus de voz

Con el fin de delimitar el corpus de voces, fue necesario definir las palabras sobre las cuales se fundamentarían las pruebas de dicción del idioma español. Para este fin, se optó por buscar la lista de palabras más difíciles de pronunciar del idioma en el que se enfocan las pruebas, es decir, del español. De la búsqueda realizada se encontró que, según una encuesta realizada por la empresa SpinVox [9], la lista está conformada entre otras por las siguientes palabras:

- Indecisión,
- Madrid,
- Helicóptero,
- Albóndiga,
- Ineptitud,
- Prejuicios.

5.1.2 Grabación y almacenamiento.

Una vez definidas las palabras de reconocimiento, fue necesario adquirir las muestras de voz (repeticiones) necesarias para definir el Corpus de Voces. Para este efecto, se buscó realizar las grabaciones de voz con las mejores condiciones de aislamiento de sonido posible. Al no disponer de una cámara anecóica, se optó por realizar las grabaciones mediante el uso de una cámara especial para video conferencias ubicada en las instalaciones del Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI), en donde se realizó una estancia corta con el apoyo de CONACyT y el IPN.

En base a las referencias consultadas, los hablantes que realizan las repeticiones para conformar el corpus de voz son aleatorios, es decir, no poseen una preparación especial. En base a esto es que, y de acuerdo a que cada modelo se crea en base a las características que poseen las palabras que conforman el entrenamiento, se optó por definir a los locutores para la conformación de las repeticiones como expertos en materia de comunicación. Esto se realiza dada la necesidad que el modelo creado posea características propias de dicción inherente a la información que lo conforma. La lista de hablantes está conformada por 4 estudiantes de la licenciatura en comunicaciones de la Universidad Autónoma de Baja California.

Según la cantidad de información (repeticiones que conforman el corpus de voz) que posee el modelo, es la forma en que se conforma el mismo. Desde esta perspectiva, se optó por definir el número de repeticiones en 10 repeticiones de cada palabra por hablante. A pesar de que no conforman una cantidad significativa de repeticiones (como se ha utilizado en otras

investigaciones) se optó por esta cantidad con el fin de que el modelo sea caracterizado con ciertas restricciones y así, además de lograr el reconocimiento de las palabras, poder delimitar la dicción de rechazo o aceptación (esto se explicará a detalle en el capítulo siguiente).

De acuerdo a las características anteriormente descritas, para la recopilación de las repeticiones de los locutores se utilizó un porta estudio de grabación **TASCAM DP-008** dadas sus características de portabilidad y buena calidad de grabación. En cuanto al micrófono utilizado, no existen consideraciones especiales dado que, la parte del proceso de adquisición, se realiza mediante el estudio de grabación portátil. Cada uno de los archivos fue obtenido con una frecuencia de 44100 Hz (calidad de CD), en formato .wav y con una calidad de audio de 24 bits.

Una vez recopiladas todas las repeticiones de cada palabra por cada uno de los locutores, se procedió a su almacenamiento (conformación final del corpus de voz). Al respecto se creó una estructura por medio del uso de nombres clave; esto con la finalidad de que su extracción para los futuros procesos pudiera realizarse de forma automática.

A continuación se define la configuración utilizada para las etapas de Pre-procesamiento y extracción de características la cual se aplica, en primera instancia, para cada una de las palabras que conforman el corpus de voz.

5.2 Pre-procesamiento.

Una vez definida la palabra a la que se le someterá al proceso de reconocimiento, y extraída de la dirección respectiva del almacenamiento, el pre-procesamiento se configuró como sigue:

5.2.1 Aplicación de filtro "pasa voz".

Con el fin de filtrar la señal que se obtiene, como etapa inicial, se configura un filtro capaz de dejar pasar sólo las frecuencias de voz de interés (ver figura 5.1).

El filtro utilizado fue un filtro tipo ventana de Hamming con una frecuencia de muestreo de 44100 Hz, frecuencias de corte de 100 a 8000 Hz y orden 800. Esta configuración responde a su buen desempeño en pruebas de domótica [18].

El utilizar un filtro tipo ventana, en lugar de un pasa-bajas, es que las frecuencias de interés por debajo de 100 Hz no resultan de interés por lo que, para evitar procesamiento extra, se optó por aplicar un rechazo por debajo de la misma.

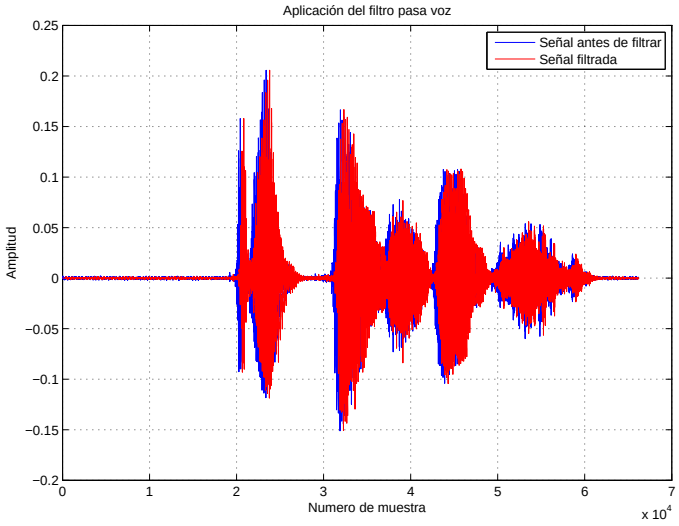


Figura 5.1 Filtrado pasa voz de la palabra “prejuicios”.

5.2.2 Recorte de la señal de voz (detector de voz).

Como se puede observar en la figura 5.1, la señal de voz posee partes de la señal donde existen silencios prolongados. Es claro gracias a dicha figura que existen silencios antes, durante, y al final de la pronunciación. En el reconocimiento de palabras aisladas (como se vio en el capítulo anterior) es necesario eliminar de la señal los silencios antes y después de la pronunciación con el cuidado de no confundir a los intermedios.

El proceso que se llevó a cabo consistió en, una vez normalizada la señal respecto del valor máximo presente en la señal, se calculó la energía promedio. Una vez obtenido el valor mencionado, se eligió un umbral de decisión de 0.01 el cual delimita que muestras han de tomarse como de interés (sonidos vocalizados) y cuales no (silencios). Con estos parámetros se divide la señal en tramas de información (400 ms) y se extraen aquellas que cumplen con la condición de superar el valor de umbral. Luego de realizar este proceso, se procede a detectar el inicio y fin de la palabra mediante un contador de ventanas de energía subsecuentes cuidando

el aspecto de confundir silencios inicial y final con silencios entre palabras. Finalmente, se realiza el recorte correspondiente según lo obtenido en el proceso anterior. El resultado final de recorte de la señal de voz se muestra en la figura 5.2.

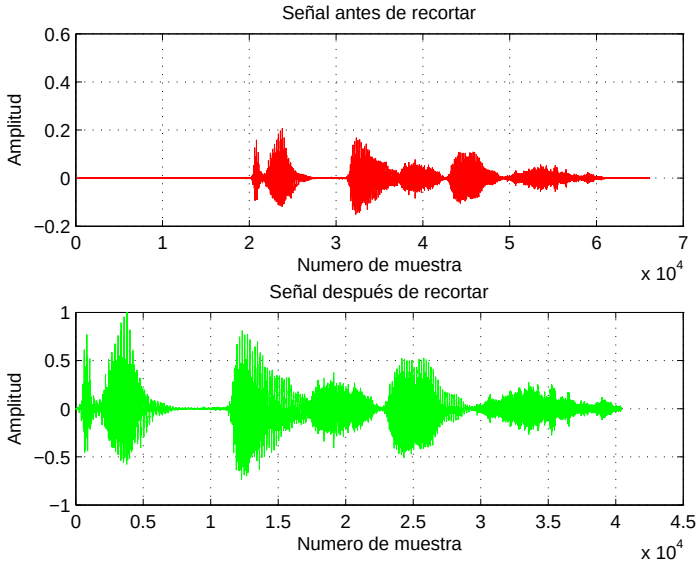


Figura 5.2 Recorte de voz aplicado a la señal de voz.

5.2.3 Pre-énfasis.

Como parte del proceso, y dado que es necesario ecualizar la señal de voz, el proceso de pre-énfasis es un elemento indispensable en la mayoría de los sistemas de reconocimiento. Un aspecto interesante es que algunos de estos sistemas, proponen como primera fase el pre-énfasis sin tomar en cuenta ninguno de los procesos anteriores.

La ecuación recursiva utilizada para llevar a cabo el pre-énfasis se expresa en (5.1). Mediante esta ecuación se hace posible el aumentar la magnitud de las muestras enfatizando la variación entre puntos adyacentes [15] (ver figura 5.3)

$$y(n) = x(n) - \tilde{a}x(n-1), 0.9 \leq \tilde{a} \leq 1.0. \tag{5.1}$$

donde $y(n)$ es la señal filtrada, $x(n)$ es la muestra actual y $x(n-1)$ es la muestra anterior(de aquí que el proceso comienza con la segunda muestra).

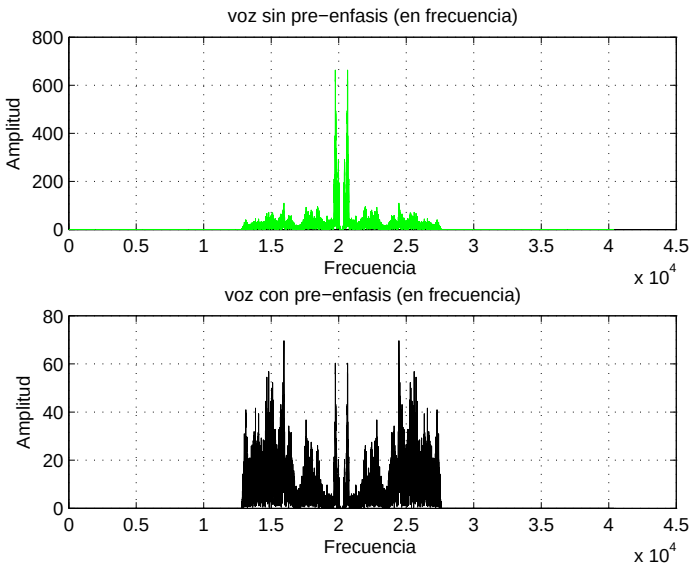


Figura 5.3 Espectrograma original y obtenido después de la aplicación de pre-énfasis.

5.2.4 Normalización.

Como se mencionó en el capítulo anterior, la normalización se utiliza para ajustar la amplitud de las muestras de voz y como parte de un ajuste (conservación de la potencia) previo al proceso de segmentación y ventaneo. El proceso consiste en calcular el valor máximo que posee la señal de voz y en base a dicho valor, dividir cada una de las muestras de voz entre dicho valor.

5.2.5 Segmentación y Ventaneo.

La señal de voz es una señal de naturaleza aleatoria y por tanto para su análisis es necesario aplicar un procesamiento que asegure su estacionariedad.

El método de segmentación se encarga de "dividir" la señal en tramas de análisis. Se decidió que esta segmentación se realizara cada 30ms con el fin de asegurar esta condición. Con respecto al traslape entre tramas se decidió que fuera del 50% dado que este valor es el más utilizado y ha demostrado dar buenos resultados.

Una vez que se obtiene una trama, el proceso continua aplicando el enventanado correspondiente con la finalidad de ajustar la trama de análisis. Para el caso presente se optó por

utilizar una ventana Hamming del mismo ancho que el de la trama en análisis (ver figura 5.4). El proceso anterior se puede observar gráficamente en la figura 5.5.

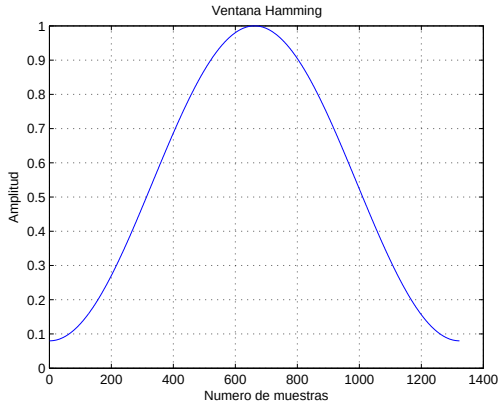


Figura 5.4 Ventana Hamming utilizada.

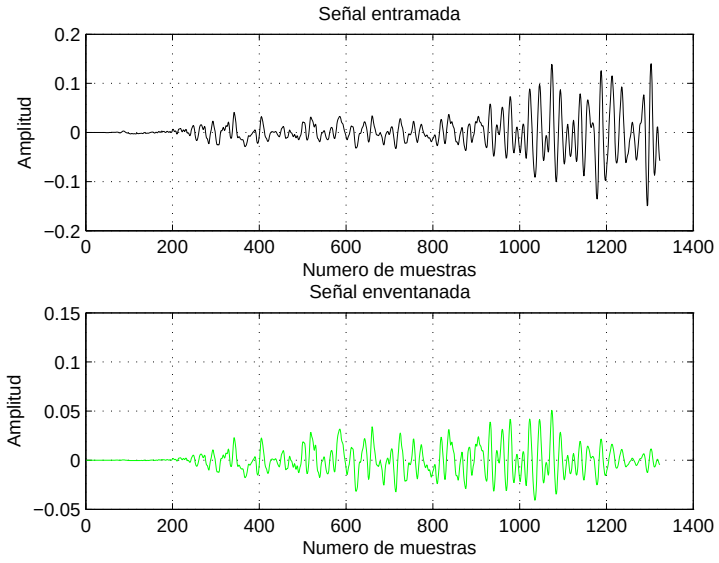


Figura 5.5 Aplicación de segmentación y ventaneo.

5.3 Extracción de características.

Para extraer la información característica de cada repetición de voz, es necesario el análisis de cada una de las tramas que se obtuvieron del proceso anterior. Para este fin se optó por realizar el análisis de la energía contenida en cada trama.

Para llevar a cabo este proceso, al igual que el proceso de "detección de voz", la energía por tramas se obtiene mediante la ecuación:

$$E = \frac{1}{N} \sum_{n=1}^{N} |X[n]|^2, \tag{5.2}$$

donde E es la energía total por trama calculada, N es el número total de muestras según el tamaño definido para las tramas y $X[n]$ es la muestra procesada.

El proceso de extracción de la energía, como se ha mencionado, debe aplicarse a cada una de las tramas con el fin de obtener un vector característico con los niveles de energía de la señal de voz (figura 5.6). La obtención de estos vectores de energía permite distinguir entre características similares en el proceso de reconocimiento.

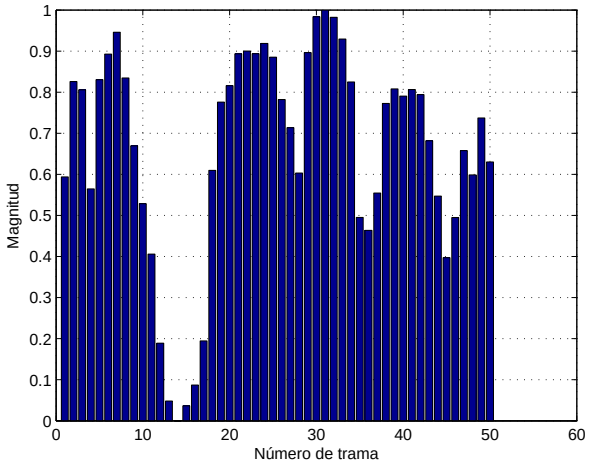


Figura 5.6 Extracción de energía por trama (vector de energía normalizado).

Una vez que se ha creado el vector de energía característico de la repetición de voz analizada, es necesario aplicar un cuantificador escalar con el fin de clasificar los niveles de energía en un valor discreto.

Para poder aplicar un cuantificador escalar es necesario, en primera instancia, conocer el valor máximo que posee la señal de voz así como el valor mínimo (para este caso siempre es cero dada la normalización previa). En cuanto al número de escalones utilizados para la cuantificación se hicieron pruebas con diferentes valores (14, 22, etc.), sin embargo, el valor utilizado finalmente se fijó en 30.

Para determinar el rango se utilizó la siguiente ecuación:

$$Rango = \frac{M - n}{E}, \tag{5.3}$$

donde M es el valor máximo que posee la señal de voz, n el valor mínimo y E el número de escalones. De la ecuación anterior, los valores mínimo y número de escalones permanecen fijos, mientras que el valor máximo (y por tanto el rango) varia según la señal de voz analizada.

Con estos parámetros, el proceso de cuantificación analiza cada uno de los niveles de energía y les asigna un valor discreto de los 30 posibles (ver figura 5.7) de acuerdo con la siguiente condición:

$$(Etrama(i,1) \geq (Rango * (j-1)) \& (Etrama(i,1) \leq (Rango) * j), \tag{5.4}$$

donde $Etrama$ es la energía por trama, i es el número de ventana analizada y j es el número de escalón correspondiente.

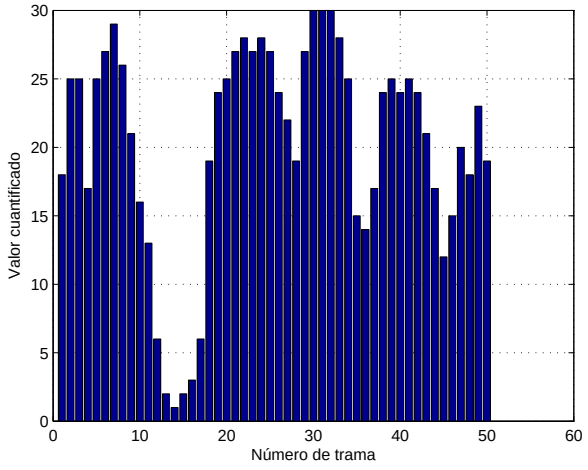


Figura 5.7 Cuantificación escalar correspondiente al vector de energía de la figura 5.6.

La ecuación anterior ser repite de forma cíclica hasta analizar todos y cada uno de los niveles de los elementos del vector de energía. Una vez finalizada la cuantificación del total de muestras de voz, se crea una matriz con los elementos resultantes tal como se muestra en la figura 5.8.

	1	2	3	4	5	6	7	8	9
1	18	25	25	17	25	27	29	26	21
2	15	13	10	23	24	22	20	19	13
3	20	14	23	29	30	29	27	24	18
4	15	16	11	20	22	22	20	18	11
5	14	15	11	9	16	24	25	22	17
6	11	17	15	18	25	25	23	21	16
7	15	14	6	19	29	29	25	20	12
8	15	21	18	14	25	29	30	28	23
9	18	25	19	8	23	30	29	27	21
10	29	30	28	28	28	28	25	23	19
11	28	20	23	27	27	28	27	25	17
12	28	26	21	26	28	28	27	25	20
13	30	25	23	30	28	28	27	21	17

Figura 5.8 Ejemplo de matriz de niveles de energía del corpus de voz correspondiente.

5.4 Entrenamiento.

Como se explicó en el capítulo anterior, para poder llevar a cabo el entrenamiento de los Modelos Ocultos de Markov se utilizó el algoritmo de **Baum-Welch**. Para ello es necesario definir algunas matrices ó parámetros iniciales (Modelo Markoviano Semilla) a partir de los cuales se creará el modelo para la palabra en entrenamiento.

El modelo Markoviano, de acuerdo a la aplicación en reconocimiento de voz, es un modelo tipo izquierda-derecha lo cual emula el comportamiento natural de la voz (no posee retrocesos entre estados). El número de símbolos utilizados, como se definió en la cuantificación, es de 30 símbolos diferentes; mientras que el número de observaciones está regido según el método utilizado para el reconocimiento. Este último parámetro es de vital importancia dado que los algoritmos propuestos para la realización de esta investigación se basan en modificaciones sobre este punto.

En lo referente a la matriz de transición de estados (ver figura 5.9) y la matriz de generación de símbolos (ver ejemplo de la figura 5.10) se programaron para no favorecer la transición entre estados o la generación de un símbolo, es decir, equiprobable. Por otro lado, el vector de estado inicial se programa para iniciar con el primer estado.

En cuanto a las dimensiones de cada una de las matrices, se obtienen de acuerdo a lo siguiente:

- Las dimensiones de la matriz de transición de estados o matriz A, es decir el número de renglones y columnas, es igual al número de observaciones, esto es al número de ventanas de análisis (elementos del vector de energía). Los valores no nulos corresponden a las transiciones posibles y se calculan de acuerdo al tipo de modelo utilizado.

- El número de renglones de la matriz de generación de símbolos o matriz B, es igual al número de ventanas de análisis; mientras que para el número de columnas es igual al número de niveles de cuantificación. El valor de cada elemento (igual para toda la matriz) corresponde a :

$$P = \frac{1}{E}, \tag{5.5}$$

  donde E es el número de escalones (niveles de cuantificación).

- El vector columna de estado inicial posee el mismo número de elementos que el número de ventanas de análisis. El valor de 1 corresponde a la posición del estado inicial.

0.3	0.3	0.3	0	0	0	0	0	0	0	0
0	0.3	0.3	0.3	0	0	0	0	0	0	0
0	0	0.3	0.3	0.3	0	0	0	0	0	0
0	0	0	0.3	0.3	0.3	0	0	0	0	0
0	0	0	0	0.3	0.3	0.3	0	0	0	0
0	0	0	0	0	0.3	0.3	0.3	0	0	0
0	0	0	0	0	0	0.3	0.3	0.3	0	0
0	0	0	0	0	0	0	0.3	0.3	0.3	0.3
0	0	0	0	0	0	0	0	0.3	0.3	0.3
0	0	0	0	0	0	0	0	0	0.5	0.5
0	0	0	0	0	0	0	0	0	0	1

Figura 5.9 Ejemplo de matriz cuadrada de transición de estados izquierda-derecha (12 observaciones).

0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2

Figura 5.10 Ejemplo de matriz generación de símbolos (12 observaciones y 5 símbolos).

Una vez obtenida la matriz de datos (o matriz de observaciones), la matriz de transición de estados, la matriz de generación de símbolos, y el vector columna de estado inicial, se procede al proceso de entrenamiento. Para este efecto, se utilizó una librería de código de libre acceso especial para la aplicación en MATLAB (ver referencia [27]). Esta librería utiliza el algoritmo de **Baum-Welch** para crear el modelo de Markov de la palabra en entrenamiento. Uno de los parámetros que es necesario configurar para utilizar esta función, además de la asignación de las matrices antes mencionadas, es el llamado: *número de iteraciones*. Este valor es de importancia dado que determina el tiempo de cálculo que le llevará al proceso para crear el modelo con los datos ingresados. Para el caso presente, el valor se fijó, en base a algunas prueba realizadas en 50 iteraciones (para una explicación más detallada analizar a profundidad la referencia [27]).

Con estos valores, al finalizar el entrenamiento, el algoritmo produce las siguientes matrices:

- Matriz entrenada de transición de estados.
- Matriz entrenada de generación de símbolos.
- Matriz de estado inicial entrenado.
- Logaritmo de probabilidad acumulada.

Este último parámetro es el resultado de sumar los valores de probabilidad de cada iteración (el logaritmo escala los valores de probabilidad dado que suelen ser muy pequeños). El cambio de este valor está ligado al número de iteraciones, es decir, su valor va disminuyendo con cada iteración hasta llegar a la estabilidad (de aquí la importancia mencionada anteriormente sobre la elección del número de iteraciones). Estos parámetros permiten crear la curva de aprendizaje que es el número de iteraciones hasta llegar a la estabilidad del valor de probabilidad conjunta (como se puede ver en la figura 5.11).

Una vez entrenado el modelo (y el de las demás palabras del vocabulario) los valores de las matrices obtenidas se almacenan (con su respectiva etiqueta) para su uso en la etapa de prueba. Vale la pena hacer notar que no es necesario repetir este proceso cada vez que se

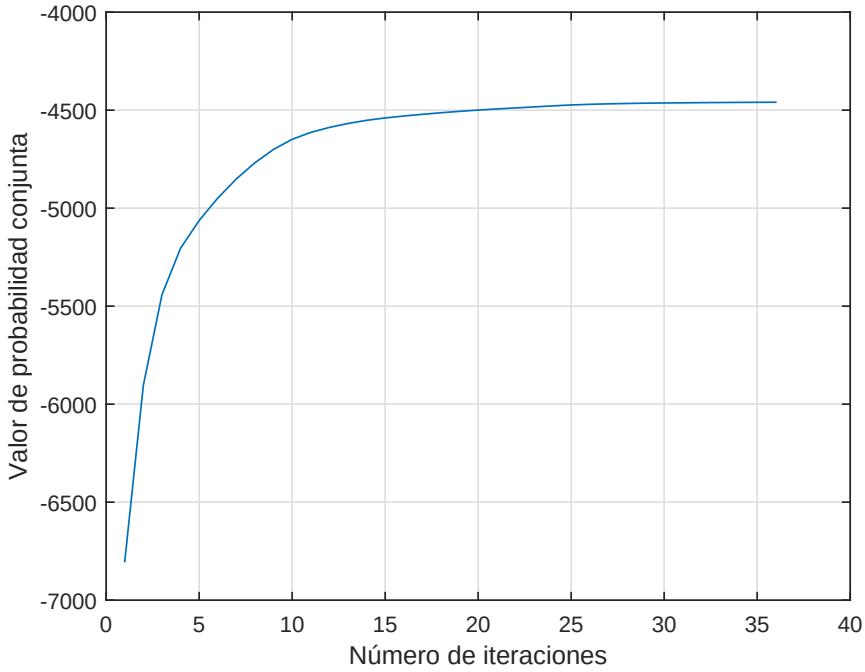


Figura 5.11 Ejemplo de Curva de aprendizaje con 50 iteraciones.

realice el reconocimiento sino que, por el contrario, una vez entrenado cada modelo, tanto las repeticiones del corpus de voces como los demás parámetros que no caracterizan al modelo entrenado, no se necesitan conservar para la etapa de reconocimiento.

5.5 Prueba.

La etapa de prueba corresponde al reconocimiento de palabras una vez que ya se ha llevado a cabo la etapa de entrenamiento. Este proceso consiste en que, una vez que se ingresa una nueva repetición a reconocer, de los modelos entrenados correspondientes al vocabulario, cual es la probabilidad de que la repetición a reconocer haya sido generada por un modelo en específico; y en base a ello reconocer la palabra.

Las etapas para llevar a cabo el reconocimiento (prueba), son similares a las que se llevan a cabo para la etapa de entrenamiento. La diferencia radica en que el proceso solo se hace respecto de la nueva palabra a identificar. Es decir:

1. Adquisición,

2. Pre-procesamiento,

3. Extracción de características,

Una vez que se han extraído las características de la repetición de voz (vector de características energético) se procede a realizar la etapa de prueba. Para llevar a cabo este proceso se utiliza, al igual que para la etapa de entrenamiento, la herramienta desarrollada por Kevin Murphy (ver referencia [27]) pero utilizando el algoritmo forward. Para utilizar esta herramienta los parámetros de entrada son las matrices resultantes de la etapa de entrenamiento (matriz de transición de estados, matriz de generación de símbolos y la matriz de estado inicial) y el vector característico de la repetición a identificar. Con estos elementos, el algoritmo entrega finalmente un parámetro denominado como *logaritmo de probabilidad*. El logaritmo de probabilidad indica cuál es la probabilidad de que la palabra en etapa de prueba haya sido generada por el modelo definido por las matrices ingresadas en el algoritmo. Este valor de probabilidad varía según las matrices de entrenamiento ingresadas y es en base a ello que se puede identificar a la palabra ingresada (según el algoritmo, aquel que posea el menor valor de probabilidad).

Los sistemas de reconocimiento convencionales realizan esta etapa de entrenamiento obteniendo el logaritmo de probabilidad respecto de cada modelo e identificando al de menor valor, sin embargo, para el caso de la presente investigación, la diferencia radical es que este proceso de reconocimiento no necesita obtener todos estos valores de probabilidad sino que solo se obtiene respecto de la palabra elegida para llevar a cabo el entrenamiento. En el capítulo siguiente se describen las pruebas pertinentes para llegar a esta definición dentro del algoritmo desarrollado; sin embargo, se menciona desde este punto para aclarar la diferencia con otros sistemas.

5.6 Variación con los métodos propuestos.

Las etapas descritas en este capítulo hasta este momento conforman un bosquejo general del utilizado para realizar las pruebas de dicción sin embargo, como se mencionó dentro del mismo, se proponen algunas variaciones con la finalidad de adecuarlo para la realización de pruebas de dicción en el idioma español.

5.6.1 Modelos Ocultos de Markov con DTW (MDTW).

Como se mencionó en el capítulo anterior, se propone el uso de un método de alineación temporal de las repeticiones de voz para el proceso de entrenamiento y prueba mediante la técnica a la cual se le denominó para el uso en esta tesis como: **Modelos Ocultos de Markov con DTW (MDTW)**.

El primer paso para la implementación de este algoritmo es definir la muestra de voz respecto de la cual se hará la alineación. Este proceso es sumamente complejo dado que la muestra de voz respecto de la cual se alinean las muestras, modificará el corpus de voz y por tanto la etapa de reconocimiento. Para el caso presente, la elección de esta muestra de voz se seleccionó de acuerdo a la cual, subjetivamente, posee la mejor dicción desde la perspectiva de la audición.

Supongamos que se tienen dos señales de voz como las que se muestran en la figura 5.12.

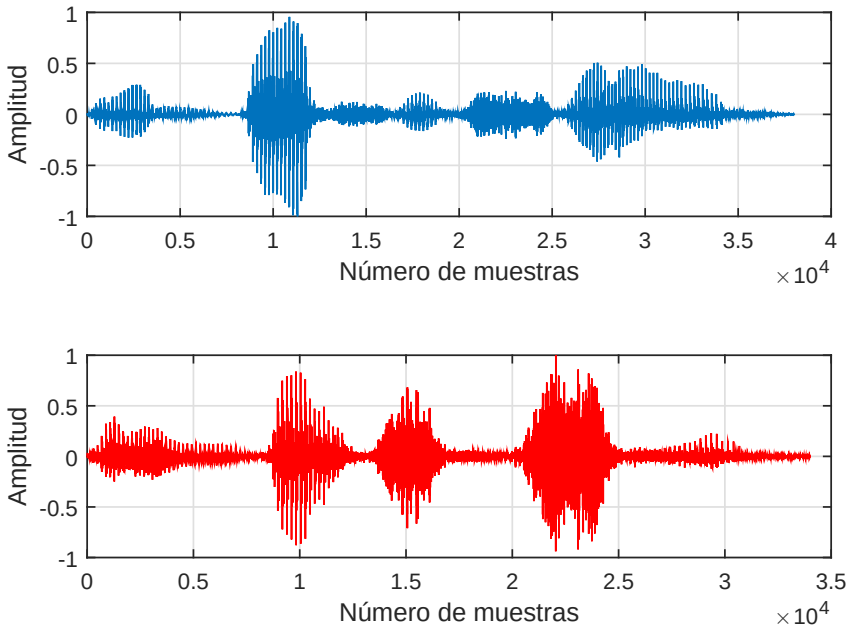


Figura 5.12 Ejemplo de muestras de voz pronunciando la palabra “indesición”.

Es de notar en la figura anterior, que las semejanzas entre ambas (temporalmente) no parecen tener mayor relación que algunas distribuciones energéticas, sin embargo, ambas corresponden a la pronunciación de la misma palabra (luego del pre-procesamiento correspondiente) pero de diferentes hablantes. La idea es entonces, aplicar el alineamiento dinámico a ambas señales con la finalidad de lograr el empate necesario para que posean la misma cantidad de muestras y así almacenarlas dentro de una misma matriz. El problema principal radica, como se mencionó ya anteriormente, en que este "ajuste" modifica la información y por tanto existe un "costo por alineación".

El proceso para llevar a cabo la alineación a grandes rasgos se divide, en los siguientes procesos (extraídos del código de libre acceso de la referencia [26]):

1. Identificar la señal de voz esclavo y guía.
2. Calcular el espectrograma de ambas señales mediante el uso de la Transformada de Fourier en Tiempo Corto (STFT).
3. Construir la matriz de distancias locales entre ambos espectrogramas.
4. Utilizar programación dinámica con el fin de calcular el emparejamiento de menor costo.
5. Obtener la versión alineada de la muestra de voz definida como esclavo.

El resultado de la aplicación de este método se muestra en la figura 5.13.

Luego del proceso de alineamiento, se continua con el proceso de entrenamiento. Luego de concluir este proceso, en la etapa de prueba, también es necesario aplicar una alineación respecto de la repetición de voz con la cual se hizo el alineamiento y continuar con el proceso estándar (ver algoritmo 1, el cual se describe también en la figura 4.7). De aquí que para esta etapa, además de las matrices entrenadas para cada palabra del vocabulario, se necesita conservar la señal de voz respecto de la cual se hace el alineamiento.

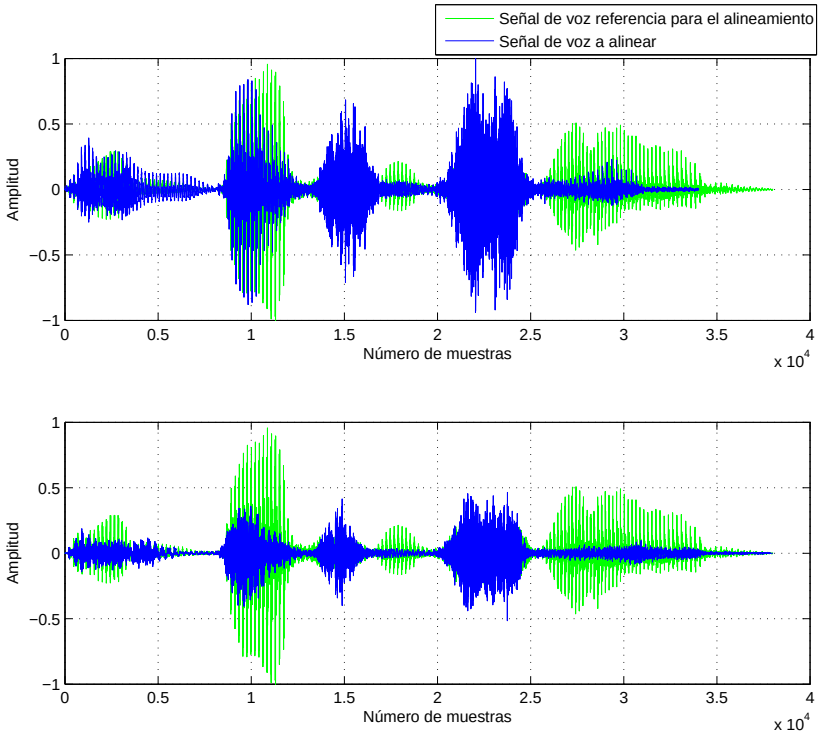


Figura 5.13 Ejemplo de Alineamiento de muestras de voz pronunciando la palabra "indesición".

5.6.2 Modelos Ocultos de Markov con DTW aproximado por izquierda y derecha (MID).

En base al algoritmo anterior (algoritmo 1) se propone un segundo método en el cual la señal de voz se divide en dos unidades de información. Esté método, para su uso en esta investigación, se denomina como: **Modelos Ocultos de Markov con DTW aproximado por izquierda y derecha (MID)**.

Al respecto, como primera parte del proceso, al igual que en el caso anterior, se elige una muestra sobre la cual se pretende hacer el alineamiento y se le aplican todas las fases del pre-procesamiento (hasta antes de la segmentación y ventaneo). Después de esto, el proceso continua normalmente hasta la etapa del alineamiento. Después, en base al tamaño de la voz muestra, la señal alineada se divide en dos partes: izquierda y derecha (ver figura 5.14). Es así, como el proceso de extracción de características, almacenamiento y entrenamiento se aplica para cada parte por separado. En cuanto a la etapa de reconocimiento, se realiza el alineamiento

Algoritmo 1 Pseudocódigo MDTW

Entrada: Muestras de voces a entrenar.

Salida: Calificación de la dicción detectada.

1: seleccionar la palabra a entrenar.
2: **repetir**
3: leer la repetición a probar.
4: aplicar pre-procesamiento de la señal de voz (recorte, pre-énfasis, etc.).
5: alinear las muestras de voz.
6: extraer características (entramado, enventanado y cuantificación en niveles de energía).
7: almacenar.
8: **hasta que** se entrenen todas las muestras de voz previamente almacenadas.
9: entrenar el Modelo de Markov (creación de la matriz de transición, matriz de probabilidad, etc.) con la matriz de estados.
10: **si** se prueba una nueva repetición **entonces**
11: leer la repetición a probar.
12: aplicar pre-procesamiento de la señal de voz (recorte, pre-énfasis, etc.).
13: alinear las muestras de voz.
14: extraer características (entramado, enventanado y cuantificación en niveles de energía).
15: calcular el logaritmo de la probabilidad con el modelo entrenado y la repetición actual.
16: **fin si**

tal como se propuso en la sección anterior y se divide la señal en dos partes igualmente. Por último, se calcula el valor de logaritmo de probabilidad respectiva y, en base a decisiones respecto del valor calculado, se identifica la palabra en prueba. El algoritmo 2 muestra el proceso que se lleva a cabo para la implementación de esta técnica que también se presentó en la figura 4.8.

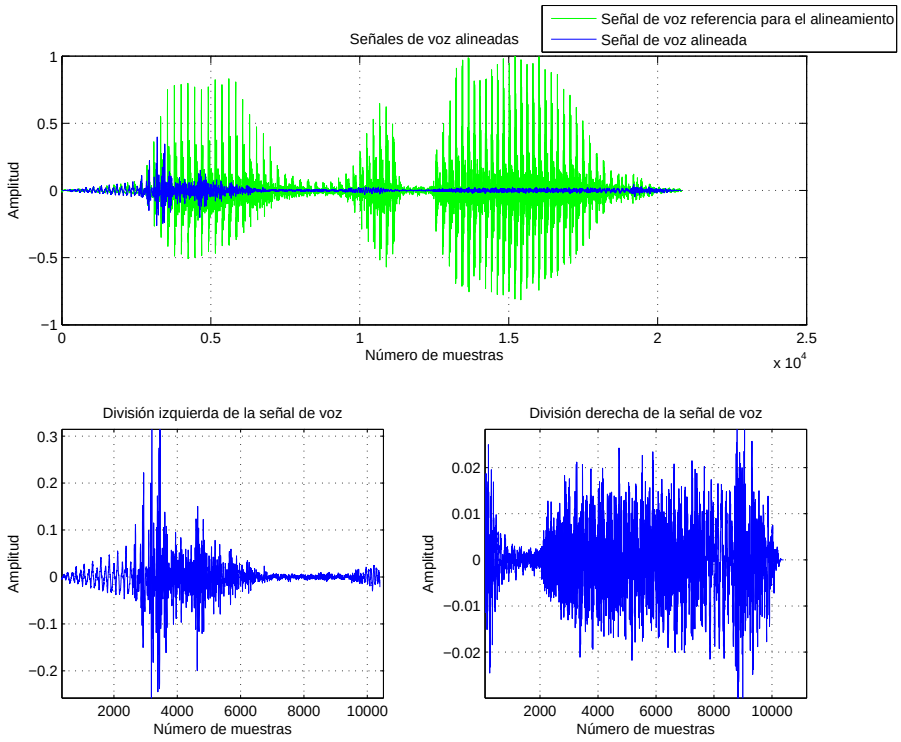


Figura 5.14 Ejemplo de alineamiento y división de la señal de voz pronunciando la palabra "madrid".

5.6.3 Modelos Ocultos de Markov con relleno de palabras (MRP).

El último método propuesto corresponde a: **Modelos Ocultos de Markov con relleno de palabras (MRP)**. Este método, a diferencia de los anteriores, solo hace uso del ajuste en el almacenamiento y no propiamente en el proceso previo a la extracción de información.

El proceso para aplicar el relleno de palabras comienza, en la etapa de entrenamiento, una vez que se obtiene el vector de información (en este caso de niveles de energía). Una vez que se tiene el vector de energía es necesario calcular el tamaño del vector actual. Si se trata del primer vector de información analizado, entonces se almacena sin modificaciones y se pasa a la extracción de la siguiente muestra del corpus de voces; si se trata de cualquier otro vector, el proceso de relleno se corresponde de acuerdo a si este nuevo vector es más grande, más pequeño, o del mismo tamaño (ver figura 5.15) que los vectores almacenados (el relleno, propiamente dicho, consiste en agregar tantos "1′s" como sea necesario).

Algoritmo 2 Pseudocódigo MID

Entrada: Muestras de voces a entrenar.

Salida: Calificación de la dicción detectada.

1: seleccionar la palabra a entrenar.
2: **repetir**
3: leer la repetición a probar.
4: aplicar pre-procesamiento de la señal de voz (recorte, pre-énfasis, etc.).
5: alinear las muestras de voz.
6: dividir la señal en dos partes (izquierda y derecha).
7: extraer características (entramado, enventanado y cuantificación en niveles de energía).
8: almacenar.
9: **hasta que** se entrenen todas las muestras de voz previamente almacenadas.
10: entrenar el Modelo de Markov (creación de la matriz de transición, matriz de probabilidad, etc.) con la matriz de estados de la parte izquierda y derecha.
11: **si** se prueba una nueva repetición **entonces**
12: leer la repetición a probar.
13: aplicar pre-procesamiento de la señal de voz (recorte, pre-énfasis, etc.).
14: alinear las muestras de voz.
15: dividir la señal en dos partes (izquierda y derecha).
16: extraer características de cada parte (entramado, enventanado y cuantificación en niveles de energía).
17: calcular el logaritmo de probabilidad con el modelo entrenado con la parte izquierda y derecha y la repetición actual.
18: **fin si**

La ventaja de este método es que no se modifica la información acústica correspondiente (como en los métodos anteriores); sin embargo, el relleno puede implicar algunos efectos negativos como el tiempo de cómputo requerido para rellenar una matriz creada cuando un vector

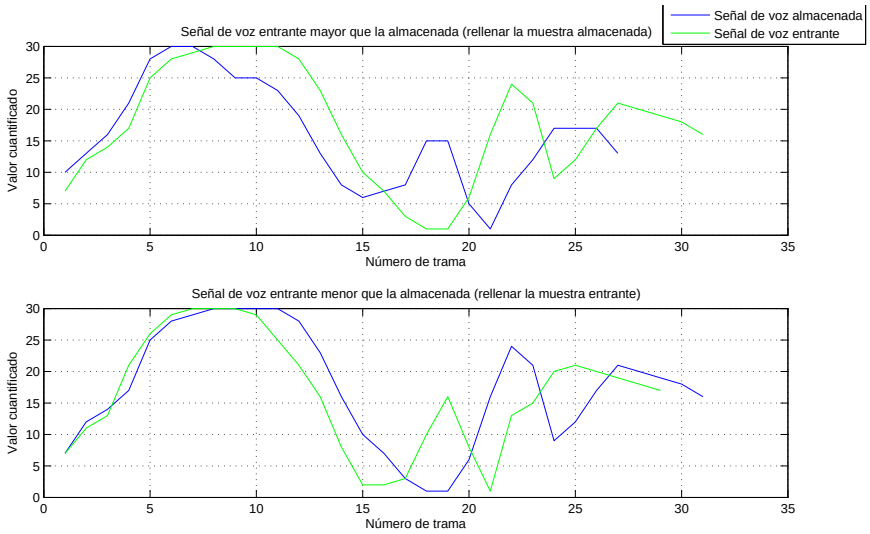


Figura 5.15 Ejemplo de análisis para la aplicación del relleno de palabras.

entrante es mayor que el tamaño común de los vectores almacenados hasta ese momento. El algoritmo 3 describe el proceso para aplicar este método.

La lista de programas implementados y su respectivo algoritmo se encuentra en el apéndice B. En el siguiente capítulo se describen tanto las pruebas realizadas con todos y cada uno de los métodos, así como la implementación en términos de una interfaz de usuario final para la realización de pruebas de dicción.

Algoritmo 3 Pseudocódigo MRP

Entrada: Muestras de voces a entrenar.

Salida: Calificación de la dicción detectada.

1: seleccionar la palabra a entrenar.
2: **repetir**
3: leer la repetición a probar.
4: aplicar pre-procesamiento de la señal de voz (recorte, pre-énfasis, etc.).
5: extraer características (entramado, enventanado y cuantificación en niveles de energía).
6: rellenar el vector característico en base al tamaño requerido para el ajuste.
7: almacenar.
8: **hasta que** se entrenen todas las muestras de voz previamente almacenadas.
9: entrenar el Modelo de Markov (creación de la matriz de transición, matriz de probabilidad, etc.) con la matriz de estados.
10: **si** se prueba una nueva repetición **entonces**
11: leer la repetición a probar.
12: aplicar pre-procesamiento de la señal de voz (recorte, pre-énfasis, etc.).
13: extraer características (entramado, enventanado y cuantificación en niveles de energía).
14: calcular el logaritmo de probabilidad con el modelo entrenado y la repetición actual.
15: **fin si**

Capítulo 6

Pruebas y resultados.

El objetivo de nuestro trabajo de investigación es el de desarrollar pruebas de dicción en el idioma español, es por ello que en los capítulos anteriores se propuso una metodología para el desarrollo de la investigación y 3 métodos para llevar a cabo el reconocimiento de palabras.

Para poder enfocar la investigación desarrollada en la identificación de la dicción, es necesario tomar en cuenta tres características fundamentales: **Pronunciación**, **Acentuación** y **Velocidad**. Cada uno de estos parámetros quedan justificados desde la perspectiva en que la pronunciación puede ser vista como la distribución de la señal, la acentuación como los niveles máximos que posee la señal y la velocidad como la distribución en tiempo de cada uno de estos niveles.

Todos estos parámetros, vistos desde una perspectiva general en un modelo, son contemplados en la creación de un modelo específico pues se toma en cuenta la información contenida en cada una de las palabras que conforman el corpus de voces y en base a ellas se crea el modelo. Si este modelo es creado con información *especial* entonces el modelo se crea con *parámetros especiales* (estos parámetros especiales son los que se acaban de describir en el párrafo anterior).

Una vez justificado el hecho de que mediante un modelo se puede caracterizar mejor una palabra (en comparación con otros métodos de reconocimiento) y por ende este será utilizado para la realización de pruebas de dicción, las pruebas de dicción consisten entonces en, de entre la variabilidad de métodos propuestos, identificar aquel o aquellos métodos que posean las mejores características para calificar la dicción.

Como se describió en el capítulo anterior, el corpus de voz está conformado por repeticiones de las palabras del vocabulario de 4 hablantes distintos. Para cada uno de los métodos probados se utilizaron 40 repeticiones para la etapa de prueba y 10 repeticiones para la etapa de entrenamiento.

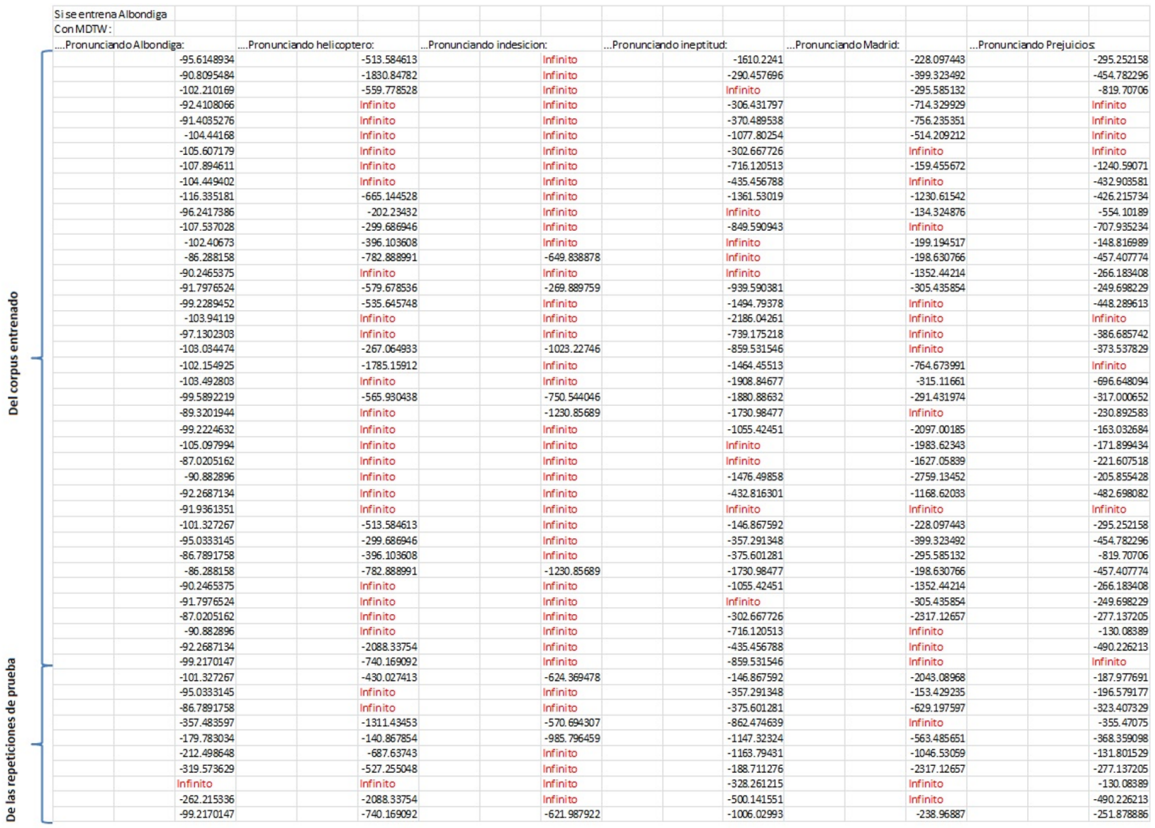
La primer prueba consistió en, para cada una de las palabras y con cada uno de los métodos, calcular el logaritmo de probabilidad acumulada (loglik) y, en base a estos resultados, definir la forma de calificar la dicción.

Para entender mejor este proceso, supongamos que la palabra en entrenamiento es la palabra "albóndiga" y se realiza el proceso de reconocimiento por medio del método **MDTW**. Es necesario asegurar que la palabra que se reconoce, para el modelo específico entrenado, sea sólo esa. Es por ello, que se calculó el logaritmo de probabilidad acumulada para los casos en que, entrenado el modelo de la palabra seleccionada, se intentará reconocer otra del corpus de voz. Los resultados obtenidos con estas características serían los encontrados en la figura 6.1.

En la figura 6.1 , los primeros 40 resultados del logaritmo de probabilidad corresponden a las pruebas realizadas con las mismas repeticiones con las que se realizó el entrenamiento del modelo; las restantes son repeticiones diferentes a las utilizadas en el entrenamiento. El valor del logaritmo de probabilidad acumulada es entregado como un valor negativo que entre más cercano sea al "cero", más probabilidad tiene de pertenecer al modelo entrenado.

El objetivo de utilizar el reconocimiento de las palabras inherentes al modelo se justifica desde la perspectiva de conocer el resultado cuando las palabras a reconocer son las mismas repeticiones con las que se entrenó al modelo. Al respecto se puede observar que los valores obtenidos son, de toda la tabla, los menores y más cercanos entre si (como es de esperarse). Sin embargo al probar la misma palabra pero de diferente hablante (a los que conformaron el corpus de voz), se observan valores cercanos a los de reconocimiento de las palabras de entrenamiento.

Para los demás casos cuando, los mismos hablantes que conformaron el corpus de entrenamiento, pronuncian palabras diferentes, los valores del logaritmo de probabilidad se disparan incluso a valores infinitos (que es, como se puede observar, el caso ideal cuando la palabra en prueba no tiene ninguna correspondencia con el modelo entrenado). Dado que este caso



Si se entrena Albondiga
Con MDTW :

....Pronunciando Albondiga:	....Pronunciando helicoptero:	...Pronunciando indesicion:	...Pronunciando ineptitud:	...Pronunciando Madrid:	...Pronunciando Prejuicios:
-95.6148934	-513.584613	Infinito	-1610.2241	-228.097443	-295.252158
-90.8095484	-1830.84782	Infinito	-290.457696	-399.323492	-454.782296
-102.210169	-559.778528	Infinito	Infinito	-295.585132	-819.70706
-92.4108066	Infinito	Infinito	-306.431797	-714.329929	Infinito
-91.4035276	Infinito	Infinito	-370.489538	-756.235351	Infinito
-104.44168	Infinito	Infinito	-1077.80254	-514.209212	Infinito
-105.607179	Infinito	Infinito	-302.667726	Infinito	Infinito
-107.894611	Infinito	Infinito	-716.120513	-159.455672	-1240.59071
-104.449402	Infinito	Infinito	-435.456788	Infinito	-432.903581
-116.335181	-665.144528	Infinito	-1361.53019	-1230.61542	-426.215734
-96.2417386	-202.23432	Infinito	Infinito	-134.324876	-554.10189
-107.537028	-299.686946	Infinito	-849.590943	Infinito	-707.935234
-102.40673	-396.103608	Infinito	Infinito	-199.194517	-148.816989
-86.288158	-782.888991	-649.838878	Infinito	-198.630766	-457.407774
-90.2465375	Infinito	Infinito	Infinito	-1352.44214	-266.183408
-91.7976524	-579.678536	-269.889759	-939.590381	-305.435854	-249.698229
-99.2289452	-535.645748	Infinito	-1494.79378	Infinito	-448.289613
-103.94119	Infinito	Infinito	-2186.04261	Infinito	Infinito
-97.1302303	Infinito	Infinito	-739.175218	Infinito	-386.685742
-103.034474	-267.064933	-1023.22746	-859.531546	Infinito	-373.537829
-102.154925	-1785.15912	Infinito	-1464.45513	-764.673991	Infinito
-103.492803	Infinito	Infinito	-1908.84677	-315.11661	-696.648094
-99.5892219	-565.930438	-750.544046	-1880.88632	-291.431974	-317.000652
-89.3201944	Infinito	-1230.85689	-1730.98477	Infinito	-230.892583
-99.2224632	Infinito	Infinito	-1055.42451	-2097.00185	-163.032684
-105.097994	Infinito	Infinito	Infinito	-1983.62343	-171.899434
-87.0205162	Infinito	Infinito	Infinito	-1627.05839	-221.607518
-90.882896	Infinito	Infinito	-1476.49858	-2759.13452	-205.855428
-92.2687134	Infinito	Infinito	-432.816301	-1168.62033	-482.698082
-91.9361351	Infinito	Infinito	Infinito	Infinito	Infinito
-101.327267	-513.584613	Infinito	-146.867592	-228.097443	-295.252158
-95.0333145	-299.686946	Infinito	-357.291348	-399.323492	-454.782296
-86.7891758	-396.103608	Infinito	-375.601281	-295.585132	-819.70706
-86.288158	-782.888991	-1230.85689	-1730.98477	-198.630766	-457.407774
-90.2465375	Infinito	Infinito	-1055.42451	-1352.44214	-266.183408
-91.7976524	Infinito	Infinito	Infinito	-305.435854	-249.698229
-87.0205162	Infinito	Infinito	-302.667726	-2317.12657	-277.137205
-90.882896	Infinito	Infinito	-716.120513	Infinito	-130.08389
-92.2687134	-2088.33754	Infinito	-435.456788	Infinito	-490.226213
-99.2170147	-740.169092	Infinito	-859.531546	Infinito	Infinito
-101.327267	-430.027413	-624.369478	-146.867592	-2043.08968	-187.977691
-95.0333145	Infinito	Infinito	-357.291348	-153.429235	-196.579177
-86.7891758	Infinito	Infinito	-375.601281	-629.197597	-323.407329
-357.483597	-1311.43453	-570.694307	-862.474639	Infinito	-355.47075
-179.783034	-140.867854	-985.796459	-1147.32324	-563.485651	-368.359098
-212.498648	-687.63743	Infinito	-1163.79431	-1046.53059	-131.801529
-319.573629	-527.255048	Infinito	-188.711276	-2317.12657	-277.137205
Infinito	Infinito	Infinito	-328.261215	Infinito	-130.08389
-262.215336	-2088.33754	Infinito	-500.141551	Infinito	-490.226213
-99.2170147	-740.169092	-621.987922	-1006.02993	-238.96887	-251.878886

Figura 6.1 Resultados de prueba con diferentes palabras entrenando el modelo para la palabra “albóndiga”.

ideal no se cumple para todos los casos, es necesario delimitar un umbral de aceptación de reconocimiento. Es decir, si por ejemplo se obtiene un valor de probabilidad de -101.327267 y uno de -1006.029935, entonces el primer valor posee una probabilidad más alta de pertenecer al modelo respecto del segundo valor que está muy alejado del valor esperado.

El proceso anterior se repite para cada una de las palabras de vocabulario y en base a la distribución del rango de valores obtenidos cuando la palabra fue correctamente reconocida, entonces se pueden definir los valores máximos de aceptación (loglik) para los tres métodos propuestos de reconocimiento:

- MDTW:300.
- MID:100.
- MRP:300.

Un caso particular ocurre cuando se prueba el método de MID ya que, al ser dividido en dos unidades, el proceso de rechazo o aceptación se define en base a ambos valores obtenidos. La figura 6.2 muestra la tabla resultante para el entrenamiento de la palabra "helicoptero" una vez que se han aplicado los criterios de máxima aceptación.

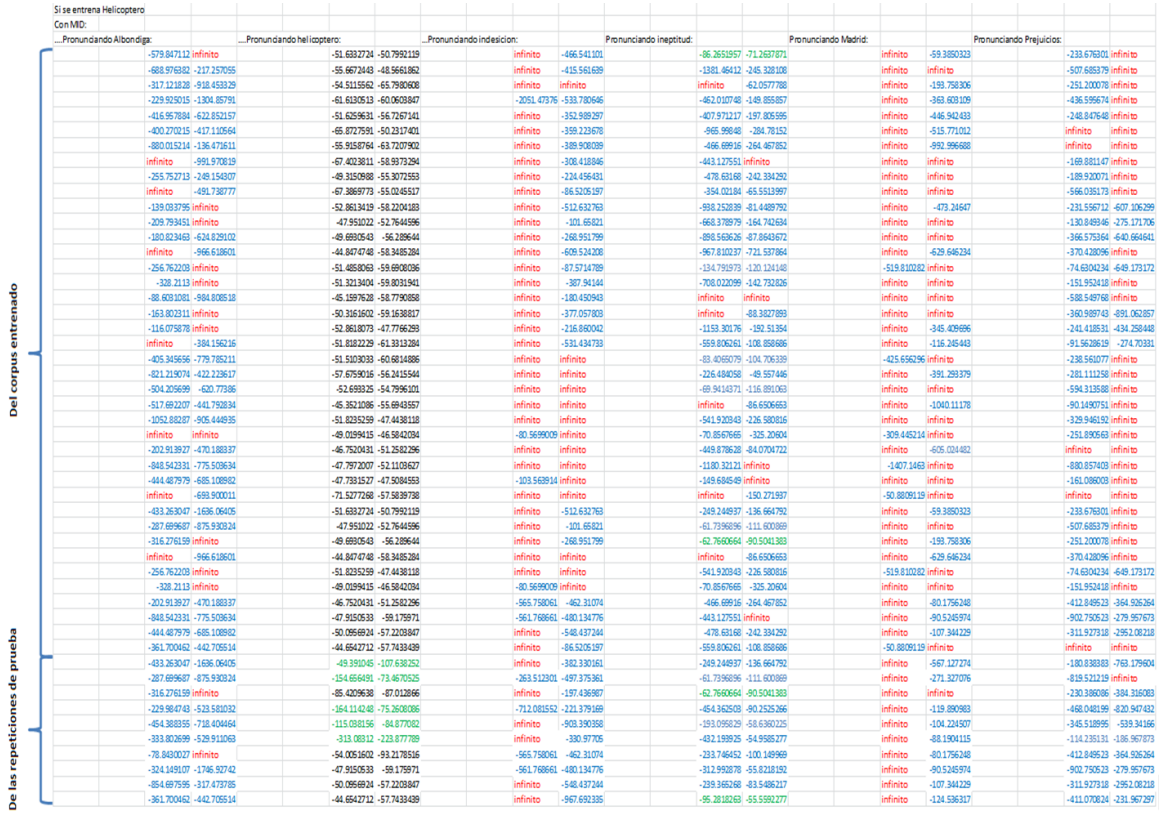


Figura 6.2 Aplicación de criterio de máxima aceptación en prueba del modelo de la palabra "helicóptero" por MID.

En la tabla de la figura 6.2, los valores en color *azul* identifican a los valores de rechazo por el criterio de máxima aceptación. Por otro lado, el rechazo por error de confusión (cuando la palabra pertenece al modelo pero sobre pasa el criterio) se muestra en color *verde*.

En base al análisis de los resultados obtenidos respecto del proceso anterior, se utilizan algunos parámetros con el fin de calificar cada uno de los algoritmos (ver referencia [21]). El cálculo de cada una de las variables a utilizar se obtienen en base a la tabla 6.1.

Los parámetros a calificar son:

Tabla 6.1 Tabla de cálculo de parámetros.

		Predichos	
		Positivo	Negativo
Reales	Positivo	a	b
	Negativo	c	d

- *Exactitud (Ac)*: Proporción del total de predicciones de repeticiones correctas.

$$Ac = \frac{a+d}{a+b+c+d}. \tag{6.1}$$

- *Tasa de Verdadero Positivo (TP)*: Proporción de repeticiones positivas correctamente identificadas.

$$TP = \frac{a}{a+b}. \tag{6.2}$$

- *Tasa de Falso Positivo (Fp)*: Proporción de repeticiones negativas incorrectamente clasificados como positivas.

$$Fp = \frac{c}{c+d}. \tag{6.3}$$

- *Tasa de Verdaderos Negativos (Tn)*: Proporción de repeticiones verdaderas negativas correctamente identificadas.

$$Tn = \frac{d}{c+d}. \tag{6.4}$$

- *Tasa de Falsos Negativos (FN)*: Proporción de repeticiones positivas incorrectamente clasificados como negativas.

$$FN = \frac{b}{a+b}. \tag{6.5}$$

- *Precisión (P)*: Proporción de repeticiones positivas correctamente predichos.

$$P = \frac{a}{a+c}. \tag{6.6}$$

Cada uno de estos parámetros se calculan para cada una de las palabras y para cada uno de los métodos de análisis. Finalmente, de los resultados obtenidos, se obtiene la tabla 6.2 con el método que presenta el mejor desempeño para cada parámetro.

Tabla 6.2 Tabla de Resultados obtenidos por parámetro de calificación en base a umbrales.

Palabra	Ac	Tp	Fp	Tn	FN	P
Albóndiga	MID (97%)	MRP (96%)	MID (1.6%)	MID (98.4%)	MRP (4%)	MID (91.8%)
Helicóptero	MDTW (98%)	MDTW (100%)	MRP (0.8%)	MRP (99.2%)	MDTW (0%)	MRP (95.8%)
Indecisión	MID (97.6%)	**(86%)	MID (0%)	MID (100%)	**(14%)	MID (100%)
Ineptitud	MID (99.6%)	*(100%)	MID (0%)	MID (100%)	*(0%)	MID (100%)
Madrid	MRP (99.3%)	MRP (98%)	MRP (0.4%)	MRP (99.6%)	MRP (2%)	MRP (98%)
Madrid	MDTW (98%)	*(96%)	MID (0.4%)	MID (99.6%)	*(4%)	MID (97.7%)

*=MDTW ó MRP.

**=Cualquiera de los tres métodos.

Tabla 6.3 Evaluación Final del mejor algoritmo para cada palabra.

Palabra	**Método**
Albondiga	MRP
Helicoptero	MDTW
Indesición	MID
Ineptitud	MRP
Madrid	MRP
Prejuicios	MDTW

En la tabla 6.2 se muestran las palabras entrenadas y el algoritmo que obtiene el mejor desempeño en el parámetro probado. En base a dicha tabla se puede identificar al mejor algoritmo según el enfoque deseado. Para el caso de pruebas de dicción, que es el enfoque de esta investigación, es necesario asumir que el usuario que realiza las pruebas de dicción es cooperativo, es decir, desea pronunciar la palabra que se le pida repetir con el fin de probar su dicción (este es el caso de estudiantes o nativos en un idioma en específico). Es por lo tanto que TP y Fn son los parámetros más importantes a tomar en cuenta en la elección del algoritmo a utilizar para probar cada palabra.

En base a lo anterior se obtiene la tabla 6.3. En la tabla, en base a las pruebas realizadas, se propone el mejor algoritmo para las futuras pruebas de dicción respecto de cada palabra. Con base en ello, el método MRP es utilizado en palabras que tienen mayor variedad de fonemas debido a que no altera en medida alguna la información como en el caso de las técnicas de alineamiento dinámico. Por otro lado, la técnica de MDTW es aplicada en palabras de mayor extensión en donde el alineamiento dinámico no influye de manera tan notable en el reconocimiento. Por último, la técnica de MID no posee desempeño superior a las anteriores en ninguno de los casos y es sólo aplicable para aquellas palabras que poseen mayor cantidad de fonemas fricativos.

Hasta este punto, se tienen ya los argumentos necesarios para las pruebas de dicción (dado que se realizaron en base a modelos creados con características especificas de dicción). Sin embargo, y con el fin de obtener alguna información sobre la dicción presente en cada una de

Tabla 6.4 Decisión en base a umbrales

Técnica Utilizada	Excelente Dicción	Buena Dicción	Mala Dicción
MDTW	loglik < 150	loglik entre 150 y 300	loglik > 300
MID	loglik < 100	loglik entre 100 y 150	loglik > 150
MRP	loglik < 150	loglik entre 150 y 300	loglik > 300

las muestras de voz, es necesario calificar a cada dicción presente en ellas. Se decidió tomar en cuenta tres niveles de clasificación (dicción excelente, buena dicción, mala dicción) de acuerdo al logaritmo de probabilidad presente en cada repetición. Los valores de clasificación y de umbral en base al logaritmo de probabilidad acumulada se muestran en la tabla 6.4.

Cabe señalar en este punto que, mediante los métodos de reconocimiento propuestos, es posible el diferenciar entre dicciones, sin embargo estas diferencias se califican de acuerdo a los valores de umbral utilizados, los cuales se obtienen en base a las observaciones respecto de las pruebas realizadas y a relaciones subjetivas encontradas entre la calificación del loglik y la información auditiva (en materia de dicción) de cada repetición.

Con la investigación previa y los resultados obtenidos, la última fase de nuestro trabajo consiste en la creación de una interfaz de usuario para la realización de futuras pruebas de dicción integrando todos los métodos de reconocimiento propuestos. Se busca que la interfaz sea sencilla de utilizar y que la información de salida (calificación obtenida) sea fácil de entender. Para este fin se optó por realizar la implementación en MATLAB en el entorno GUIDE (para más información sobre la forma de programación en el entorno GUIDE se sugiere la revisión de la referencia [28]).

La interfaz de usuario final desarrollada se muestra en la figura 6.3 (para una explicación detallada sobre el uso de la interfaz desarrollada revisar el apéndice A).

El proceso de prueba consiste a grandes rasgos en seleccionar, del apartado de *dicción base*, una palabra del menú *palabra a entrenar* y escuchar la dicción base correspondiente a dicha palabra. Después, en la sección de dicción de prueba, se graba y escucha la dicción contra la que se quiere probar. Finalmente, en la sección de resultados se muestra la calificación (en base a colores) obtenida con la dicción a probar.

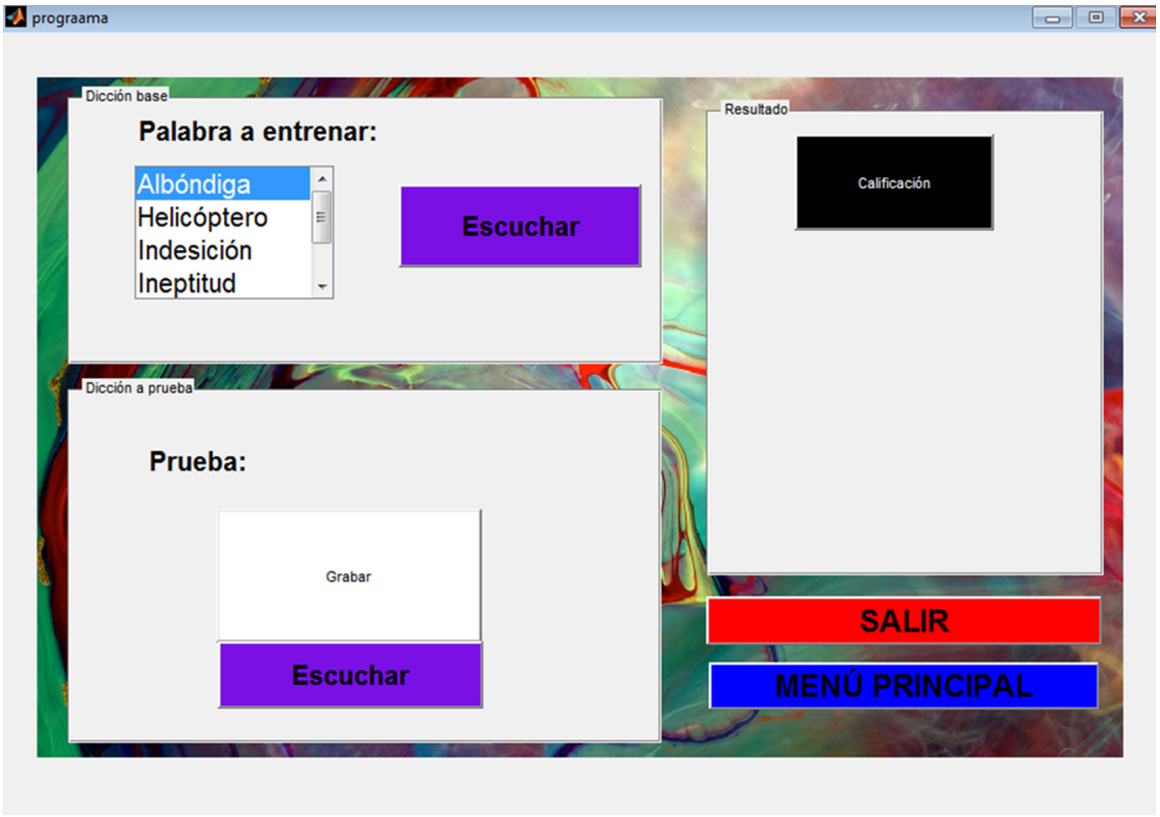


Figura 6.3 Interfaz gráfica de pruebas elaborada en MATLAB 2011.

Capítulo 7

Conclusiones.

Los algoritmos presentados en los capítulos 4, 5 y 6 tienen la finalidad de ser aplicados para la realización de pruebas de dicción en el idioma español. Debido a este enfoque es que se utilizaron diversas técnicas y parámetros para calificar, de un vocabulario definido, repeticiones de voz con diferentes dicciones. Específicamente para esta investigación, se utilizaron Modelos Ocultos de Markov junto con algunas modificaciones (Modelos Ocultos de Markov con DTW (MDTW), Modelos Ocultos de Markov con DTW aproximados por izquierda y derecha (MID) y Modelos Ocultos de Markov con relleno de palabras (MRP)). En base a los resultados obtenidos, se concluye que es posible la aplicación en software para reconocimiento del habla con la finalidad de realizar pruebas de dicción en un idioma (en este caso español) contando con una eficiencia (para cada uno de los algoritmos presentados) mayor al 90 % (como se muestra en la tabla 6.2). Con ello se cumple la hipótesis planteada desde un inicio.

Se analizaron varios esquemas de reconocimiento de voz de los cuales se utilizaron y propusieron aquellos que poseen una estructura adecuada para la aplicación con Modelos Ocultos de Markov.

Uno de los puntos a resaltar dentro de la investigación es que, el optar por realizar un corpus de voces multilocutor con características específicas, influye directamente en el reconocimiento. Para el caso específico de esta investigación, se observó que utilizar un corpus de voz con características especiales de dicción, logra el desarrollo de modelos capaces de caracterizar de una manera más especifica cada palabra. Con esto se concluye que es posible adaptar los modelos de Markov para realizar pruebas de dicción.

Cabe señalar en este punto, que fue posible el diferenciar entre diferentes dicciones mediante los métodos propuestos, sin embargo, estas diferencias se califican de acuerdo a los valores de umbral utilizados. Estos valores se obtienen en base a observaciones respecto de las pruebas realizadas. Es por ello que, entre las mejoras posibles que se pueden implementar está, el diseñar las pruebas de dicción en consulta con un especialista en lenguaje, el cual, puede ayudar a mejorar la clasificación subjetiva realizada al modificar el umbral de decisión de acuerdo a métricas objetivas.

El conjunto de algoritmos integrados en la interfaz final posee la flexibilidad suficiente para ampliar la cantidad de palabras de prueba de dicción y, más aún, creemos que se pueden ampliar las pruebas de dicción a otros idiomas modificando el corpus de voces al idioma que se deseé.

De los resultados experimentales se observa que, para el caso de pruebas de dicción en el idioma español, el algoritmo que posee el mejor desempeño de forma general es el MRP, sin embargo, los demás algoritmos poseen un desempeño aceptable y muy cercano. Es por eso que para el caso de las palabras en prueba presentes, la metodología final de prueba conviene que utilice el mejor método de reconocimiento para cada palabra. En base a lo anterior y sin descartar ninguno de los métodos desarrollados, estas técnicas pueden ser utilizadas en otras aplicaciones de reconocimiento de voz; realizando las pruebas pertinentes para la elección del mejor algoritmo según sea el caso.

Entre los problemas que se encontraron al realizar las pruebas de dicción, se encontró que es necesario realizarlas en entornos ideales de aislamiento acústico dado que las pruebas son vulnerables al ruido y es posible que se obtengan resultados no deseados si no se cumple con esta condición. Es en base a esto que se recomienda realizar un buen filtrado antes de realizar las pruebas de dicción para evitar condiciones de inestabilidad.

Entre las ventajas del uso de Modelos Ocultos de Markov respecto de otras técnicas se encuentra la reducción del tiempo de cálculo. A pesar de manejar un gran número de información, el proceso es relativamente rápido dadas las características de esta técnica. Esto es debido a que, como se explicó anteriormente, la etapa de entrenamiento solo se realiza una sola vez y

es almacenada para, cada vez que se realice el reconocimiento, se haga uso de esta información sin necesidad de volver a realizar el entrenamiento.

Apéndice A: Uso de la interfaz de usuario

Una vez que se identificaron los algoritmos pertinentes para cada una de las palabras del vocabulario utilizado, como resultado de la investigación, se creó una interfaz en MATLAB con el fin de realizar futuras pruebas de dicción de palabras aisladas en el idioma español.

A.1 Antes de comenzar.

Para poder utilizar la interfaz desarrollada es necesario pre-cargar algunos archivos especiales (descargable del sitio web de la referencia [27]). La lista de archivos necesarios son:

1. HMM
2. KPMstats
3. KPMtools
4. netlab3.3

Una vez cargados los archivos en MATLAB, se puede ejecutar en linea de comando la interfaz desarrollada.

A.2 Estructura y funcionamiento.

La estructura de la interfaz desarrollada mediante el GUIDE de MATLAB se esquematiza de la siguiente manera (ver figura A.1):

- Menú principal
  - Iniciar: Abre la interfaz para la realización de nuevas pruebas de dicción.
  - Ejemplos: Abre una interfaz con algunas dicciones de prueba.

Si se desea aprender para iniciar, cómo funciona la interfaz para las pruebas de dicción, es necesario presionar el botón **Ejemplos** (ver figura A.3)

Figura A.1 Menú principal.

Figura A.2 Submenú **Iniciar**.

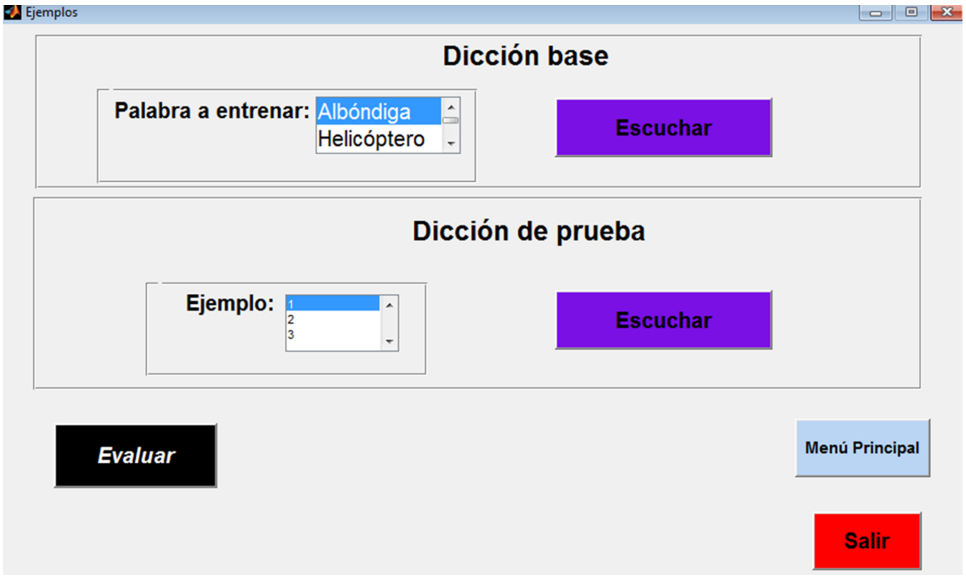


Figura A.3 Submenú **Ejemplos**.

Dado que lo primero que se necesita para probar la dicción es conocer la dicción "esperada" (base), es necesario definir, de entre la lista de palabras posibles (vocabulario predefinido), aquella que se desea probar. Esto se lleva a cabo seleccionando de la lista "Palabra a entrenar" la palabra elegida y presionando el botón "Escuchar" de la sección "Dicción base".

Una vez conocido este parámetro, en la sección "Dicción de prueba " se pueden probar algunas dicciones ejemplo (3 por cada palabra) y conocer el resultado obtenido en base a la dicción encontrada. Para llevar a cabo este proceso es necesario elegirlos de la lista "Ejemplo" y presionar el botón "Escuchar".

El proceso de calificación de dicción se programó en base al despliegue de 3 colores diferentes:

- Color verde: Excelente dicción,
- Color Amarillo: Buena dicción,
- Color Rojo: Mala dicción.

Para conocer la calificación presente en el ejemplo de la palabra seleccionada es necesario presionar el botón "Evaluar". En la figura A.4 se muestra un ejemplo del resultado obtenido.

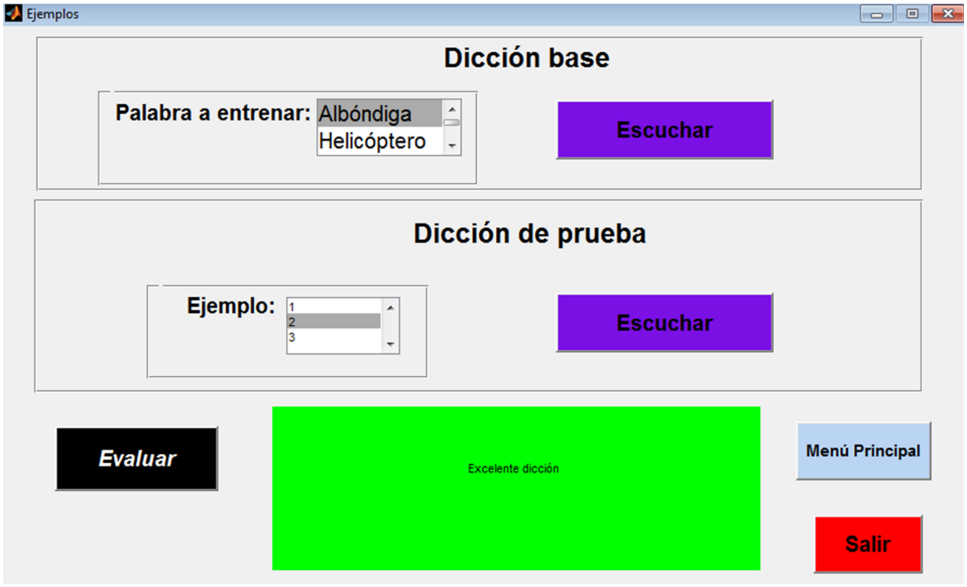


Figura A.4 Ejemplo propuesto de la prueba de dicción de la palabra "Albóndiga"

Por otro lado, el botón "Iniciar" del menú principal, es utilizado para probar nuevas repeticiones de voz (ver figura A.2). El proceso para seleccionar y escuchar la dicción base es similar a la descrita en la fase de pruebas.

Después de realizar este proceso, en el menú "Dicción de prueba", se puede grabar y escuchar (mediante el botón correspondiente) la repetición de la palabra a probar. Finalmente, al igual que el submenú anterior, mediante el botón "Calificación" se obtiene el resultado de la dicción encontrada (figura A.4).

Apéndice B: Lista de programas.

La lista de algoritmos programados en MATLAB que se utilizaron en la realización de esta tesis son:

1. Lectura de archivos de audio.
2. Filtro pasa-voz.
3. Recorte.
4. Pre-énfasis.
5. Normalización.
6. Alineamiento Dinámico en el tiempo (DTW).
7. Segmentación, enventanado y extracción de la energía.
8. Cuantificación.
9. Modelos Ocultos de Markov con DTW (MDTW).
10. Modelos Ocultos de Markov con DTW aproximado por izquierda y derecha (MID).
11. Modelos Ocultos de Markov con relleno de palabras (MRP).

B.1 Lectura de archivos de audio.

```
%%%%%%%%%%
%Dirección%
%%%%%%%%%%

%Nombre del archivo: direccion.m%

%Esta función realiza el cargado desde la ubicación indicada de
%cada una de las repeticiones de la palabra a entrenar.Para ello
%los archivos de audio están subdivididos en carpetas según el
%número de hablante y el numero de repetición de cada palabra.

function [palabra] = direccion(a,b,p_entrenada)
%En la entrada a=indice del hablante
%              b=numero de repeticion
%              p_entrenada=palabra a extraer
numero_hablante = num2str(a); ...
                                                          %Indice ...
    del hablante
repeticion = num2str(b); ...
                                                                ...
    %Indice del dato
palabra = strcat('Hablante',numero_hablante,'\', p_entrenada, ...
    '_',repeticion, '.wav');%Dirección + indice
```

B.2 Filtro pasa-voz.

```
%%%%%%%%%
%Pasavoz%
%%%%%%%%%

%Nombre del archivo: pasavoz.m

%Esta funcion realiza el filtrado de la señal de voz, permitiendo ...
    solo el
%paso de las frecuencias que de la voz humana ( de 200 a 8000 Hz)
%Se configura desde 100 para dejar un margen (/Fuente:B. Facundo, O. ...
    Flavio,
%P. Felipe, G. Mariano, Reconocimiento de voz para la aplicación en ...
    domótica,
%Universidad Tecnológica Nacional:Facultad Regional San Nicolás, 2008).
function filtrada=pasavoz(voz)
filtro = fir1(800,[100 8000]/22050);    %configuracion del filtro
filtrada=filter(filtro,1,voz);          %Señal filtrada
```

B.3 Recorte.

```
%%%%%%%%%
%Recorte%
%%%%%%%%%

%Nombre del archivo: recorte.m

%Esta función se encarga de extraer de la señal de voz grabada
%unicamente las porciones de señal que poseen información con cierto ...
    margen
%de error

function b=recorte(voz)
 longitud = length(voz);                    %Calcula la longitud del ...
     vector de voz ingresado
d=max(abs(voz));                            %Obtiene el maximo ...
    absoluto de la señal de voz
voz=voz/d;                                  %Normaliza respecto de ...
    ese valor
energ_prom = sum(voz.*voz)/longitud;        %Obtiene la energia ...
    promedio de la señal
THRES = 0.01;                               %Elige el valor de ...
    umbral de desicion entre muestras
j=1;                                        %Contador para inicio ...
    del ciclo
for i = 1:400:longitud-400;                 %El tamaño de la ventana ...
    se define como 400 que a la frecuencia de muestreo( 44100 ) son ...
    10 ms.
seg = voz(i:i+399);                         %es un segmento de voz ...
    de ancho 400
e = sum(seg.*seg)/400;                      %Se obtiene la energia ...
    normalizada en esa muestra
```

```
21 if( e> THRES*energ_prom);                    %si la energia es mayor ...
       que el valor de thres
22       a(j)=i;                                %Indices de las ventanas ...
             que superan el umbral
23       j=j+1;
24 end;
25 end
26 c=length(a);                                 %Calcula la longitud del ...
       vector
27 contador=0;                                  %Inicia el contador de ...
       elementos
28 p=0;                                         %Indice para desicion de ...
       inicio y fin
29 for i=1:c-1
30      if contador==0                          %Si el contador = 0
31      inicio=a(i);                            %podria ser el inicio de ...
            la palabra
32      end
33      distancia=a(i+1)-a(i);                  %Se calcula la distancia ...
            entre ventanas subsecuentes
34     if distancia<10000                       %Si la distancia es ...
           menor que 0.125 milisegundos
35         contador=contador+1;                 %aumenta el contador de ...
               ventanas;
36     else                                     %de otra forma
37         fin=a(i);                            %el indice corresponde ...
               al final de la ventana
38         if contador<10                       %Si el numero de ...
               ventanas es menor a 10
39             contador=0  ;                    %no es una palabra
40             inicio=0;                        %El indice no es el ...
                   inicio que se busca ni
41             fin=0;                           %el final;
42         else                                 %de otra forma
```

```
43              p=1;                                %se toma esos indices ...
                    como inicio y fin
44            break;
45      end
46      end
47  end
48  if p==0                                         %Si no posee otros ...
        elementos de indices y fin de sonido,
49       inicio=a(1) ;                              %el primer valor es el ...
             inicio;
50       fin=a(end);                                %y el ultimo valor es el fin
51  end
52
53       b=voz(inicio:fin,1);                       %Recorta la señal con ...
             esos indices encontrados
```

B.4 Pre-énfasis.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Pre-énfasis de la señal de voz%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Nombre del archivo: preenfasis.m

%La función de pre-enfasis enfatiza las frecuencias formantes para no
%perderlas en procesos siguientes. La mayoria de la información está ...
   en las
%frecuencias bajas. Remueve la componenete de CD de la señal y aplana
%espectralmente la señal. Es un filtro pasa altas de primer orden.

function [voz2]=preenfasis(voz)
N=size(voz);                              %Tamaño de la señal
a=0.94;                                   %Toma un valor entre 0.9 y 1.0
voz2(1,1)=voz(1,1);
for i=2:N(1)
 voz2(i,1)=voz(i,1)-(a.*voz(i-1,1));      %Selecciona solamente el ...
    canal izquierdo
end
                                          %Se obtiene la señal con ...
                                             pre-énfasis
end
```

B.5 Normalización.

```
%%%%%%%%%%%%%%%
%Normalización%
%%%%%%%%%%%%%%%

%Nombre del archivo: normal.m

%Esta función normaliza las muestra de voz


function voznorm=normal(vozenf)
d=max(abs(vozenf));         %Obtiene el maximo absoluto de la señal ...
    de voz
voznorm=vozenf./d;          %Normalización
```

B.6 Alineamiento Dinámico en el tiempo (DTW).

```
1
2  %DTW%
3
4  %Nombre del archivo: dtw.m
5
6  %Este programa fue tomado del sitio web:
7  %http://www.ee.columbia.edu/ln/rosa/matlab/dtw (el cual se encuentra
8  %referenciado en al presente investigación).
9
10 function d2x=dtw(d1,d2,sr)
11
12  ml = min(length(d1),length(d2));
13
14  % Calculate STFT features for both sounds (25% window overlap)
15  D1 = specgram(d1,512,sr,512,384);
16  D2 = specgram(d2,512,sr,512,384);
17
18  % Construct the 'local match' scores matrix as the cosine distance
19  % between the STFT magnitudes
20  SM = simmx(abs(D1),abs(D2));
21
22  % Use dynamic programming to find the lowest-cost path between the
23  % opposite corners of the cost matrix
24  % Note that we use 1-SM because dp will find the *lowest* total cost
25  [p,q,C] = dp(1-SM);
26
27
28  % Bottom right corner of C gives cost of minimum-cost alignment of ...
       the two
29  C(size(C,1),size(C,2));
30  % This is the value we would compare between different
31  % templates if we were doing classification.
```

```
32
33  % Calculate the frames in D2 that are indicated to match each frame
34  % in D1, so we can resynthesize a warped, aligned version
35  D2i1 = zeros(1, size(D1,2));
36  for i = 1:length(D2i1); D2i1(i) = q(min(find(p ≥ i))); end
37  % Phase-vocoder interpolate D2's STFT under the time warp
38  D2x = pvsample(D2, D2i1-1, 128);
39  % Invert it back to time domain
40  d2x = istft(D2x, 512, 512, 128);
41  d2x = resize(d2x', length(d1),1);
42
43
44 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46
47 function M = simmx(A,B)
48 % M = simmx(A,B)
49 %    calculate a sim matrix between specgram-like feature matrices A ...
       and B.
50 %    size(M) = [size(A,2) size(B,2)]; A and B have same #rows.
51 % 2003-03-15 dpwe@ee.columbia.edu
52
53 % Copyright (c) 2003 Dan Ellis <dpwe@ee.columbia.edu>
54 % released under GPL - see file COPYRIGHT
55
56 EA = sqrt(sum(A.^2));
57 EB = sqrt(sum(B.^2));
58
59 % this is 10x faster
60 M = (A'*B)./(EA'*EB);
61
62
63 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
64 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65
```

```
66 function [p,q,D] = dp(M)
67 % [p,q] = dp(M)
68 %    Use dynamic programming to find a min-cost path through matrix M.
69 %    Return state sequence in p,q
70 % 2003-03-15 dpwe@ee.columbia.edu
71
72 % Copyright (c) 2003 Dan Ellis <dpwe@ee.columbia.edu>
73 % released under GPL - see file COPYRIGHT
74
75 [r,c] = size(M);
76
77 % costs
78 D = zeros(r+1, c+1);
79 D(1,:) = NaN;
80 D(:,1) = NaN;
81 D(1,1) = 0;
82 D(2:(r+1), 2:(c+1)) = M;
83
84 % traceback
85 phi = zeros(r,c);
86
87 for i = 1:r;
88   for j = 1:c;
89     [dmax, tb] = min([D(i, j), D(i, j+1), D(i+1, j)]);
90     D(i+1,j+1) = D(i+1,j+1)+dmax;
91     phi(i,j) = tb;
92   end
93 end
94
95 % Traceback from top left
96 i = r;
97 j = c;
98 p = i;
99 q = j;
100 while i > 1 & j > 1
```

```
101   tb = phi(i,j);
102   if (tb == 1)
103     i = i-1;
104     j = j-1;
105   elseif (tb == 2)
106     i = i-1;
107   elseif (tb == 3)
108     j = j-1;
109   else
110     error;
111   end
112   p = [i,p];
113   q = [j,q];
114 end
115
116 % Strip off the edges of the D matrix before returning
117 D = D(2:(r+1),2:(c+1));
118
119 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
120 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
121
122 function c = pvsample(b, t, hop)
123 % c = pvsample(b, t, hop)   Interpolate an STFT array according to ...
      the 'phase vocoder'
124 %     b is an STFT array, of the form generated by 'specgram'.
125 %     t is a vector of (real) time-samples, which specifies a path ...
      through
126 %     the time-base defined by the columns of b.  For each value of t,
127 %     the spectral magnitudes in the columns of b are interpolated, and
128 %     the phase difference between the successive columns of b is
129 %     calculated; a new column is created in the output array c that
130 %     preserves this per-step phase advance in each bin.
131 %     hop is the STFT hop size, defaults to N/2, where N is the FFT size
132 %     and b has N/2+1 rows.  hop is needed to calculate the 'null' phase
133 %     advance expected in each bin.
```

```
134 %     Note: t is defined relative to a zero origin, so 0.1 is 90% of
135 %     the first column of b, plus 10% of the second.
136 % 2000-12-05 dpwe@ee.columbia.edu
137 % $Header: ...
      /homes/dpwe/public_html/resources/matlab/dtw/../RCS/pvsample.m,v ...
      1.3 2003/04/09 03:17:10 dpwe Exp $
138
139 if nargin < 3
140   hop = 0;
141 end
142
143 [rows,cols] = size(b);
144
145 N = 2*(rows-1);
146
147 if hop == 0
148   % default value
149   hop = N/2;
150 end
151
152 % Empty output array
153 c = zeros(rows, length(t));
154
155 % Expected phase advance in each bin
156 dphi = zeros(1,N/2+1);
157 dphi(2:(1 + N/2)) = (2*pi*hop)./(N./(1:(N/2)));
158
159 % Phase accumulator
160 % Preset to phase of first frame for perfect reconstruction
161 % in case of 1:1 time scaling
162 ph = angle(b(:,1));
163
164 % Append a 'safety' column on to the end of b to avoid problems
165 % taking *exactly* the last frame (i.e. 1*b(:,cols)+0*b(:,cols+1))
166 b = [b,zeros(rows,1)];
```

```
167
168 ocol = 1;
169 for tt = t
170   % Grab the two columns of b
171   bcols = b(:,floor(tt)+[1 2]);
172   tf = tt - floor(tt);
173   bmag = (1-tf)*abs(bcols(:,1)) + tf*(abs(bcols(:,2)));
174   % calculate phase advance
175   dp = angle(bcols(:,2)) - angle(bcols(:,1)) - dphi';
176   % Reduce to -pi:pi range
177   dp = dp - 2 * pi * round(dp/(2*pi));
178   % Save the column
179   c(:,ocol) = bmag .* exp(j*ph);
180   % Cumulate phase, ready for next frame
181   ph = ph + dphi' + dp;
182   ocol = ocol+1;
183 end
184
185
186 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
187 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
188
189
190 function x = istft(d, ftsize, w, h)
191 % X = istft(D, F, W, H)                    Inverse short-time Fourier ...
      transform.
192 %   Performs overlap-add resynthesis from the short-time Fourier ...
      transform
193 %   data in D.  Each column of D is taken as the result of an F-point
194 %   fft; each successive frame was offset by H points (default
195 %   W/2, or F/2 if W==0). Data is hann-windowed at W pts, or
196 %       W = 0 gives a rectangular window (default);
197 %       W as a vector uses that as window.
198 %       This version scales the output so the loop gain is 1.0 for
199 %       either hann-win an-syn with 25% overlap, or hann-win on
```

```
200 %       analysis and rect-win (W=0) on synthesis with 50% overlap.
201 % dpwe 1994may24.  Uses built-in 'ifft' etc.
202 % $Header: ...
      /home/empire6/dpwe/public_html/resources/matlab/pvoc/RCS/istft.m,v ...
      1.5 2010/08/12 20:39:42 dpwe Exp $
203
204 if nargin < 2; ftsize = 2*(size(d,1)-1); end
205 if nargin < 3; w = 0; end
206 if nargin < 4; h = 0; end  % will become winlen/2 later
207
208 s = size(d);
209 if s(1) ~= (ftsize/2)+1
210   error('number of rows should be fftsize/2+1')
211 end
212 cols = s(2);
213
214 if length(w) == 1
215   if w == 0
216     % special case: rectangular window
217     win = ones(1,ftsize);
218   else
219     if rem(w, 2) == 0   % force window to be odd-len
220       w = w + 1;
221     end
222     halflen = (w-1)/2;
223     halff = ftsize/2;
224     halfwin = 0.5 * ( 1 + cos( pi * (0:halflen)/halflen));
225     win = zeros(1, ftsize);
226     acthalflen = min(halff, halflen);
227     win((halff+1):(halff+acthalflen)) = halfwin(1:acthalflen);
228     win((halff+1):-1:(halff-acthalflen+2)) = halfwin(1:acthalflen);
229     % 2009-01-06: Make stft-istft loop be identity for 25% hop
230     win = 2/3*win;
231   end
232 else
```

```
233   win = w;
234 end
235
236 w = length(win);
237 % now can set default hop
238 if h == 0
239   h = floor(w/2);
240 end
241
242 xlen = ftsize + (cols-1)*h;
243 x = zeros(1,xlen);
244
245 for b = 0:h:(h*(cols-1))
246   ft = d(:,1+b/h)';
247   ft = [ft, conj(ft([((ftsize/2)):-1:2]))];
248   px = real(ifft(ft));
249   x((b+1):(b+ftsize)) = x((b+1):(b+ftsize))+px.*win;
250 end;
```

B.7 Segmentación, enventanado y extracción de la energía.

```
%%%%%%%%%%%%%%%%%%%%%%%%%
%Entramado y enventanado%
%%%%%%%%%%%%%%%%%%%%%%%%%

%Nombre del archivo: ventanasmod1.m

%La función es realizar la segmentación y aplica a cada segmento una ...
    ventana Hamming en dicho orden para
%poder extraer las características de la señal para procesos posteriores
%del reconocimiento

function Etrama=ventanasmod1(voz2,Fs2)
voz2 = voz2';
tt = 0.030;                                 %Ancho del segmento ...
    30mseg ( Debe de estar entre 10 y 45 msegundos)-ENTRAMADO
nmt = Fs2*tt;                               %Para convertir ese ...
    tamaño de ventana en numero de muestras cada ventana
M=floor((nmt)/2);                           %Indica el valor de ...
    traslape entre ventanas (50 por ciento aproximadamente)
Ntramas =ceil((length(voz2)-nmt)/M);        %Numero de tramas en que ...
    se divide la señal
h = hamming(nmt);                           %Define el ancho de la ...
    ventana hamming-ENVENTANADO
E1=0;
for i = 0:Ntramas-1                         %Recorre la matriz
temp = voz2(i*M+1:i*M+nmt);                 %y entrama la señal ...
    original,
sp=h'.*temp;                                %para después ...
    multiplicarla por la ventana hamming
E1(i+1,1)=(log(mean((sp.^2))));             %y obtener la energia ...
    por trama
end
```

```
24 Emin1=min(E1);                                %Cálculo del minimo de ...
       la energia de todas las tramas
25 Etrama=E1-Emin1;                              %y obtener el vector ...
       final de energia por trama
```

B.8 Cuantificación.

```
1
2
3  %%%%%%%%%%%%%%%%%%%%%%
4  %Cuantificador escalar%
5  %%%%%%%%%%%%%%%%%%%%%%
6
7
8  %Nombre del archivo: cuantificador.m
9
10 %El cuantificador escalar clasifica los niveles de energía en un valor
11 %discreto
12
13 function [cuantificado]=cuantificador(vector)
14 Etrama=vector;                                                    ...
       %Asigna el nombre de vector al vector energia de entrada
15 M=max(Etrama);                                                   ...
       %Obtiene el valor máximo que tiene el vector
16 n=min(Etrama);                                                   ...
       %y el más pequeño ( en este caso siempre es cero gracias al ...
       proceso fuera del ciclo
17 divi=30;                                                         ...
       %Determina el número de escalones ( posibles valores) que puede ...
       tomar la cuantificador
18 rango=(M-n)/divi;                                                ...
       %Determina el rango de avance entre escalones (ancho del escalon)
19 [a,b]=size(Etrama);                                              ...
       %Determina el tamaño de muestras a cuantizar
20 cuantificado=ones(size(Etrama));                                 ...
       %Crea vector de cuantificacion
21 for i=1:a                                                        ...
       %desde el primer elemento al último
```

```
22      for j=1:divi                                                    ...
           %Las opciones que puede tomar ( en este caso 14 valores ...
           diferentes (0-30))
23      if (Etrama(i,1)≥ (rango*(j-1)) &&  Etrama(i,1) ≤ (rango)*j)    ...
           %Si cumple con la condición para caer en este escalon
24      cuantificado(i,1)=j;                                           ...
           %toma el valor que lo representará
25      break                                                          ...
           %y sale del ciclo y continua con los demas elementos del vector
26      elseif Etrama(i,1)==M                                          ...
           %de otra forma retoma el ciclo y aumenta "j" hasta encontrar ...
           el escalon que le corresponde
27            cuantificado(i,1)=30;                                    ...
                 %Y continua el proceso hasta terminar con todo el ...
                 vector de energia y a la salida se tiene el
28      end                                                            ...
           %vector cuantizado
29      end
30  end
31  end
```

B.9 Modelos Ocultos de Markov con DTW (MDTW).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Modelos Ocultos de Markov con DTW%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Este algoritmo realiza el reconocimiento de la palabra pronuniada ...
    en base a un alineamiento dinamico en el
%tiempo previo a la construccion de las matrices a utilizar con ...
    Markov y muestra el resultado del análisis en
%base a su dicción

%%%%%%%%%%%%%%%%%%%%%
%Inicio del programa%
%%%%%%%%%%%%%%%%%%%%%

clear all   %Borrar variables
clc         %Limpiar pantalla

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Definición de la palabra de entrenamiento%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


palabra='prejuicios';                      %Palabra a entrenar
[palabra] = direccion(5,1,palabra);       %Repetición para el alineamiento
[voz,Fs2,NBITS2]=wavread(palabra);        %Lectura de la repetición
voz=pasavoz(voz);                         %Aplicación de filtro pasavoz
vozr=recorte(voz);                        %Recorte de la señal
vozenf=preenfasis(vozr);                  %Aplica preénfasis a la señal
vozp=normal(vozenf);                      %Normalización

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Entrenamiento de la palabra elegida%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mult=0;                                                    %Ciclo de ...
    entrenamiento,
for j=1:4                                                  %para los ...
    hablantes que conforman el corpus
    for i=1:10                                             %y las ...
        repeticiones que se tiene de cada uno
        palabra='prejuicios';                               %Definición de ...
            la palabra elegida,
        [palabra] = direccion(j,i,palabra);                %del hablante y ...
            repetición en curso
        [voz,Fs2,NBITS2]=wavread(palabra);                 %Lectura de la ...
            repetición
        voz=pasavoz(voz);                                  %Aplicación de ...
            filtro pasavoz
        vozr=recorte(voz);                                 %Recorte de la señal
        vozenf=preenfasis(vozr);                           %Aplica ...
            preénfasis a la señal
        voz=normal(vozenf);                                %Normalización
        vozw=dtw(vozp,voz,Fs2);                            %Aplicación del ...
            alineamiento dinámico (DTW) respecto de la repetición elegida
        vozenerg=ventanasmod1(vozw,Fs2);                   %Segmentación, ...
            enventanado y obtención de la energía por trama
        vozenerg=normal(vozenerg);                         %Normalización ...
            del vector de energía
        [Cet]=cuantificador(vozenerg);                     %Entrega un ...
            vector cuantificado de la energia por trama
        Edetrama(i+mult,:)=Cet(:,1)';                      %Almacenamiento ...
            en la matriz de símbolos
    end
    mult=mult+10;
end
```

```
j=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Entrenamiento con del modelo de Markov%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[a,b]=size(Edetrama); ...
                                                             %Creación ...
    de las matrices para entrenamiento
A=(diag(ones(1,b),0)+ ...
    diag(ones(1,b-1),1)+diag(ones(1,b-2),2))*(1/3);          %Matriz ...
    cuadrada de probabilidad de transición equiprobable
A(b,b)=1;
A(b-1,b)=0.5;
A(b-1,b-1)=0.5;
B=(ones(b,30))*(1/30); ...
                                                            %Matriz de ...
    probabilidad de generación de símbolos equiprobable
inicio=zeros(1,b)';
inicio(1,1)=1; ...
                                                                   ...
    %Matriz de probabilidad de inicio
[l1,mpin,main,mbin]=learn_dhmm(Edetrama,inicio,A,B,50); ...
                        %Función de entrenamiento
j=0;



%%%%%%%%%%%%%%%%%%%%
%Prueba del sistema%
%%%%%%%%%%%%%%%%%%%%

mult=0;
for j=5                                                        %Prueba ...
    del hablante j,
```

```
78      for i=1:10                                                  ...
           %repetición i y
79          palabra='prejuicios';                                   %de la ...
               palabra seleccionada
80          [palabra] = direccion(j,i,palabra);                     ...
               %Repetición para el alineamiento
81          [vozorig,Fs2,NBITS2]=wavread(palabra);                  %Lectura ...
               de la repetición
82          voz1=vozorig;
83          voz1=pasavoz(voz1);                                     ...
               %Aplicación de filtro pasavoz
84          vozr=recorte(voz1);                                     %Recorte ...
               de la señal
85          vozenf=preenfasis(vozr);                                %Aplica ...
               preénfasis a la señal
86          voz1=normal(vozenf);                                    ...
               %Normalización
87          vozw=dtw(vozp,voz1,Fs2);                                ...
               %Aplicación del alineamiento dinámico (DTW) respecto de ...
               la repetición elegida
88          vozenergia2=ventanasmod1(vozw,Fs2);                     ...
               %Segmentación, enventanado y obtención de la energía por ...
               trama
89          vozenergia2=normal(vozenergia2);                        ...
               %Normalización
90          [Cet2]=(cuantificador(vozenergia2))';                   %Entrega ...
               un vector cuantificado de la energia por trama
91          orcentaje=logo(Cet2,mpin,main,mbin);                    %Cálculo ...
               del logaritmo de probabilidad de observar dicha secuencia
92
93          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94          %Prueba de dicción en base a umbrales%
95          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96
97          %En base a la dicción presente en la muestra de voz a probar se
```

```
98          %despliega el resultado encontrado en base al umbral
99
100         if abs(orcentaje)<150
101             fprintf('Excelente Dicción');
102             color=1;
103             despliegue(voz1,Cet2,vozorig,color)
104         else if abs(orcentaje)<300
105                 fprintf('Buena Dicción');
106                 color=2;
107                 despliegue(voz1,Cet2,vozorig,color)
108             else
109                 fprintf('Mala Dicción,repita el ejercicio')
110             end
111         end
112
113
114
115     end
116     mult=mult+10;                                          %El proceso ...
        continua hasta terminar con las muestras a probar
117 end
```

B.10 Modelos Ocultos de Markov con DTW aproximado por izquierda y derecha (MID).

```
1
2  %Modelos Ocultos de Markov con DTW aproximado por izquierda y ...
       derecha (MID)%%
3
4
5  %Este algoritmo realiza el reconocimiento de la palabra pronuniada ...
       en base
6  %a un alineamiento dinamico de dos fases (parte izquierda y derecha ...
       de la
7  %señal). Mismas que se entrenan y prueban para dar una calificación ...
       en base
8  %a la dicción encontrada.
9
10 %%%%%%%%%%%%%%%%%%%%%%
11 %Inicio del programa%
12 %%%%%%%%%%%%%%%%%%%%%%
13
14 clear all   %Borrar variables
15 clc         %Limpiar pantalla
16
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18 %Definición de la palabra de entrenamiento%
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20
21 palabra='madrid';                    %Palabra a entrenar
22 [palabra] = direccion(5,1,palabra); %Repetición para el alineamiento
23 [voz,Fs2,NBITS2]=wavread(palabra);  %Lectura de la repetición
24 voz=pasavoz(voz);                    %Aplicación de filtro pasavoz
25 vozr=recorte(voz);                   %Recorte de la señal
26 vozenf=preenfasis(vozr);             %Aplica preénfasis a la señal
27 vozm=normal(vozenf);                 %Normalización
```

```
28 [b,n]=size(vozm);                    %Extracción del tamaño de la señal
29 rec=round(b/2);                      %Tamaño de cada fase
30
31
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33 %Entrenamiento de la palabra elegida%
34 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35
36 h=0;                                  %Contador %Ciclo de entrenamiento,
37 for j=1:4                             %para los hablantes que ...
       conforman el corpus
38 for i=1:10                            %y las repeticiones que se tiene ...
       de cada uno
39 palabra='madrid';                     %Definición de la palabra elegida,
40 [palabra] = direccion(j,i,palabra); %del hablante y repetición en curso
41 [voz,Fs2,NBITS2]=wavread(palabra);  %Lectura de la repetición
42 voz=pasavoz(voz);                     %Aplicación de filtro pasavoz
43 vozr=recorte(voz);                    %Recorte de la señal
44 vozenf=preenfasis(vozr);              %Aplica preénfasis a la señal
45 voz1=normal(vozenf);                  %Normalización
46
47 vozw=dtw(vozm,voz1,Fs2);              %Aplicación del alineamiento ...
       dinámico (DTW) respecto de la repetición elegida
48
49
50 [a,b]=size(vozw);                     %Extraccion del tamaño de la ...
       repetición
51 inicio=a-rec+1;                       %Indice de recorte
52 vozi=vozw(1:rec);                     %Recorte de la parte izquierda
53 vozd=vozw(inicio:end);                %Recorte de la parte derecha
54
55
56
57 vozenerg1=ventanasmod1(vozi,Fs2);    %Segmentación, enventanado y ...
       obtención de la energía por trama de la parte izquierda
```

```
58 vozenerg2=ventanasmod1(vozd,Fs2);   %Segmentación, enventanado y ...
       obtención de la energía por trama de la parte derecha
59
60
61
62
63
64
65
66 [Cet1]=cuantificador(vozenerg1);                %la función entrega ...
       un vector cuantificado de la energia por trama de la parte izquierda
67 [Cet2]=cuantificador(vozenerg2);                %la función entrega ...
       un vector cuantificado de la energia por trama de la parte derecha
68
69
70 Edetramaiz(i+h,:)=Cet1(:,1)';                   %Almacenamiento en ...
       la matriz de símbolos parte izquierda
71 Edetramade(i+h,:)=Cet2(:,1)';                   %Almacenamiento en ...
       la matriz de símbolos parte derecha
72
73 end
74 h=h+10;
75 end
76
77 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
78 %Entrenamiento con del modelo de Markov%
79 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
80
81
82 [a,b]=size(Edetramaiz); ...
                                                  %Creación de las ...
       matrices para entrenamiento
83 A=(diag(ones(1,b),0)+ ...
       diag(ones(1,b-1),1)+diag(ones(1,b-2),2))*(1/3);    %Matriz ...
       cuadrada de probabilidad de transición equiprobable
```

```
84 A(b,b)=1;
85 A(b-1,b)=0.5;
86 A(b-1,b-1)=0.5;
87 B=(ones(b,30))*(1/30); ...
                                                    %Matriz de ...
      probabilidad de generación de símbolos equiprobable
88 inicio=zeros(1,b)';
89 inicio(1,1)=1; ...
                                                          %Matriz ...
      de probabilidad de inicio
90 [l1i,mpiz,maiz,mbiz]=learn_dhmm(Edetramaiz,inicio,A,B,50); ...
                  %Función de entrenamiento de la parte izquierda
91 [l1d,mpde,made,mbde]=learn_dhmm(Edetramade,inicio,A,B,50); ...
                  %Función de entrenamiento de la parte derecha
92
93 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94 %Prueba del sistema%
95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96
97 mult=0;
98  for j=5 ...
                                                                    ...
       %Prueba del hablante j,
99 for i=7 ...
                                                                    ...
      %repetición i y
100   palabra='madrid'; ...
                                                          %de la ...
        palabra seleccionada
101  [palabra] = direccion(j,i,palabra); ...
                                          %Repetición para el alineamiento
102 [vozorig,Fs2,NBITS2]=wavread(palabra); ...
                                       %Lectura de la repetición
103 voz1=vozorig;
```

```
104 vozp=pasavoz(voz1); ...
                                                        %Aplicación ...
      de filtro pasavoz
105 vozr=recorte(vozp); ...
                                                        %Recorte de ...
      la señal
106 vozenf=preenfasis(vozr); ...
                                                    %Aplica ...
      preénfasis a la señal
107 vozp=normal(vozenf); ...
                                                      %Normalización
108
109 vozw=dtw(vozm,vozp,Fs2); ...
                                                    %Aplicación del ...
      alineamiento dinámico (DTW) respecto de la repetición elegida
110
111 [a,b]=size(vozw); ...
                                                           ...
      %Extraccion del tamaño de la repetición
112 inicio=a-rec+1; ...
                                                          %Indice ...
      de recorte
113 vozip=vozw(1:rec); ...
                                                       %Recorte de ...
      la parte izquierda
114 vozdp=vozw(inicio:end); ...
                                                     %Recorte de la ...
      parte derecha
115
116
117 vozenerg1p=ventanasmod1(vozip,Fs2); ...
                                            %Segmentación, enventanado y ...
      obtención de la energía por trama de la parte izquierda
```

```
118 vozenerg2p=ventanasmod1(vozdp,Fs2); ...
                                          %Segmentación, enventanado y ...
    obtención de la energía por trama de la parte derecha
119
120
121
122
123 [Ceti]=cuantificador(vozenerg1p);                %la función entrega ...
    un vector cuantificado de la energia por trama de la parte izquierda
124 [Cetd]=cuantificador(vozenerg2p);               %la función entrega ...
    un vector cuantificado de la energia por trama de la parte derecha
125
126 pIndesicioniz=logo(Ceti',mpiz,maiz,mbiz);       %Cálculo del ...
    logaritmo de probabilidad de observar dicha secuencia (parte ...
    izquierda)
127 pIndesicionde=logo(Cetd',mpde,made,mbde);       %Cálculo del ...
    logaritmo de probabilidad de observar dicha secuencia (parte derecha)
128
129         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
130         %Prueba de dicción en base a umbrales%
131         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
132
133         %En base a la dicción presente en la muestra de voz a probar se
134         %despliega el resultado encontrado en base al umbral
135
136
137     if abs(pIndesicioniz)<100 && abs(pIndesicionde)<100
138             fprintf('Excelente Dicción');
139             color=1;
140             despliegue(voz1,Cet2,vozorig,color)
141         else if abs(pIndesicioniz)<150 && abs(pIndesicionde)<150
142                 fprintf('Buena Dicción');
143                 color=2;
144                 despliegue(voz1,Cet2,vozorig,color)
145             else
```

```
146                 fprintf('Mala Dicción,repita el ejercicio')
147             end
148     end
149 end
150 mult=mult+10;                                  %El proceso continua ...
       hasta terminar con las muestras a probar
151  end
152 j=0;
```

B.11 Modelos Ocultos de Markov con relleno de palabras (MRP).

```
1
2  %Modelos Ocultos de Markov con relleno de palabras (MRP)%
3
4  %Este algoritmo realiza el reconocimiento de la palabra pronuniada ...
       en base a un relleno de palabras
5  %y muestra el resultado del análisis en base a su dicción.
6
7  %%%%%%%%%%%%%%%%%%%%%%
8  %Inicio del programa%
9  %%%%%%%%%%%%%%%%%%%%%%
10 clear all             %Borrar variables
11 clc                   %Limpiar pantalla
12 tbase=0;              %inicializar tamaño de la base en cero
13 h=0;                  %contador en cero
14
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 %Entrenamiento de la palabra elegida%
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19 for j=1:4                                %Ciclo de entrenamiento,
20 for i=1:10                               %para los hablantes que ...
       conforman el corpus y las repeticiones que se tiene de cada uno
21 palabra='madrid';                       %Elección de la palabra
22 [palabra] = direccion(j,i,palabra);     %del hablante y repetición ...
       en curso
23 [voz,Fs2,NBITS2]=wavread(palabra);      %Lectura de la repetición
24 voz=pasavoz(voz);                       %Aplicación de filtro pasavoz
25 vozr=recorte(voz);                      %Recorte de la señal
26 vozenf=preenfasis(vozr);                %Aplica preénfasis a la señal
27 voz1=normal(vozenf);                    %Normalización
28 vozenerg=ventanasmod1(voz1,Fs2);        %Segmentación, enventanado y ...
       obtención de la energía por trama
```

```
29 [Cet]=cuantificador(vozenerg);          %Entrega un vector ...
      cuantificado de la energia por trama
30 t=size(Cet);                            %Tamaño del vector obtenido
31 if tbase==0                             %Si es el primer vector ...
      (repetición)
32     tbase=t(1);                         %Lo coloca con el mismo tamaño
33 else                                    %si no es la primera ...
      repetición y
34     if tbase>t(1)                       %el tamaño de la base ...
          (vectotr almacenado) es mayor a la entrante (vector entrante)
35         Cet=Cet';                       %ajuste del vector
36         RBT1=tbase-t(1);                %calculo de la diferencia
37         Cet=[Cet,ones(1,RBT1)];         %ajuste del nuevo vector en ...
              base al tamaño almacenado
38         Cet=Cet';                       %ajsute del vector
39     else if tbase<t(1)                  %o si el tamaño de la base ...
          (vector almacenado) es menor que el entrante
40         RBT1=t(1)-tbase;                %calculo de la diferencia
41         [filas,columnas]=size(Edetrama);% tamaño de la matriz almacenada
42         Edetrama=[Edetrama,ones(filas,RBT1)]; %se rellena la matriz ...
              en base al nuevo tamaño
43         tbase=columnas+RBT1;                  %se ajusta el indice ...
              del tamaño
44     else
45
46         end
47     end
48 end
49 Edetrama(i+h,:)=Cet(:,1)';              %Almacenamiento en la matriz ...
      de símbolos
50
51
52 end
53 h=h+10;
54 end
```

```
55
56 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
57 %Entrenamiento con del modelo de Markov%
58 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59 [a,b]=size(Edetrama); ...
                                                    %Creación de ...
      las matrices para entrenamiento
60 A=(diag(ones(1,b),0)+ ...
      diag(ones(1,b-1),1)+diag(ones(1,b-2),2))*(1/3);     %Matriz ...
      cuadrada de probabilidad de transición equiprobable
61 A(b,b)=1;
62 A(b-1,b)=0.5;
63 A(b-1,b-1)=0.5;
64 B=(ones(b,30))*(1/30); ...
                                                  %Matriz de ...
      probabilidad de generación de símbolos equiprobable
65 inicio=zeros(1,b)';
66 inicio(1,1)=1; ...
                                                          ...
      %Matriz de probabilidad de inicio
67 [l1,mpin,main,mbin]=learn_dhmm(Edetrama,inicio,A,B,50); ...
                          %Función de entrenamiento
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Prueba del ...
      sistema%%%%%%%%%%%%%%%%%%%%%%%%
69
70 mult=0;
71  for j=5 ...
                                                              ...
       %Prueba del hablante j,
72 for i=1:10 ...
                                                            ...
      %repetición i y
73   palabra='madrid'; ...
                                                      %de la ...
        palabra seleccionada
```

```
74  [palabra] = direccion(j,i,palabra); ...
                                          %Repetición para el ...
       alineamiento
75 [vozorig,Fs2,NBITS2]=wavread(palabra); ...
                                       %Lectura de la repetición
76  voz1=vozorig;
77 voz1=pasavoz(voz1); ...
                                                    %Aplicación ...
      de filtro pasavoz
78 vozr=recorte(voz1); ...
                                                    %Recorte de ...
      la señal
79 vozenf=preenfasis(vozr); ...
                                                 %Aplica ...
      preénfasis a la señal
80 voz1=normal(vozenf); ...
                                                       %Normalización
81 vozenergia2=ventanasmod1(voz1,Fs2); ...
                                          %Segmentación, enventanado ...
      y obtención de la energía por trama
82 vozenergia2=normal(vozenergia2); ...
                                             %Normalización
83 [Cet2]=(cuantificador(vozenergia2))';              %Entrega un ...
      vector cuantificado de la energia por trama
84 orcentaje=logo(Cet2,mpin,main,mbin);               %Cálculo del ...
      logaritmo de probabilidad de observar dicha secuencia
85
86         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
87          %Prueba de dicción en base a umbrales%
88          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
89
90          %En base a la dicción presente en la muestra de voz a probar se
91          %despliega el resultado encontrado en base al umbral
92
93          if abs(orcentaje)<150
```

```
94              fprintf('Excelente Dicción');
95              color=1;
96              despliegue(voz1,Cet2,vozorig,color)
97          else if abs(orcentaje)<300
98                  fprintf('Buena Dicción');
99                  color=2;
100                 despliegue(voz1,Cet2,vozorig,color)
101             else
102                 fprintf('Mala Dicción,repita el ejercicio')
103             end
104         end
105
106
107
108 end
109 mult=mult+10;                                   %El proceso continua ...
       hasta terminar con las muestras a probar
110 end
```

Referencias

[1] C. Seelbach, "A perspective on early commercial applications of voice-processing technology for telecommunications and aids for handicapped", En: Human-Machine Communication by Voice, Irvine, CA, USA, 8-9 de Febrero de 1993, pp. 9989-9990, Proc. Natl. Acad. Sci., 1995.

[2] B. Plínio, "On the Defense of von Kempelen as the Predecessor of Experimental Phonetics and Speech Synthesis Research", En: The Ninth International Conference on the History of the Language Sciences, São Paulo−Campinas, 27 al 30 de Agosto de 2002, pp. 101−106, John Benjamins Publishing Company, 2007.

[3] E. David, O. Selfridge, "Eyes and Ears for Computers", *Proceedings of the IEEE*, Vol.50, Mayo 1962, pp. 1093−1101.

[4] B. Gold, N. Morgan, D. Ellis, *Speech and audio signal processing: Processing and Perception of Speech and Music*, 2da. edición, editorial WILEY, 2011, 688 páginas.

[5] R. Esparza, "Cómo funciona Siri", Revista Como funciona: Edición México, No.04, 2014, p. 45.

[6] C. Cristían, *La voz hablada y cantada*, 8va. edición, editorial EDAMEX, 1994, 257 páginas.

[7] El correo de la UNESCO, "¿Quién habla qué?", Guerra y paz en el frente de las lenguas, No.04, 2000, pp. 20-21.

[8] L. Rabiner, B. Juang, *Fundamentals of speech recognition*, editorial Prentice Hall, 1993, 507 páginas.

[9] CRIBEO [en línea]. Barcelona (España), [fecha de consulta: 26 de junio de 2014]. Disponible en: `http://www.cribeo.com/ocio y cultura/1004/las-10-palabras-mas-del-espanol`.

[10] H. Silva, Reconocimiento Automático de locutor y realización de un sistema experimental, Tesis de Maestría, Departamento de Electrónica y Telecomunicaciones, Centro de investigación científica y de educación superior de Ensenada, 1994.

[11] A. Ramírez, Reconocimiento automático del locutor mediante técnicas dependientes e independientes del vocabulario para un sistema acotado por el ancho del banda telefónico y realización de us sistema experimental , Tesis de Maestría, Departamento de Electrónica y Telecomunicaciones, Centro de investigación científica y de educación superior de Ensenada, 1996.

[12] J. Varela, J. Loaiza, Reconocimiento de palabras aisladas mediante redes neuronales sobre FPGA, Tesis de licenciatura, Facultad de Ingenierías: Eléctrica, Electrónica, Física y de Sistemas, Universidad Tecnológica de Pereira, 2008.

[13] S. Rascón, Reconocimiento de voz para un control de acceso mediante una red neuronal de retropropagación, Tesis de licenciatura, Escuela superior de ingeniería mecánica y eléctrica Unidad Culhuacan, México, D.F., 2009.

[14] G. Bendezú, Filtro adaptivo LMS y su aplicación en el reconocimiento de palabra aisladas para el control de equipo de sonido por medio de la voz, Tesis de licenciatura, Pontificia Universidad Católica del Perú: Facultad de Ciencias e Ingeniería, Lima, Perú, 2004.

[15] J. Pech, Desarrollo de un sistema de reconocimiento de voz para el control de dispositivos utilizando mixturas gaussianas, Tesis de Maestría, Instituto Politécnico Nacional, Centro de Investigación en Computación, México, D.F, 2006.

[16] G. Velásquez, Sistema de reconocimiento de voz en Matlab, Tesis de Licenciatura, Universidad de San Carlos de Guatemala: Facultad de Ingeniería, Guatemala, 2008.

[17] A. Cano, Codificación vectorial de voz en español, Tesis de Maestría, Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, México, 2003.

[18] F. Bellesi, F. Ortiz, F. Poblete, M. González, *Reconocimiento de voz para la aplicación en domótica*, Universidad Tecnológica Nacional: Facultad Regional San Nicolás, 2008.

[19] J. Rodríguez, Sistema de reconocimiento del locutor basado en modelado no paramétrico, Tesis de Maestría, Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, México, D.F, 2008.

[20] L. Beltrán, Simulación de modelos ocultos de markov aplicados al reconocimiento de palabras aisladas, utilizando el programa Matlab, Tesis de Licenciatura, Escuela Politécnica Nacional: Escuela de Ingeniería, Quito, 2003.

[21] G. Pérez, Herramientas de Segmentación y Evaluación de Series Temporales Basadas en Modelos Ocultos de Markov, Tesis de Licenciatura, Universidad Carlos III de Madrid, España, Madrid, 2010.

[22] S. Prasad, T. Kishore, "Hybrid HMM/DTW based Speech Recognition with Kernel Adaptive Filtering Method", International Journal on Computational Sciences & Applications, Vol.1, No.4, 2014.

[23] S. Furui, "Recent Advances in Spontaneous Speech Recognition and Understanding", En: Spontaneous Speech Processing and Recognition, Tokyo, Japan, 13-16 de Abril de 2003, ISCA & IEEE Workshop, 2003.

[24] G. de las Heras, L. Rodríguez, *Materiales para cuidar mi voz*, Fundación MAPFRE-UCLM, 44 páginas.

[25] J. Flores, Técnicas para el reconocimiento de voz en palabras aisladas en la lengua náhuatl, Tesis de Maestría, Instituto Politécnico Nacional, Centro de Investigación en Computación, México, D.F, 2009.

[26] D. Ellis, Dynamic Time Warp (DTW) in Matlab [Algoritmo Computacional], (Fecha de consulta Mayo, 2015), [Online] `http://www.ee.columbia.edu/ln/rosa/matlab/dtw/`.

[27] K. Murphy, Hidden Markov Model (HMM) Toolbox for Matlab [Algoritmo Computacional], (Fecha de consulta Mayo 2015), [Online] `http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`.

[28] D. Barragán, *Manual de Interfaz Gráfica de Usuario en Matlab Parte 1*, 2008, 74 páginas.

[29] L. Rabiner, "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE. Vol. 77. No. 2. Febrero de 1989. pp. 257-286.

[30] A. Buzó, A. Gray, R. Gray, J. Markel, "Speech Coding Based Upon Vector Quantization", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. assp-28, No. 5, Octubre de 1980, pp. 562-574.