UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
ELÉCTRICA – PROCESAMIENTO DIGITAL DE SEÑALES

ACÚSTICA FORENSE: IDENTIFICACIÓN DE LOCUTORES AUTOMATIZADA

TESIS QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
ING. EDUARDO GARCÍA LETECHIPIA

TUTOR PRINCIPAL
DRA. LUCIA MEDINA GOMEZ, FACULTAD DE CIENCIAS

MÉXICO, D. F. MAYO 2014

JURADO ASIGNADO:

Presidente:	DR. JESÚS SAVAGE CARMONA
Secretario:	DR. FELIPE ORDUÑA BUSTAMANTE
Vocal:	DRA. LUCIA MEDINA GOMEZ
1er. Suplente:	DR. VÍCTOR GARCÍA GARDUÑO
2do. Suplente:	DR. CARLOS RIVERA RIVERA

Lugar o lugares donde se realizó la tesis: México, D.F.

TUTOR DE TESIS:

DRA. LUCIA MEDINA GOMEZ

FIRMA

AGRADECIMIENTOS

La presente tesis está dedicada a mi madre, quien desgraciadamente no pudo estar a mi lado para ver su conclusión y presentación, pero cuyas palabras de amor y de apoyo, siguen y seguirán resonando en mi cabeza por el resto de mi vida, a tus enseñanzas y consejos guiaré a mi hijo y transmitiré tu gran amor hacia él, que en paz descanses y que Dios guíe tu andar, nos vemos de nuevo un poco más adelante en el camino.

También quiero agradecer a mi familia, mi esposa, mi hijo, mi padre y mis hermanos por su apoyo incondicional, a mis maestros y tutores, ya que sin su invaluable ayuda este trabajo no hubiese sido posible y a mis amigos que me apoyaron en este largo camino del aprendizaje.

TABLA DE CONTENIDO.

TABLA DE FIGURAS.

LISTA DE TABLAS.

INTRODUCCIÓN.

Como en casi toda rama del saber humano y como lo planteo el famoso filosofo Sócrates, cuanto más aprendemos sobre un tema solo nos damos cuenta de cuan poco sabemos y de todo lo que queda por aprender.

En esta línea de pensamiento y dando continuidad al trabajo iniciado en mi tesis de licenciatura "Acústica Forense", ésta podría ser considerada una segunda parte, una nueva aproximación al problema del reconocimiento automatizado de locutores con la meta de aumentar la efectividad lograda bajo condiciones de trabajo más reales, como lo veremos a lo largo de estas páginas.

Para el sistema que se desarrolla y prueba en esta tesis, decidí cambiar el nombre de "Acústica Forense 1.0" a ***akutzika 2.0***, esto obedeciendo mi admiración y respeto por el pueblo Maya, pero también buscando otorgar un nombre más significativo de su función, de la misma manera cambie el logotipo del sistema al icono-grama Maya de idéntico significado, akutzika o acústica en Maya (Pitts, 2008).

Para este punto pueden ya haber surgido algunas dudas, ¿Cómo evaluaremos el sistema?, ¿Cómo sabemos si este sistema es mejor o peor que sus similares?, ¿Existe una sola opción de configuración?, si existe más de una opción ¿Cuál es la mejor?

En el desarrollo de la tesis aclararemos todas estas dudas, pero de manera resumida podemos mencionar que todo sistema biométrico es evaluado mediante un corpus de evaluación, que no es sino una colección de grabaciones de voz para nuestro caso; estas grabaciones contienen texto libre, en diferentes idiomas en algunas ocasiones y en un número no fijo de grabaciones distintas por cada locutor, podemos tener para un locutor dos grabaciones en un mismo idioma y para otro 5 o más grabaciones en distintos idiomas.

Aquí empleare una parte del corpus creado para las pruebas realizadas por el NIST (National Institute of Standards and Technology) en 2008 que consta de 1,000 grabaciones de 174 locutores distintos de ambos géneros con las siguientes características:

- ✓ Identificación con 120 o más segundos de habla continua por locutor
- ✓ Entrenamiento con 120 o más segundos de habla por locutor
- ✓ Grabaciones en un ambiente sin ruido (mínimo 20 dB de SNR)
- ✓ Independencia del texto o contenido de la grabación
- ✓ Independiente del idioma

Una vez que sabemos con qué se evalúa, queda aún la duda de cómo se hace, esto será realizado mediante lo que se considera como el punto óptimo de operación de cualquier sistema biométrico, el EER o punto de error igual, que es aquel ajuste donde el sistema cometerá el mismo porcentaje de errores de falsa aceptación y de falso rechazo.

Con esto vemos que no hay un sólo ajuste o punto de operación, aunado a esto, existen múltiples opciones de configuración para un sistema de este tipo, más adelante veremos qué

parámetros se pueden modificar y cómo afecta esto a la eficiencia de la solución, de esto la principal aportación de esta tesis será abordar cómo encontrar esta configuración ideal.

Existen muchos tipos de metodologías, pero aún dentro de una metodología concreta hay muchas variantes, tamaños de las ventanas de análisis, tipos de ventanas matemáticas, número de coeficientes, parámetros de filtrado y mucho más que puede llevar a miles de combinaciones.

Esto podría no parecer muy problemático, pero dado que usaremos 1,000 grabaciones para cada prueba esto significa 500,000 comparativos biométricos y mil modelos matemáticos que significan billones de cálculos, en este sentido si dijéramos que una prueba toma 2 horas de proceso y requerimos probar algunas decenas de configuraciones distintas esto podría significar muchas horas, días o meses de procesamiento.

Se verá que encontrar la configuración ideal no es una tarea fácil, como tampoco lo es el diseño de la solución, la voz humana contiene mucha información, por esto la correcta parametrización no es una tarea simple, aquí construiré un sistema que permita enrolar una base de datos de pruebas y posteriormente nos permita lanzar búsquedas biométricas sobre la misma para identificar a un locutor en pruebas de set abierto.

Se dará énfasis por supuesto a la parte de la biometría, dejando detalles como la base de datos o la interface gráfica con la mínima atención posible por el momento, después de todo es más interesante una herramienta poco atractiva pero funcional que una interface de usuario hermosa con un motor poco eficiente detrás.

El sistema que aquí se propone logró un EER de 14.898%, esto significa que por cada 100 identificaciones se cometerían aproximadamente 15 errores, esto de entrada puede no parecer bueno, sin embargo esto coloca el sistema dentro de los primeros 30 lugares si hubiera participado en las pruebas NIST 2012.

ale la pena mencionar que ésta no es una prueba destinada a estudiantes, éste es un evento donde compañías y universidades de todo el mundo compiten, por esto quedar aproximadamente a la mitad de la lista para una primera aproximación en texto libre no es nada despreciable.

1. OBJETIVO DE LA TESIS

El objetivo de esta tesis gira alrededor de dos grandes puntos, el primero es crear un motor de identificación biométrica por voz, el segundo es afinar este motor para obtener el mejor rendimiento; como veremos a través del desarrollo de la tesis estas tareas no son simples, por este motivo se irán desarrollando de la manera más clara posible todos los temas que a mi parecer son necesarios para su correcto entendimiento.

Comenzaremos dando un recorrido por qué es la biometría por voz, esto es básicamente el cómo se puede discriminar entre diferentes muestras de voz de un mismo locutor y aquellas muestras de distintos locutores, esto es el cómo explotar la información que tenemos, que es una o más grabaciones con voz.

La minería de datos, que es un tema que abordare, es la extracción de inteligencia que nos permitan explotar la información disponible, para comprender cómo hacer este proceso, veré los antecedentes históricos de la materia para tener una mejor idea de que se ha hecho y que se está haciendo actualmente en el estado del arte.

También prestare especial atención a cómo medir la eficiencia de un sistema de esta índole, ya que esta medición será uno de los puntos primordiales que nos auxiliaran en la evaluación del sistema desarrollado, mencionado esto pasare al primer objetivo, la propuesta inicial del sistema.

1.1 La creación de un motor biométrico

Mi primer objetivo concreto es crear un sistema que permita la identificación automatizada del parlante, ésta no es quizás una idea innovadora, existen en internet múltiples tutoriales que nos pueden guiar, incluso existen proyectos que podemos descargar de manera gratuita, pero la clave es la eficiencia que se logra, las técnicas son bien conocidas de manera general, pero el robustecimiento del sistema para obtener buenos resultados no es simple.

Ahora, no se puede robustecer lo que no existe, como el Dr. Ricardo Hirata [Keisen consultores] lo plantea, “lo que no se mide, no se controla y no se puede mejorar”, no se puede generar un sistema que logre resultados competitivos de un tutorial, se debe iniciar por crear un sistema versátil que permita refinar su funcionamiento y configuración, las opciones de configuración son muy amplias como se verá más adelante.

La identificación de locutores presenta retos importantes, la principal dificultad deriva de la naturaleza cambiante de la voz [Sadaoki Furui, 2005], además de esta dificultad inherente, no existe documentación que nos indique cuál es el mejor camino o método a seguir, cada grupo de investigación toma un camino, lo desarrolla y ofrece sus resultados en el mejor de los casos, aquí detallare paso a paso la creación del motor biométrico y sus pruebas.

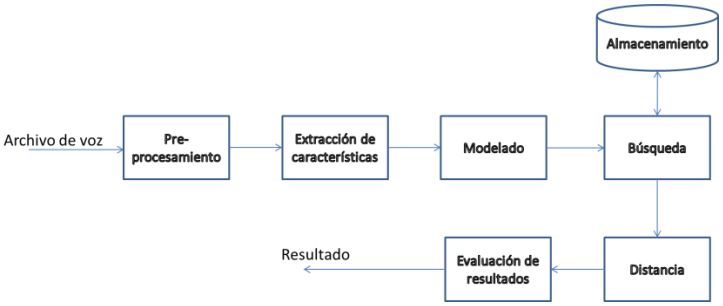


Figura 1.1: Diagrama de un sistema básico

En la figura 1.1 tenemos el diagrama de bloques de un sistema básico de identificación de locutores, un módulo de pre-procesamiento, un módulo que extrae las características del locutor, un módulo que modela al locutor en un objeto comparable, el resto corresponde al almacenamiento, comparación y desplegado de resultados.

En la figura 1.2 apreciamos un diagrama de bloques del motor biométrico hasta la creación del modelo, esta propuesta que se desarrolló se conforma de un detector de actividad vocal, un módulo de pre-énfasis, un proceso de ventaneo, una transformación de Fourier, una extracción de la energía de la señal, un banco de filtros, una escala logarítmica, una segunda transformación de Fourier, un extractor de coeficientes polinomiales y finalmente un modelado.

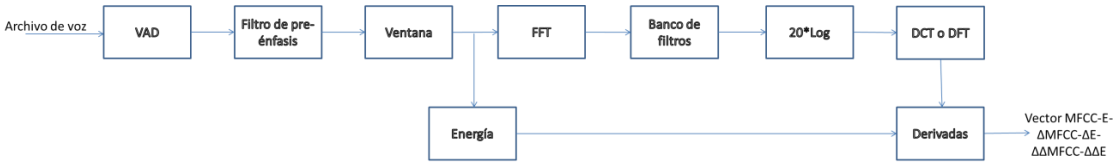


Figura 2.2: Diagrama de bloques de un motor biométrico

Como veremos existen múltiples parámetros en cada módulo que afectara el desempeño del sistema en una muy buena medida, en este caso en particular la diferencia entre la mejor y la peor configuración es superior al doble de la mejor eficiencia lograda, lo cual sería la diferencia entre un sistema que comete menos de 15 errores por cada 100 identificaciones contra uno con más de 39 errores por las mismas 100 identificaciones.

Los módulos implementados fueron los siguientes:

- Módulo de detección de actividad vocal
- Módulo de filtrado digital de pre-énfasis
- Módulo de ventaneo
- Módulo de extracción de energía
- Módulo de transformación al dominio de la frecuencia
- Módulo de banco de filtros
- Módulo de transformación cepstral
- Módulo de aproximación polinomial
- Módulo de ensamble de los coeficientes
- Módulo de creación de modelos
- Módulo de estimación de distancias
- Módulo de manejo estadístico de los resultados

1.2 Configuración del motor biométrico

Con la creación del motor biométrico cumplo el primer objetivo, ahora resta configurarlo para que trabaje de la mejor manera posible, como ya se vio, existen un gran número de parámetros que afectaran al sistema y su eficiencia; la eficiencia del sistema se puede ver disminuida en más del doble si no lo configuramos correctamente, pero para comprender mejor esto explicare como se evalúa la efectividad de un sistema biométrico.

En la tarea de detección de eventos [Yin, S.-C., Rose, R. and Kenny, P., 2007], se aprecian dos tipos comunes de error, el sistema puede dar una falsa alarma o una detección fallida. La forma adecuada para presentar el funcionamiento del sistema es generar una curva para ver el comportamiento en los diferentes puntos de operación. El punto de operación de un sistema es un parámetro que puede ser ajustado dentro de éste para llevar a cabo la toma de decisiones.

En los sistemas de reconocimiento de identidad, como los que trabajan con voz, retina o huella dactilar, se utiliza un mismo método que evalúa las dos posibles decisiones: aceptar o rechazar al locutor o sujeto que se está identificando, para tal fin se pueden presentar cuatro posibles resultados, dos correctos y dos errados:

- Aceptar un locutor registrado.
- Rechazar un impostor.
- Aceptar un impostor.
- Rechazar un locutor registrado.

Estos errores corresponden, respectivamente, a los comúnmente denominados errores de falsa aceptación y de falso rechazo. La tasa donde estos errores se igualan es conocida como EER (Equal Error Rate), que es comúnmente utilizada como medida de desempeño en los sistemas biométricos, el sistema cometerá el mismo número de errores de falsa aceptación y de falso rechazo, por esto se le determina como el punto óptimo de calibración y su punto de mejor desempeño.

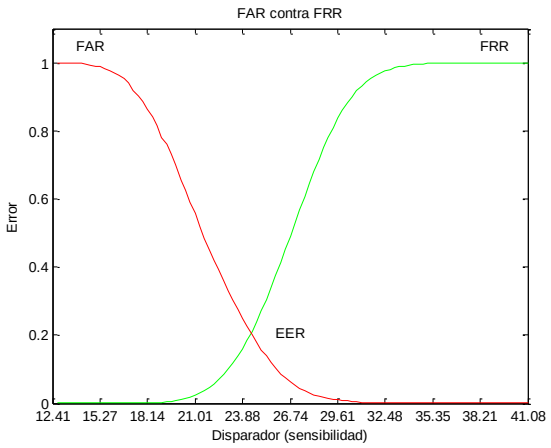


Figura 3.3: Curva FRR vs. FAR

Es importante mencionar que todo sistema posee un umbral de decisión, si éste es demasiado estricto tenderá a rechazar clientes y si es demasiado permisivo tenderá a aceptar impostores, por lo que para optimizar el desempeño del sistema se deben minimizar conjuntamente ambos

tipos de errores. Para encontrar el mejor umbral de decisión se emplean las curvas antes descritas ya que es simplemente la intersección de ambas curvas.

La curva de Falso Rechazo se construye moviendo el umbral de decisión en el rango de clasificación del sistema, identificando para cada uno de estos puntos, qué porcentaje de las ocasiones el cliente trató de verificarse y fue rechazado. Por su parte, la curva de falsa aceptación se obtiene desplazando el umbral de decisión e identificando para cada punto, qué porcentaje de los impostores fue aceptado. La intersección de estas curvas indica el umbral óptimo para el cual se minimiza el error del sistema.

Como una forma alternativa de medir la eficiencia de un sistema, tenemos por ejemplo las gráficas ROC y DET, éstas se construyen a partir de esta misma información, sólo varia su interpretación, escalas o que información va en cada eje, ofreciendo maneras alternativas de visualizar el comportamiento, pero en general el EER será el parámetro que empleare en esta tesis para medir la eficiencia de la solución.

La curva ROC (Receiver Operating Characteristics) ha sido usada tradicionalmente para ajustar el punto de operación de un sistema o para decir cuál es el mejor sistema en un proceso de toma de decisiones. Generalmente se grafica la tasa de falsas alarmas en el eje horizontal, mientras la detección correcta se grafica en el eje vertical.

La curva DET (Detection Error Tradeoff) ha sido desarrollada para apreciar el funcionamiento de un sistema de manera más sencilla debido a que se grafica la desviación normal en ambos ejes, dando un tratamiento uniforme para ambos tipos de error con una escala logarítmica en ambos ejes, lo anterior mejora la forma de observar el funcionamiento de los sistemas al generar gráficas que son más cercanas a líneas rectas.

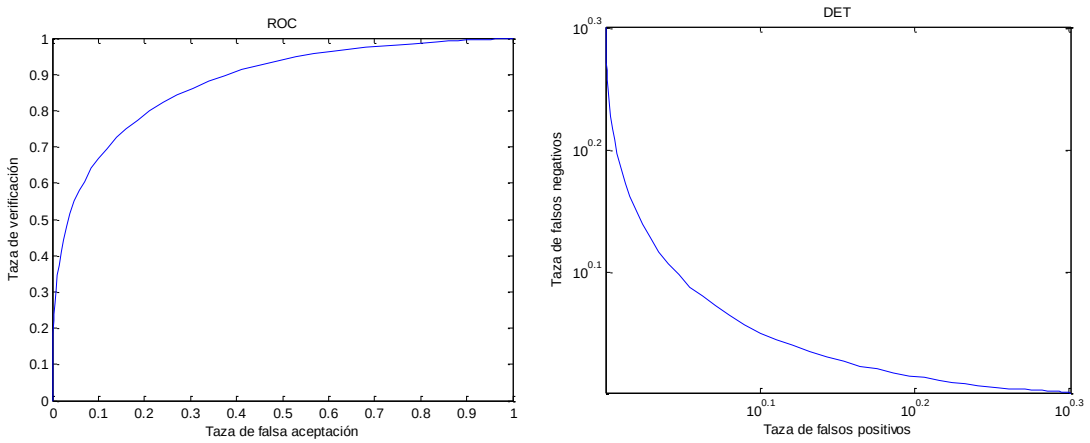


Figura 4.4: Curvas ROC y DET

En cuanto a los parámetros que podemos ajustar tenemos los siguientes:

- **Tamaño de la ventana para Fourier:** Este parámetro afecta la resolución temporal y de frecuencia que se obtiene, puede tomar múltiples valores en base a 2^n.
- **Traslape de la ventana de Fourier:** Las ventanas de cálculo se pueden traslapar para mejorar la representación de la información, esto puede ir de 0 a 99%.

- **Tipo de filtro para Fourier:** Aunado al traslape y para mejorar el efecto adverso que se crea en los bordes de la ventana, tenemos la implementación de un filtro que suavice los bordes de la ventana, este filtro puede ser uno de 17 tipos definidos en MATLAB.
- **Valor de alfa para pre-énfasis:** Este filtro puede ser opcional, aunado a esto el valor de alfa esta entre 0 y 1.
- **Número de coeficientes MFCC:** Este es un parámetro libre, existen por supuesto recomendaciones, pero en general puede ir de alrededor de 10 a 50 o más coeficientes.
- **Traslape de los filtros:** Los filtros para obtener la envolvente pueden tener un traslape entre filtros contiguos, este parámetro es libre en valores de 0 al 99%.
- **Tipo de filtro para el MFCC:** Al igual que en la parte de la transformada de Fourier se puede elegir además del traslape, la forma de los filtros.
- **Frecuencias de corte:** Aunado al banco de filtros se puede colocar un filtro paso banda que elimine partes no deseadas del espectro.
- **Reducción y centrado de vectores:** Se pueden reducir y/o centrar los vectores de los coeficientes cepstrales de manera opcional.
- **Orden de aproximación para las deltas:** Dado que los deltas son calculados por una aproximación polinomial se puede elegir el orden de dicho polinomio.
- **Armado del modelo:** El conjunto de datos que conforma al modelo puede o no incluir los coeficientes Δ, $\Delta\Delta$ y la energía además de los coeficientes cepstrales.
- **Tipo de transformación:** Este parámetro indica el tipo de escala a utilizar para el banco de filtros (escala lineal, Mel o Bark).
- **Transformación secundaria:** Para la obtención del cepstrum se requiere realizar una segunda transformación que puede ser DFT o DCT.
- **Tamaño del modelo por cuantización vectorial:** Este dato se refiere al número de vectores que representara o modelara al locutor.
- **Aplicar detector de actividad vocal:** Este como el pre-énfasis es opcional, su función es eliminar las pausas o secciones sin habla.

De esta lista es claro que pueden existir cientos o miles de posibles configuraciones, por supuesto que existen recomendaciones, pero no son consistentes, varían de acuerdo al autor, existen también reglas o razonamientos que podemos emplear, por ejemplo, si tomamos una ventana pequeña para la transformada de Fourier tendremos una mala resolución en frecuencia, si por el contrario la tomamos muy amplia tendremos una mala resolución en el tiempo, pero la pregunta final sería ¿Cuál daría la mejor eficiencia?, ¿Está relacionada por ejemplo con el tipo de ventana?

En este sentido opte por una aproximación empírica para encontrar esa configuración ideal, más adelante se verá como cada parámetro afecta la eficiencia del sistema, viéndose algunas sorpresas con valores no esperados y otros que obedecen al razonamiento lógico, siendo ésta a mi opinión la principal aportación de esta tesis.

2. BIOMETRÍA POR VOZ

De primera instancia la biometría (del griego *bios* vida y *metron* medida) es el estudio de métodos automáticos para el reconocimiento único de humanos basados en uno o más rasgos

conductuales o físicos intrínsecos. Es la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos o de conducta de un individuo para su identificación, éste término es empleado por igual para voz, huellas dactilares, retina, iris, patrones faciales, venas de la mano, etc.

Una vez definido qué es la biometría, comencemos a ver qué parámetros o características usaremos y cómo las vamos a extraer. Es importante, previo a adentrarnos a conocer un tema que es de suma importancia y no siempre de fácil comprensión, y mucho menos es fácil el evitarlo o compensarlo, este tema es la estimación de las diferencias entre locutores y las diferencias internas en un mismo locutor, en otras palabras es la discriminación entre locutores.

2.1 Discriminación entre locutores

La distinción entre un locutor y otro, así como entre distintas grabaciones de un mismo locutor es llamada "*discriminación de locutores*" (diferenciación entre-locutores e intra-locutor).

Desde el punto de vista psicológico se puede decir de la voz que *"aunque los hablantes produzcan diferentes sonidos y los oyentes los perciban como objetivamente diferentes, no son conocedores de tal diferencia; el hablante cree producir el mismo sonido, y el oyente la impresión de oír el mismo sonido"*.

De esta definición y mediante las gráficas siguientes podemos ejemplificar dichas diferencias, para esto ponemos de primera instancia las gráficas de forma de onda de una palabra pronunciada dos veces por el mismo locutor en las mismas condiciones y con la intensión de que las iteraciones fueran iguales.

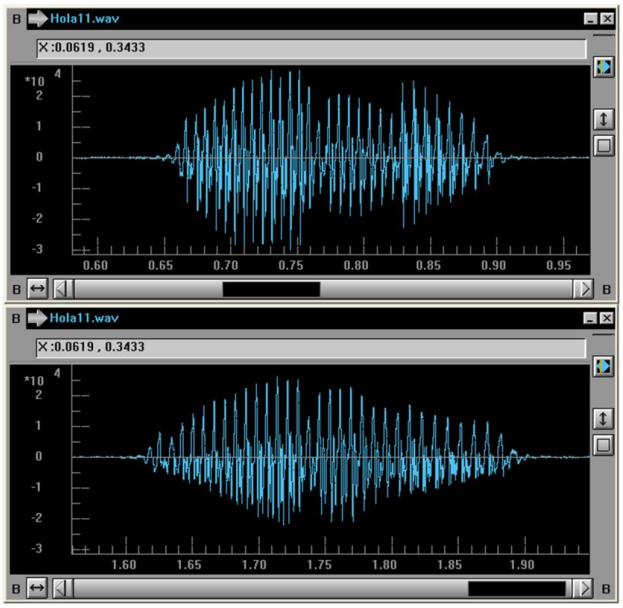


Figura 2.1: Dos iteraciones de la palabra "hola" por el mismo locutor en las mismas circunstancias

Aquí podemos ver que existen diferencias evidentes, la duración de cada pronunciación es ligeramente distinta, así como el comportamiento energético en ambas grabaciones, sin embargo también podemos observar similitudes como es de esperarse, el patrón general en ambas grabaciones es similar, de esto podemos inferir que existen diferencias entre distintas

grabaciones de una misma persona, pero al mismo tiempo se conservan también elementos suficientes como para poder decir si pertenecen a la misma persona.

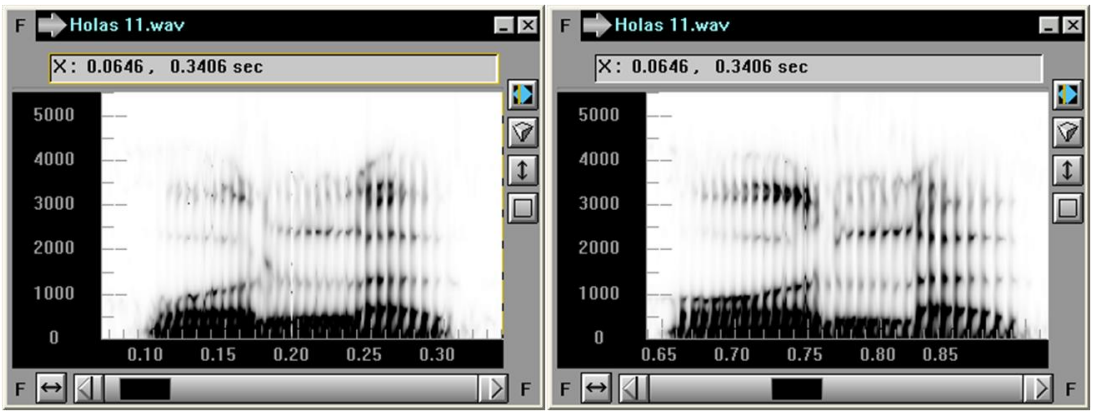


Figura 2.2: Espectrograma de las mismas palabras

En este espectrograma se pueden apreciar aún más las similitudes entre las dos grabaciones, al parecer las diferencias son mucho menores, de estas similitudes entre ambas grabaciones desde el punto de vista visual nace el método para la identificación por un experto que recibe el nombre de "Voice Print", método que a la fecha es utilizado ampliamente en el medio forense a pesar del criticismo a nivel científico por la falta de una metodología formal y escrita que describa "**cómo y qué se compara o se mide"** para establecer la identidad de una persona, pero por el momento ésto está fuera del alcance del presente documento, lo que nos interesa es que se parecen y eso es suficiente.

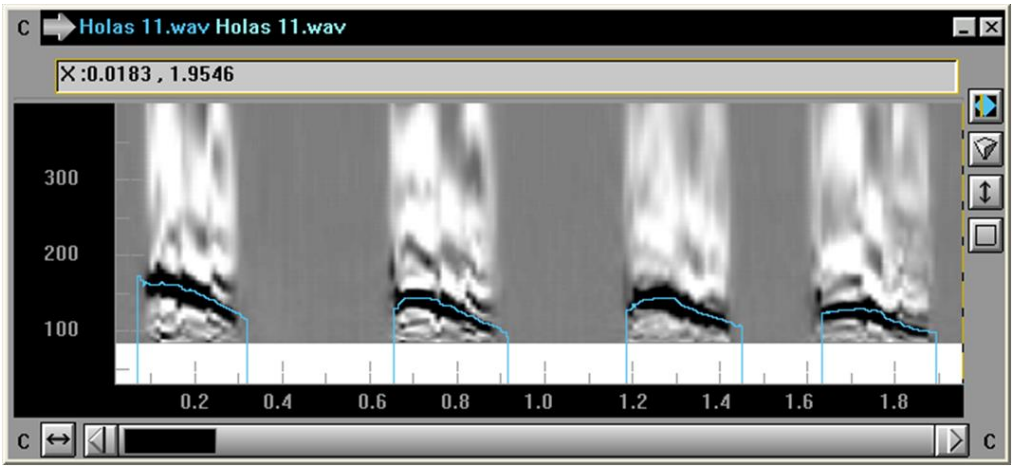


Figura 2.3: Tono fundamental de las mismas palabras

Ahora en la figura 2.3 podemos observar dos gráficas superpuestas que nos permitirán llegar a nuestro siguiente punto importante, esta gráfica muestra una auto-correlación con una gráfica del tono fundamental superpuesto, ésto nos permite corroborar por dos métodos distintos si el tono fundamental fue calculado de manera adecuada, uno es un algoritmo matemático y el otro es simplemente una representación gráfica de los datos reales, de nuevo, podemos ver que todas las gráficas, en esta ocasión cuatro iteraciones de la palabra "Hola", son similares, misma tendencia, valores similares y duración similar.

Sin embargo, al observar los datos numéricos que arroja este análisis podemos ver que la misma palabra va de 85 hasta 90 ventanas para el cálculo del tono fundamental y si obtenemos un simple histograma de estas cuatro repeticiones podemos observar lo siguiente:

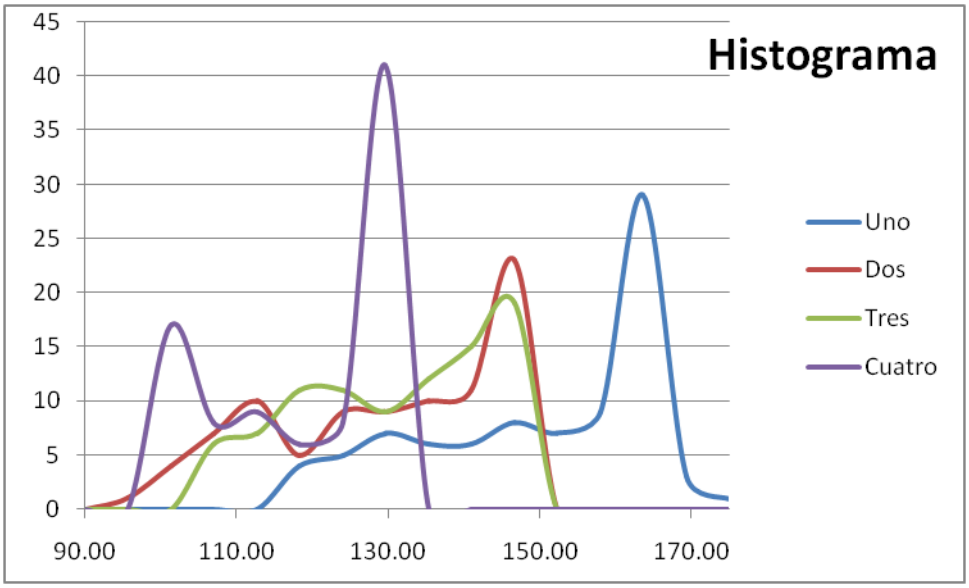


Figura 2.4: Histograma del tono fundamental de cuatro repeticiones de la palabra "Hola" por un mismo locutor

De la gráfica queda una cosa clara, lo que desde el punto de vista gráfico parece similar desde el punto de vista numérico, para una computadora puede parecer totalmente distinto, es difícil apreciar un patrón aquí, incluso es aún más difícil querer decidir si estas gráficas corresponden o no a un mismo locutor, a simple vista se podría decir que dos grabaciones son de un mismo locutor y existen dos locutores más distintos, siendo esta presunción errada.

De este simple ejercicio nos queda una cosa muy clara, existen diferencias entre las grabaciones de un mismo locutor, una persona producirá su habla de una manera similar pero no idéntica en cada ocasión, por ahora no es necesario comprender cabalmente el porqué, nos basta con estar seguros de que cada vez que se habla aunque sea la misma palabra y con la intensión de hablar idéntico cada iteración será distinta.

Con esto establecido ahora se presenta la pregunta, y si cada grabación de una misma persona es distinta ¿Cómo podré saber cuándo se trata de una persona distinta?, la respuesta es "márgenes de variación", aunque cada vez que se habla nuestra voz es distinta, ésta debe de variar en algunos parámetros en menor cuantía que cuando se trata de una persona diferente, lo que es difícil es cuantificar de manera confiable este margen de variación "aceptable".

Para ejemplificar esto de una manera simple vamos a ver nuestro siguiente ejemplo, nos interesan estos márgenes de variación "aceptables" y como calcularlos, esto también nos servirá para dar una mejor apreciación de por qué cuesta trabajo obtener uno o más parámetros adecuados para identificar a una persona por su voz.

Para esto veremos el tono fundamental de varios locutores apoyándome en las muestras de voz que recolecte durante la elaboración de mi tesis de licenciatura, tomaré diez locutores

femeninos y para cada locutor tomaré 5 repeticiones de la palabra “Probando” que serán estudiadas por separado, todas las grabaciones se obtienen en condiciones iguales y sin interrupciones. Como se aprecia en la figura 2.5 la media geométrica de los distintos locutores no es de fácil distinción, a pesar de que cada locutor es representado con una figura y color distinto, la razón es que se sobre imponen uno a otro.

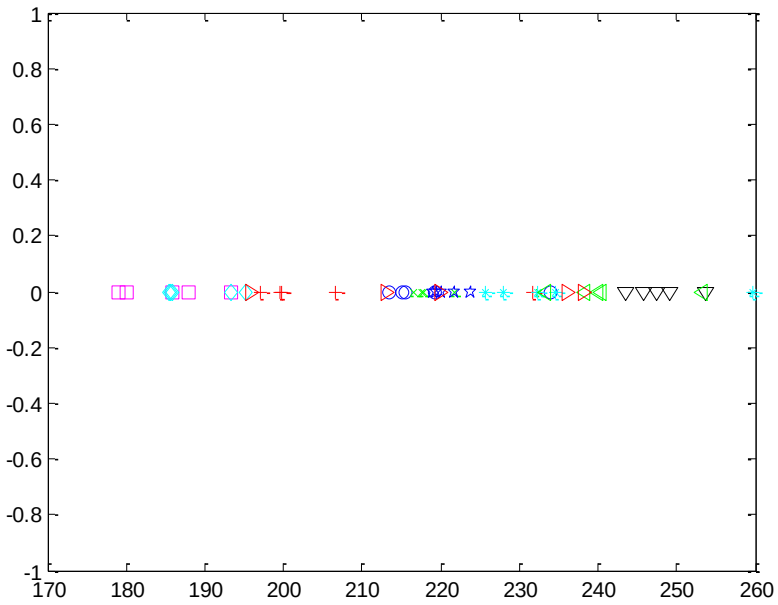


Figura 2.5: Media geométrica del tono de cada locutor

En la figura 2.6 podemos observar de igual manera con distintos colores y formas cada una de las 5 iteraciones de la misma palabra de los diez locutores femeninos en cuanto a su tono fundamental medio (media geométrica), sólo para mejorar la manera en que esto se aprecia, separaré a cada locutor por una unidad en el eje vertical que en realidad no tiene uso en este tipo de gráfica, obteniéndose lo siguiente.

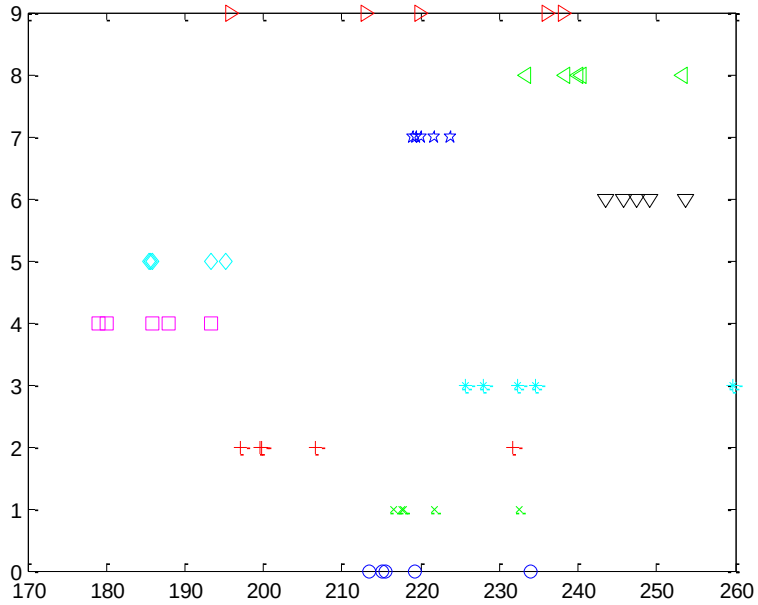


Figura 2.6: Media geométrica del tono de cada locutor en distinta 'y'

En esta gráfica es más fácil observar lo que se quiere demostrar, cada locutor tenderá a estar en una región que representa su tono fundamental "estándar" o normal, aunque en ciertas iteraciones salga del mismo, como lo vemos en la persona representada por las cruces rojas que en una de sus muestras salta fuera de su rango "común", así mismo tenemos personas con tonos sumamente estables como la representada por los triángulos negros o los cuadrados magenta, pero también tendremos personas con un tono disperso en un intervalo grande como la representada los triángulos rojos.

De este simple ejercicio inspirado en el libro de Philip Rose podemos ver que no es tan fácil discriminar o discernir entre un locutor y otro por un parámetro simple como la media geométrica del tono fundamental, como vemos por este parámetro sería difícil o imposible diferenciar entre el locutor de los cuadros magenta y el de los diamantes cian que se traslapan prácticamente por completo.

Ahora bien de este ejercicio también podemos ver que no todo está perdido, existen una gran cantidad de parámetros que pueden ser estudiados, permaneciendo en el mismo tenor de la información estadística veremos una gráfica en dos dimensiones con la mediana en su eje horizontal y la desviación estándar en su eje vertical.

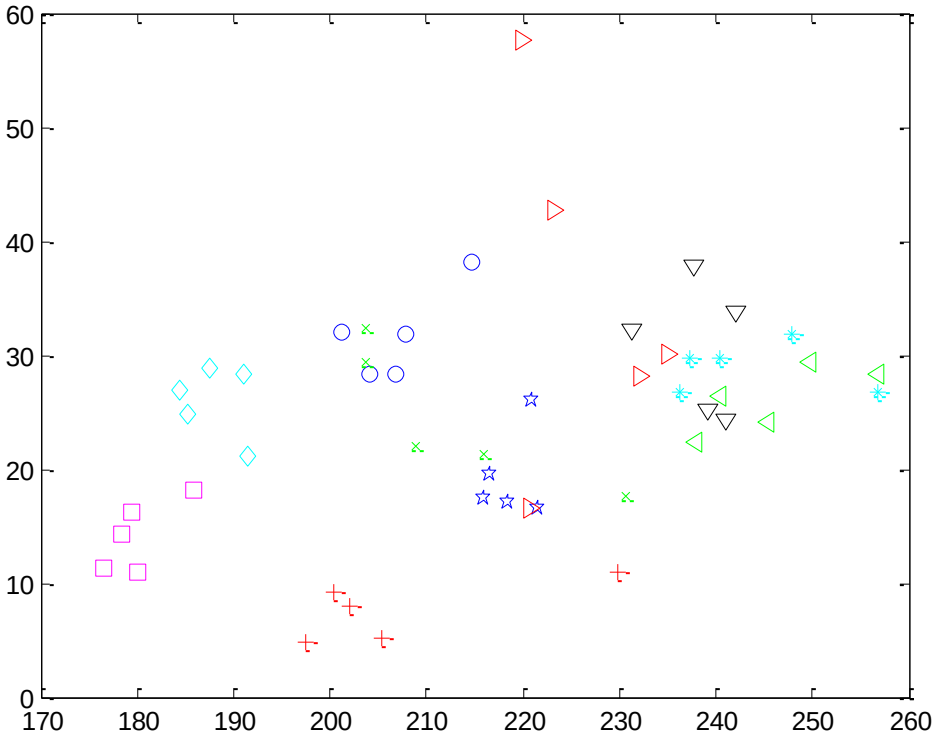


Figura 2.7: Grafica de la media y la desviación estándar

En esta grafica podemos ver que aunque todavía existen empalmes, aparentemente se obtiene una segmentación más eficiente de los locutores con distintas iteraciones de una palabra, aunque existen sujetos que aún presentan una gran dispersión o poca estabilidad, pero esto puede ser llevado aún más allá, como veremos en la siguiente gráfica con tres parámetros simultáneos.

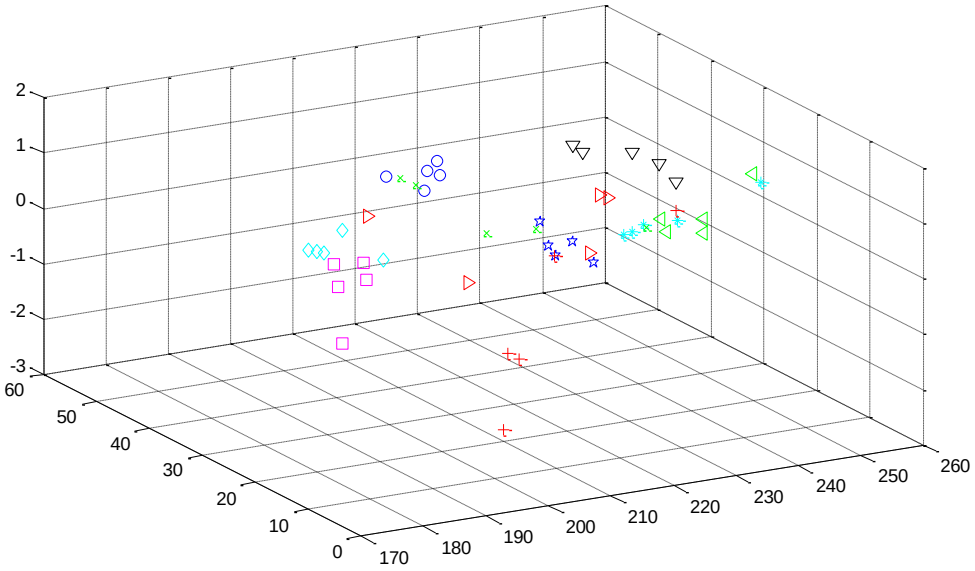


Figura 2.8: Mediana, desviación estándar y skewness

En esta gráfica podemos apreciar que sin salir de las simples medidas estadísticas del tono fundamental de una persona graficamos su mediana, la desviación estándar y su skewness (el sesgo de la distribución) reforzamos aún un poco más la separación en grupos o nubes de los datos que representan a una persona y sus distintas iteraciones.

Los códigos desarrollados en MATLAB por medio del cual se obtienen todas las gráficas mostradas, así como el resto de los códigos que se incluyen en esta tesis están en el anexo 1 al final del documento para su consulta.

Ahora que tenemos ya el concepto de que discriminar a un locutor de otro puede no resultar tan simple, veremos un poco más de cómo lograr diferenciar o discriminar entre un mismo locutor y uno distinto, poniendo especial interés en cómo reconocer si se trata de un mismo locutor.

2.2 Minería de datos (Data Mining)

La minería de datos consiste en la extracción no trivial de información que reside de manera implícita en los datos. De manera general esta información era previamente desconocida, pero aún así es de crucial importancia para algunos procesos. Este fin es logrado por la preparación o pre-proceso de la información mediante el sondeo y la exploración de ésta para “sacar” la información útil que está oculta en los datos mismos.

Este concepto en general engloba un gran número de técnicas encaminadas a la extracción de un conocimiento o inteligencia de la información, debe lograrse que sea más manejable o entendible que la información original que se encontraba implícita en los datos, en otras palabras es un proceso que extrae “inteligencia” de la información masiva que de otra manera no es trabajable o entendible.

Un ejemplo sencillo sería tener la estatura de todos los alumnos de una escuela primaria en una base de datos, esta información es almacenada junto con otra serie de datos como género, edad, peso,

nombre, etc. Aquí como podemos ver, pues tenemos mucha información pero expresada de una manera poco inteligente o poco útil.

Ahora, qué tal si tenemos una tarea concreta que debe de ser resuelta mediante la información almacenada en dicha base de datos, por ejemplo, el problema de la fabricación de uniformes escolares, la determinación de posibles problemas nutricionales para la orientación inteligente del menú de la cafetería o cualquier otro problema similar relacionado con la administración de esta escuela.

Es fácil ver que podemos sacar una campana de distribuciones de estaturas por género, misma que nos puede orientar en cuanto al porcentaje de tallas de cada uniforme que se fabricará para evitar la sobre-fabricación de tallas poco comunes o la falta de uniformes en las tallas más comunes, de igual manera, podemos también realizar una inteligencia de la información para basarnos en la estatura, peso y género para detectar porcentajes de sobrepeso infantil que puedan disparar cambios en el menú de la cafetería o incluso programas de combate a la obesidad infantil.

La minería de datos se basa en el uso de técnicas de inteligencia artificial y en el análisis estadístico de la información que de otra manera es poco manejable, aquí se emplea la extracción de modelos para abordar problemáticas tales como la predicción, la interpretación, la clasificación y la segmentación.

El proceso típico de minería de datos consta de los siguientes pasos:

- Seleccionar los datos a trabajar, tanto para aquellas variables de las que se desea obtener inteligencia como de aquellas que se usarán meramente para su proceso
- Análisis de los datos basado en histogramas, diagramas de dispersión, análisis o eliminación de datos atípicos y/o valores inválidos, etc.
- Transformaciones a los datos basados en el análisis previo, este pre-proceso puede incluir transformaciones en espacios funcionales como la transformada de Fourier, la de Hilbert, Wabelets o cualquier proceso que nos facilite el trabajo con las variables de interés o su interpretación
- Aplicar una o más técnicas de minería de datos y la consecuente construcción de un modelo predictivo, de clasificación o de segmentación
- Extracción del conocimiento mediante una o más técnicas para la obtención de un modelo que representa patrones de comportamiento o relaciones de asociación entre variables
- Interpretación de los datos obtenidos con miras a obtener inteligencia de la información procesada y su validación, misma que puede ser empleada para la comparación de resultados obtenidos por distintas metodologías o técnicas y/o para el disparo de procesos iterativos que se autoajustan para mejorar el resultado obtenido

Una de las técnicas empleadas para la minería de datos son las redes neuronales, que son un paradigma de aprendizaje y procesamiento automático basado en el funcionamiento de un sistema nervioso similar al humano, algunos ejemplos de estas redes son el perceptrón, el perceptrón multicapa, las redes de Kohonen y los árboles de decisión.

Otras técnicas comunes son el modelado estadístico, que es una expresión o ecuación que indica factores de comportamiento o de modificación de una variable, el agrupamiento o clustering donde se agrupa una serie de vectores de acuerdo a ciertos criterios (comúnmente distancias) como el K-means y el K-medoids.

En general, todas estas técnicas pueden ser supervisadas o no supervisadas, los supervisados pretenden predecir uno o más datos a partir de otros datos conocidos, los algoritmos no supervisados o de conocimiento pretenden descubrir patrones o tendencias en los datos.

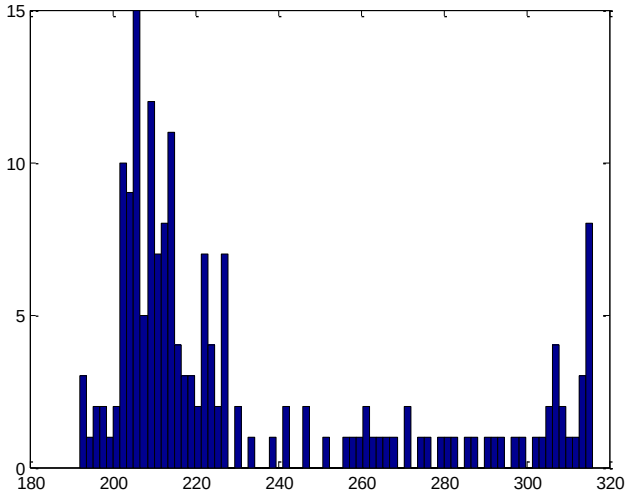
Aquí vemos que la tarea que se pretende lograr con el presente trabajo, incorpora una minería de datos que tiende a ser no supervisada y a utilizar fuertemente el análisis de agrupamiento, el análisis estadístico y el análisis discriminante, ya que tenemos una muy importante cantidad de información (aproximadamente 1,300,000 datos en menos de 2 minutos de habla), de la que pretendemos obtener una información más manejable normalmente por métodos estadísticos, que posteriormente son agrupados según su comportamiento para realizar un análisis discriminante como meta final.

También es conocido de la minería de datos que el mejor modelo o método es aquel que incorpora tantos modelos como sea posible o factible, ya que distintos modelos que "compiten" entre sí podrían emplearse para disparar procesos reiterativos que retroalimentamos al sistema para mejorar la toma de decisiones.

Otra opción en este mismo tren de ideas podría ser la generación de un modelo basado en varios modelos que puede auto-validar su confiabilidad, por ejemplo por el número de resultados en el mismo sentido, para entender mejor esta idea podríamos pensar en el empleo de tres métodos distintos, si al menos dos modelos están en el mismo sentido esta será la decisión mostrada.

Para ejemplificar lo anterior seguiremos trabajando con la información obtenida previamente de 10 locutores femeninos y sus cinco iteraciones de una misma palabra, como modelo propondremos un simple vector con información estadística, pero avancemos paso a paso.

Para comprender todo el proceso y la importancia de la minería de datos, comenzaré por dar algunos datos interesantes, de las 50 palabras con que se está trabajando actualmente la más corta tiene 158 datos sobre su tono fundamental, la más larga 224 y en promedio 193, se tienen 9,653 datos en total, esto ya es un número considerable de datos y no resulta practico de manejar directamente, valiendo la pena recordar que estamos trabajando con archivos que no llegan a 1 segundo de duración por iteración.



Ahora, también podemos fácilmente apreciar que utilizar todos los datos para caracterizar a un locutor no resulta practico, el "modelo" sería grande, difícil de manejar e incluso difícil de interpretar, razón por la cual se recurre a un modelo para representar dicha información, para el siguiente ejemplo emplearé un modelo

que consta de un vector con seis parámetros estadísticos sobre el comportamiento del tono fundamental, media, mediana, desviación estándar, varianza, skewness y kurtosis, que incluso pueden ser estimados a partir de la información que nos muestra un simple histograma del tono fundamental.

Al tratarse de una sola palabra pronunciada de una misma manera en todas las ocasiones, podemos simplificar mucho la obtención de un modelo representativo, pero es importante mencionar que ésto es sólo una pequeña prueba con mucho menor dificultad que la verdadera obtención de inteligencia sobre la información del habla, la idea del ejemplo es meramente mostrar el proceso completo que fundamentara el análisis que se llevará a cabo más adelante.

	1	2	3	4	5
1	-	142.65	174.17	742.54	549.98
2	142.65	-	31.75	600.55	407.84
3	174.17	31.75	-	569.74	376.98
4	742.54	600.55	569.74	-	192.77
5	549.98	407.84	376.98	192.77	-

Tabla 1: Distancia interna para el locutor 1

En esta tabla podemos apreciar una simple medición de la distancia Euclidiana de un vector modelo a otro del mismo locutor para cada una de las cinco repeticiones, siendo el promedio de estas distancias para el segundo locutor de 303.1175 unidades.

Distancia de un punto a otro en un plano bidimensional

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$ Ecuación 1

Para el resto de los locutores tenemos las siguientes distancias y sus promedios:

	1	2	3	4	5
1	-	441.23	432.36	653.98	654.45
2	441.23	-	11.93	212.75	213.24
3	432.36	11.93	-	221.85	222.45
4	653.98	212.75	221.85	-	3.31
5	654.45	213.24	222.45	3.31	-

Tabla 2: Distancia interna para el locutor 2

Promedio: 245.4043

	1	2	3	4	5
1	-	99.03	56.56	70.24	106.58
2	99.03	-	57.38	37.69	13.11
3	56.56	57.38	-	20.67	59.85
4	70.24	37.69	20.67	-	39.90
5	106.58	13.11	59.85	39.90	-

Tabla 3: Distancia interna para el locutor 3

Promedio: 44.8806

	1	2	3	4	5
1	-	305.45	169.95	174.04	34.47
2	305.45	-	138.49	135.44	300.59
3	169.95	138.49	-	5.17	162.34
4	174.04	135.44	5.17	-	165.77
5	34.47	300.59	162.34	165.77	-

Tabla 4: Distancia interna para el locutor 4

Promedio: 127.3381

	1	2	3	4	5
1	-	67.58	125.29	210.58	201.94
2	67.58	-	57.84	143.25	134.50
3	125.29	57.84	-	85.47	76.71
4	210.58	143.25	85.47	-	9.79
5	201.94	134.50	76.71	9.79	-

Tabla 5: Distancia interna para el locutor 5

Promedio: 89.0370

	1	2	3	4	5
1	-	355.10	171.41	280.46	391.51
2	355.10	-	184.31	75.57	37.29
3	171.41	184.31	-	109.18	220.34
4	280.46	75.57	109.18	-	111.20
5	391.51	37.29	220.34	111.20	-

Tabla 6: Distancia interna para el locutor 6

Promedio: 154.9096

	1	2	3	4	5
1	-	294.18	98.08	503.05	549.05
2	294.18	-	391.17	797.12	843.12
3	98.08	391.17	-	406.13	452.14
4	503.05	797.12	406.13	-	46.06
5	549.05	843.12	452.14	46.06	-

Tabla 7: Distancia interna para el locutor 7

Promedio: 350.4070

	1	2	3	4	5
1	-	383.48	398.02	373.23	291.34
2	383.48	-	14.95	10.95	92.24
3	398.02	14.95	-	25.47	106.84
4	373.23	10.95	25.47	-	81.90
5	291.34	92.24	106.84	81.90	-

Tabla 8: Distancia interna para el locutor 8

Promedio: 142.2726

	1	2	3	4	5
1	-	60.25	114.35	306.26	227.48
2	60.25	-	171.05	363.70	285.16
3	114.35	171.05	-	192.77	114.44
4	306.26	363.70	192.77	-	78.83
5	227.48	285.16	114.44	78.83	-

Tabla 9: Distancia interna para el locutor 9

Promedio: 153.1435

	1	2	3	4	5
1	-	118.63	911.57	633.07	2,409.99
2	118.63	-	1,030.02	514.46	2,528.49
3	911.57	1,030.02	-	1,543.80	1,498.53
4	633.07	514.46	1,543.80	-	3,042.29
5	2,409.99	2,528.49	1,498.53	3,042.29	-

Tabla 10: Distancia interna para el locutor 10

Promedio: 1138

De estos datos fácilmente podemos ordenar en una tabla los promedios y comenzar a obtener algún dato interesante.

Locutor	Promedio
3	44.8806
5	89.037
4	127.3381
8	142.2726
9	153.1435
6	154.9096
2	245.4043
1	303.1175
7	350.407
10	1138

Tabla 11: Promedio de distancias

Vemos que si tomáramos como una distancia máxima 300, perderíamos a tres locutores con un resultado equivocado, pero 3 de 7 no suena tan mal para una solución tan simple, si subimos un poco más podríamos perder sólo a 2 o 1 de los locutores, por lo que algunos pensaran, esto no es difícil, pero no hemos visto a qué distancia está una palabra de un locutor de otro distinto para poder tomar una mejor decisión.

Ahora la pregunta sería ¿Comparo una palabra de un locutor contra otra palabra de otro locutor? ¿Comparo el promedio de un locutor contra el promedio de otro?, aquí empiezan los problemas, tenemos mucha información y queremos un sólo resultado, es la misma persona o no lo es.

De primera instancia mostraremos una tabla donde se obtiene la distancia palabra por palabra para cada locutor, si comparamos al locutor 1 contra el locutor 2 obtenemos la siguiente tabla:

	1	2	3	4	5
1	1150.240	1008.726	978.113	408.773	601.346
2	709.698	567.792	537.041	33.040	160.155
3	719.189	577.128	546.274	23.664	169.301
4	497.644	355.385	324.463	245.337	52.719
5	496.892	354.717	323.863	245.898	53.266

Tabla 12: Distancias entre locutor 1 y 2

Aquí tenemos resultados que podemos considerar buenos como distancias de más de 1150 y distancias poco convenientes para nuestra aproximación de apenas 23 unidades, para complementar esto y no hacer inmensamente larga la lista de distancias obtenidas, sólo diremos que si comparamos las 5 repeticiones de cada locutor para los 10 locutores distintos su comportamiento general es el siguiente, la distancia más pequeña es de 2.62, la más grande es de 3,292.89 con una media de 542.26.

Para comprender mejor estos datos veremos un poco más a fondo los resultados, de los 2,500 comparativos anteriores, 250 deberían de ser un mismo locutor y el resto locutores distintos, así que para esos 250 comparativos en específico desechando las distancias cero (una palabra contra sí misma) tenemos una distancia mínima de 3.31, una máxima de 3,042.28 y una media de 343.62 y para los 2,250 comparaciones de locutores distintos tenemos una distancia mínima de 2.62, una máxima de 3,292.89 y una media de 559.92, lo que ya nos dice de primera instancia que este método es poco útil ya que no es lo suficientemente discriminante, mismo que no sorprende a nadie, el tono fundamental puede ser útil, pero no con un análisis tan simple, aunado a ésto, el análisis del tono fundamental presenta severas limitantes propias como su baja resistencia a los cambios por el estado de ánimo del sujeto.

Del anterior ejemplo vemos que se requiere algo más efectivo, sabemos que requerimos extraer inteligencia de la información que tenemos, pero la pregunta es ¿Cómo realizarlo?

2.3 Antecedentes históricos

Una de las primicias en la investigación científica es conocer donde estamos parados, el antecedente histórico y el estado del arte de aquello que pretendemos realizar, en nuestro caso, la investigación en el reconocimiento automático de locutores e incluso del habla tiene ya a la fecha poco más de cinco décadas, aquí se revisarán los principales temas y avances que en este tenor se han dado para brindar una perspectiva tecnológica del progreso logrado, mismo que nos dará una perspectiva de las técnicas desarrolladas y de los retos que a la fecha continúan vigentes para que estas tecnologías puedan ser eficientes y viables.

Al ser el habla el principal medio de comunicación del ser humano, la investigación en el área del procesamiento automatizado del habla ha captado la atención del ser humano, teniendo como una de sus principales metas la comunicación natural hombre-máquina, pero también otras metas importantes como la identificación de locutores, ésto por supuesto gracias a los avances en el modelado estadístico principalmente.

60's y 70's

Los primeros intentos para reconocer a los locutores de manera automática fueron realizados en los sesentas, una década más tarde que el reconocimiento del habla. S. Pruzansky en 1963 trabajando para laboratorios Bell, fue de los pioneros en la investigación mediante el uso de bancos de filtros y la correlación de espectrogramas para la medición de similitudes. Posteriormente se une a él M. V. Mathews, logrando mejorar esta técnica, más tarde G. R. Doddington trabajando para Texas Instruments remplazo el análisis de bancos de filtros con un análisis de formantes. La variabilidad intra-locutor es uno de los problemas más graves en el reconocimiento de locutores, ésto fue investigado detalladamente por W. Endres y S. Furui en los años setentas con sus respectivos trabajos sobre "espectrogramas de la voz humana y su variación en función de la edad de la persona" y "variabilidad a largo plazo en los parámetros del habla para el reconocimiento del locutor".

En estas décadas nacen también los métodos independientes del texto, éstos tienen como propósito el extraer características independientemente del contenido fonético, obtenían varias medidas al promediar parámetros a un plazo lo suficientemente largo o mediante la extracción de información estocástica y parámetros predictivos. Este tipo de parámetros incluyen la auto-correlación promedio instantánea sobre matrices de covarianza espectral como lo propuso P. D. Bricker, las matrices de covarianza espectral instantánea como lo propuso K. P. Li y G. W. Hughes, los histogramas del espectro y de la frecuencia fundamental como lo propuso B. Beek,

los coeficientes de predicción lineal como los propuso M. R. Sambur y espectros promedio de largo plazo como los plasmo S. Furui en 1974.

Dado que la eficiencia de los sistemas independientes del texto era limitada, los métodos en el dominio del tiempo y los dependientes del texto fueron también investigados. Para los métodos en el dominio del tiempo con el adecuado alineamiento, se podía realizar una comparación confiable entre dos producciones vocales del mismo texto en ambientes fonéticos similares.

En este tenor Texas Instruments construyó el primer sistema automatizado a gran escala para la verificación de locutores con un alto grado de seguridad operacional. La verificación se basaba en una producción de cuatro palabras elegidas al azar de un juego de 16 palabras monosilábicas. Bancos de filtros eran utilizados para el análisis espectral y la estrategia de decisión era secuencial requiriendo hasta 4 producciones para una prueba. Se realizaron varios millones de pruebas en un periodo de 6 años con varios centenares de locutores.

En paralelo, laboratorios Bell construyó un sistema experimental dedicado a trabajar con señales enviadas vía telefónica. S. Furui propuso la combinación de coeficientes cepstrales y sus coeficientes polinomiales de primer y segundo orden como base para incrementar la efectividad contra la distorsión del teléfono. Se implementó un sistema en línea y se probó por seis meses con múltiples llamadas de 120 locutores. El método de base cepstral se convirtió después en un estándar no sólo para el reconocimiento de locutores sino también para reconocimiento de habla.

80's

Como una alternativa al empatado de patrones de los sistemas dependientes del texto, la técnica de los modelos ocultos de Markov o HMM por sus siglas en inglés fue introducida tanto para la identificación de locutores como para el reconocimiento del habla. Las técnicas HMM tienen las mismas ventajas para la identificación de locutores que para el reconocimiento del habla, se pueden obtener modelos sumamente robustos para el habla con poca cantidad de información de entrenamiento. Los sistemas de reconocimiento de locutores basados en este método usan modelos derivados de frases, palabras o incluso fonemas, típicamente frases de unas pocas palabras como una serie de números se modelaban mediante el análisis para cada palabra, así como para las pausas y se combinaban a nivel de la oración de acuerdo a un nivel gramatical previamente definido.

Como otra alternativa se tenía la cuantización de vectores combinada con los modelos ocultos de Markov, ésto contemplaba métodos paramétricos y no paramétricos para reconocimiento independiente del texto. Como un modelo no paramétrico la cuantización vectorial fue investigada por A. E. Rosenberg y M.R. Sambur en 1987 y por F. K. Soong en el mismo año, un juego de parámetros obtenidos de ventanas de corta duración pueden ser eficientemente compactadas a una serie de datos o vectores representativos llamados libros código o VQ. Así mismo, una codificación de matrices de cuantización multi-cuadro fue investigada por M. Sugiyama y por B. H. Juang.

Por parte de los modelos paramétricos, los modelos ocultos de Markov fueron investigados por A. B. Poritz, mismo que propuso un HMM ergódico (por ejemplo, todos los posibles estados de transición son válidos). El habla era caracterizada como una secuencia de transiciones a través

de una HMM de 5 estados en el espacio de características acústicas. N. Tishby expandió la idea de Poritz al usar una HMM de 8 estados ergóticos auto-regresiva representada por una función de densidad de probabilidad continua con una mezcla de 2 a 8 componentes por estado, este modelo resultó tener una resolución más alta que la propuesta por Poritz. R. Rose y R. A. Reynolds propusieron una HMM de un sólo estado que es ahora llamado modelo de mezclas Gaussianas (GMM) y que presenta un modelo paramétrico robusto.

90's

En esta década se incrementa la investigación y el robustecimiento se convierte en el punto central. T. Matsui y S. Furui comparan el método basado en VQ con el HMM ergódico discreto y continuo, particularmente desde el punto de vista de la resistencia del algoritmo contra las variaciones intra-locutor, descubren que el método HMM ergódico continuo es muy superior al discreto y que éste también es igual de robusto que el basado en el método VQ cuando se tiene el suficiente material de entrenamiento. Investigaron las tasas de efectividad en la identificación de locutores utilizando un modelo HMM continuo como función del número de estados y mezclas, se demostró que las tasas de éxito estaban estrictamente ligadas al número de mezclas y no con el número de estados. Esto significa que al usar información sobre transiciones entre estados es ineficiente en las técnicas independientes del texto y por lo tanto la técnica GMM alcanza prácticamente el mismo desempeño que un modelo HMM ergódico multi-estado.

T. Mitsui y otros propusieron un método de reconocimiento de locutores que pedía una frase específica, esta frase cambiaba cada vez que se usaba el sistema, el sistema aceptaba la frase sólo cuando se determinaba que el locutor registrado produjo la sentencia. Dado que el vocabulario es ilimitado, los impostores no pueden saber que frase se les pedirá producir. Este método no sólo podía discriminar exitosamente cuando el verdadero locutor hablaba, sino que podía distinguir si se dijo la frase solicitada incluso cuando lo decía un locutor ya registrado.

Otro de los temas de interés fue la normalización de las calificaciones o distancias en los casos de un mismo locutor, esto es, como minimizar la variabilidad en un mismo locutor, que es uno de los problemas más difíciles en la verificación de locutores, como ya sabemos existen diferencias importantes en distintas grabaciones de un mismo locutor, y esto se acentúa más por la influencia de las condiciones de grabación y del canal, un locutor no puede producir dos veces la misma frase aun cuando así lo quiera.

En este tenor el radio de verosimilitud y la probabilidad a posteriori fueron investigadas por múltiples científicos como A. Higgins en sus trabajos sobre la verificación de locutores con frases producidas al azar, T. Matsui y S. Furui en su trabajo sobre la normalización de similitudes para la verificación de locutores por medio de la probabilidad a posteriori y D. Reynolds en su trabajo sobre la identificación y verificación de locutores mediante GMM, para reducir el costo computacional de calcular la normalización se propuso agrupar a los locutores y el uso de modelos poblacionales.

Aunado a las líneas de investigación ya mencionadas, se desarrollaron en paralelo técnicas relacionadas al reconocimiento del habla por adaptación del locutor, esto se reflejó también en la mejora de las voces sintetizadas al añadir tesituras naturales de un locutor en particular,

permitiendo la transformación de una voz sintetizada en otra distinta con el simple cambio de parámetros.

Por otra parte, la separación de locutores en un mismo diálogo o conversación aparece como una extensión del reconocimiento del parlante, como lo abordaron H. Gish, M. Siu y R. Rohlicek en sus trabajos sobre la segregación de locutores para la identificación, M. Siu y otros en sus trabajos sobre algoritmos no supervisados para la segmentación de locutores y L. Wilcox y otros con su trabajo para la segmentación mediante la identificación del locutor. Estas técnicas de agrupamiento de locutores y de segmentación han sido importantes en áreas del reconocimiento de voz, de indexado automatizado y para la creación de motores de búsqueda.

2000's

En esta década se dedicó la mayor parte de los esfuerzos a las técnicas de normalización, en estas técnicas se emplea la sustracción de la media y la división por la desviación estándar, ambos términos se estiman de la distribución de puntuaciones basadas en impostores.

Para la obtención de la distribución de puntuaciones de impostores existen diferentes métodos, Z-norm, H-norm, T-norm, HT-norm, C-Norm y D-norm como lo propusieron F.J. Bimbot y otros científicos en el 2004. Los métodos de estado del arte en el reconocimiento de locutores independiente del texto, asocian una o más parametrizaciones de normalización de nivel como CMS (sustracción cepstral de la media), normalización por varianza de los parámetros, adaptación de parámetros u otras conjuntamente con un modelo global y uno o más métodos de normalización de puntaje.

Otras de las metodologías que se plantean también en esta década son las de extracción de características de alto nivel tales como el idiolecto, pronunciación, prosodia y otras que se han utilizado con cierto éxito en la identificación independiente del texto. Típicamente estos métodos producen una secuencia de símbolos acústicos de la grabación y realizan el reconocimiento usando una tabla de frecuencias de ocurrencia de los símbolos.

En el trabajo realizado por G. R. Doddington sobre el idiolecto, unigramas y biogramas de palabras transcritas manualmente fueron utilizadas para caracterizar a un locutor en particular para calcular un radio de verosimilitud estándar, pero de manera manual.

El progreso en los últimos 50 años en la identificación automatizada de locutores y por ende de una de sus ramas cercanas, el reconocimiento de habla se puede resumir en la siguiente tabla:

Pasado	Actual
Empatado de patrones	Modelado estadístico con base en una población de referencia
Bancos de filtros y resonancia espectral	Características cepstrales
Normalización en tiempo heurística	DTW y programación dinámica
Métodos basados en distancias	Métodos basados en LR
Aproximación de la máxima probabilidad	Aproximación discriminativa
Reconocimiento con palabras aisladas	Reconocimiento con habla continua
Basado en hardware	Basado en software
Habla limpia	Habla telefónica o con ruido
Sin aplicaciones practicas	Múltiples campos de aplicación

2.4 Estado del arte

Una vez vistos los antecedentes históricos que nos ayudan a repasar el camino andado y a comprender las tendencias actuales de investigación, podemos dar inicio a ahondar un poco más en algunas de las principales características que individualizan a una persona y que nos permiten identificarla por su voz.

De la extensa investigación que ha existido en los últimos años podemos ver que de manera general la información de bajo nivel que nos permite individualizar a una persona, se encuentran en la envolvente de la espectral o cepstrum, ésta nos permite separar las características del tracto vocal del resto de la información contenida en el habla, éstas se extraen en un análisis de corta duración (short-time analysis) de sus coeficientes cepstrales.

En periodos lo suficientemente cortos la voz se comporta como una señal cuesi-estable, esto es, si tomamos porciones del habla cortos (menos de 30 milisegundos) la variación es poca, motivo por el cual se dividirá la señal de voz en cuadros o ventanas de pequeña duración, para comprender mejor esto basta con observar las figuras siguientes, en la figura 2.9 vemos un poco más de un segundo de habla, la señal sube y baja constantemente cambian su frecuencia, su energía, etc. No hay parámetros estables a estudiar.

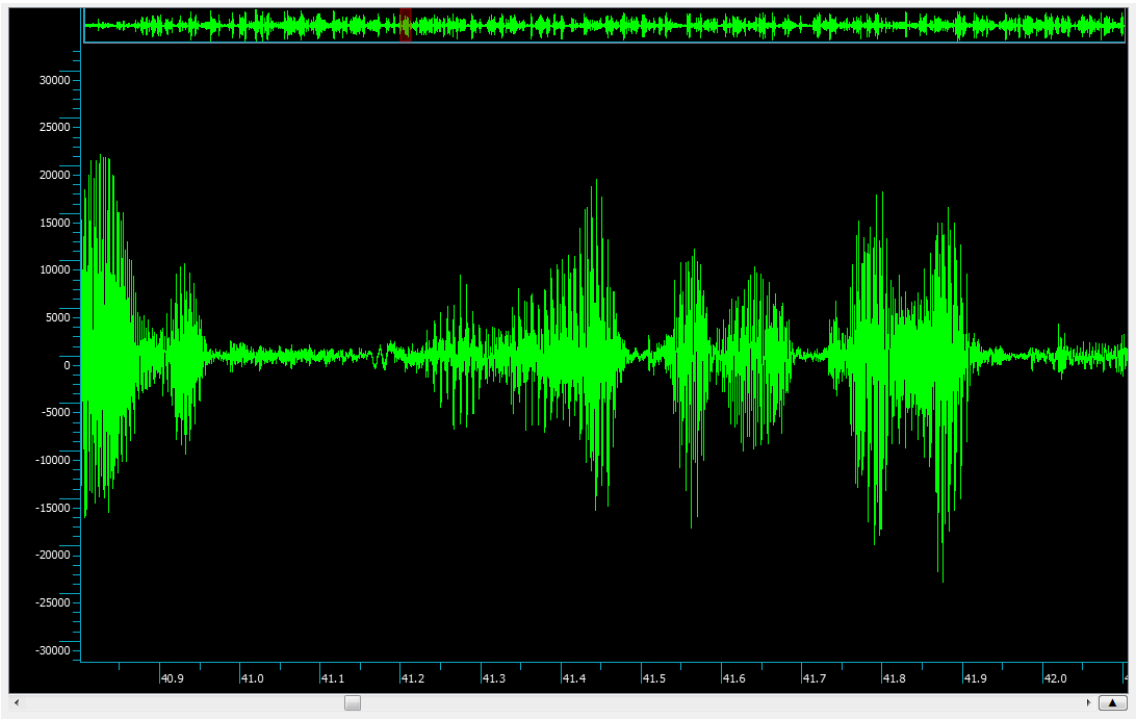


Figura 2.9: Aproximadamente un segundo de habla

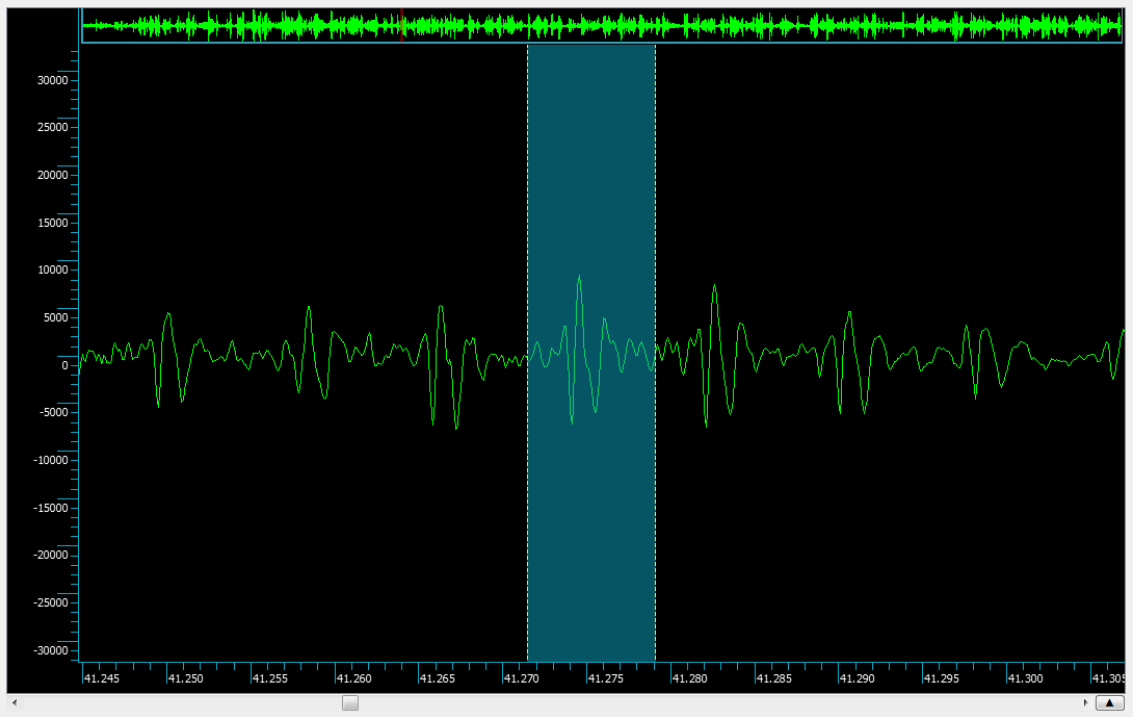


Figura 2.10: Aproximadamente 150 milisegundos de habla

En la figura 2.10 vemos que la señal se muestra más estable, se repite, es periódica, motivo por el cual esta porción de información sería más analizable, si tomamos un sólo periodo de la señal como la que está en el recuadro azul tendremos un espectro considerablemente estable, como el que se puede apreciar en la figura 2.11, aquí de nuevo se ve que la imagen, sin importar mucho su significado o interpretación es similar en toda la ventana, con más o menos energía, con un poco más o menos de componentes, pero en general parecida.

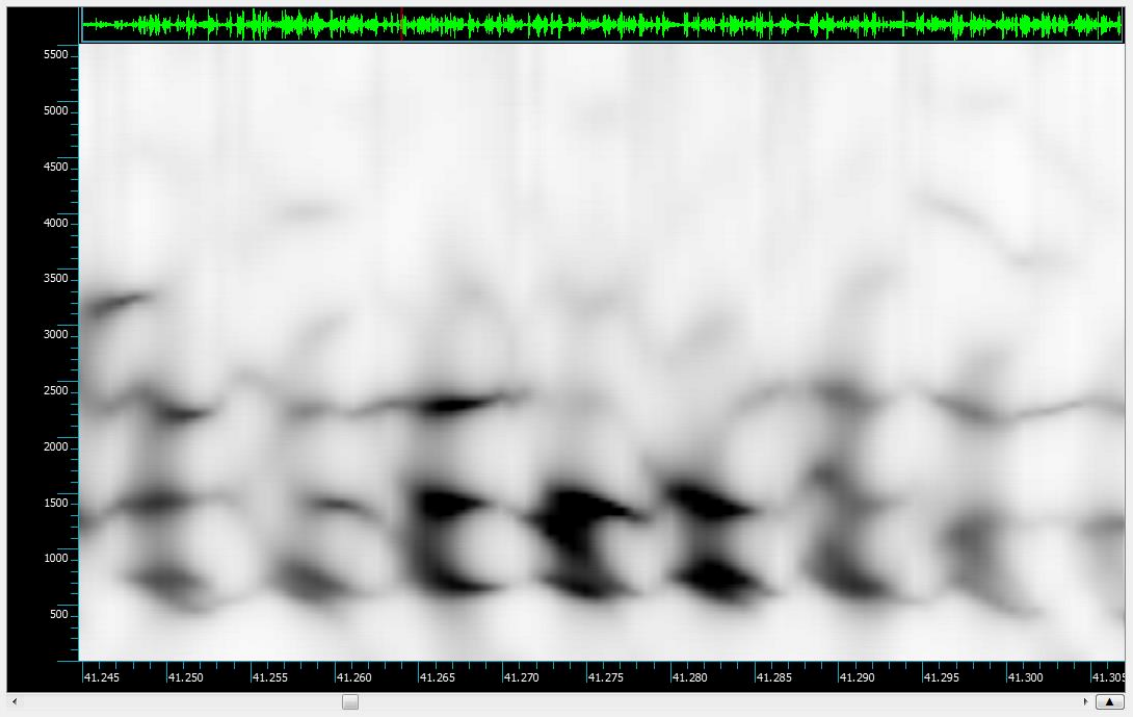


Figura 2.11: Espectrograma de la porción de 150 milisegundos

En la figura 2.12 vemos el promedio del espectrograma de la figura 2.11, si hubiéramos obtenido este promedio de un periodo más largo veríamos que es poco representativo, en este caso, los máximos del promedio coinciden totalmente con los armónicos del tono fundamental (las porciones más obscuras del espectrograma).

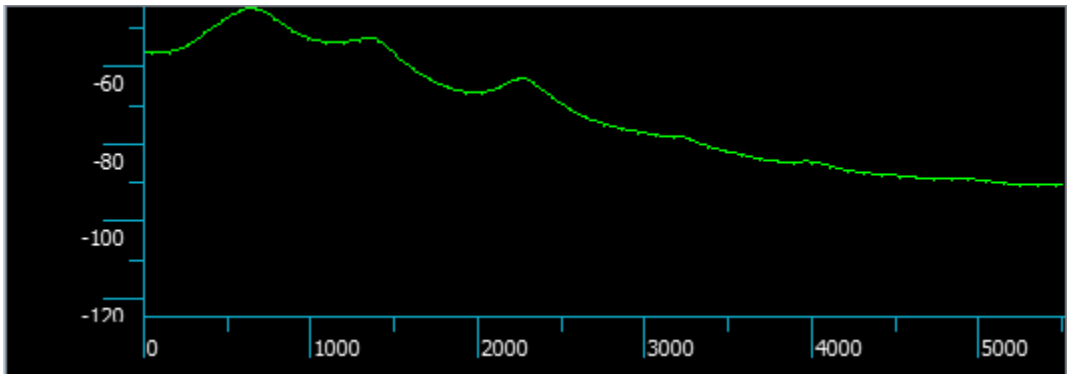


Figura 2.12: Promedio del espectrograma

Ahora bien, para evitar el efecto borde o de perdida se sobrepondrán las ventanas entre sí, esto puede ser en distintos porcentajes, pero un ejemplo podría ser con un 50% aunado a la aplicación de una ventana matemática a las mismas para evitar el efecto frontera, un ejemplo típico de ventana matemática utilizada en el estudio de la voz humana es la ventana de Hamming que está dada por la siguiente ecuación:

$$w(n)=0.54-0.46\cos(2n\pi/(N-1)) \qquad \text{Ecuación 2: Ventana de Hamming}$$

Este tipo de configuración nos deja preparados para realizar una transformación de la señal a un dominio que nos permita observar de manera más clara cómo es que está compuesta la señal, esto de manera común se realiza mediante la transformada de Fourier de cada ventana como se muestra en la figura 2.13.

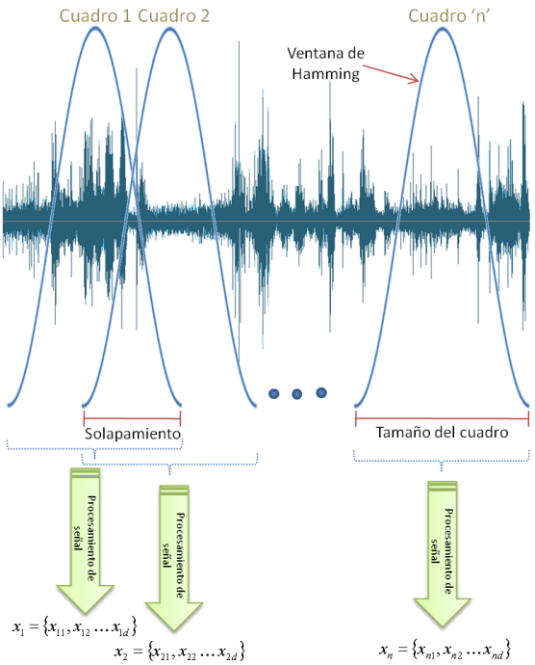


Figura 2.13: Procesamiento de la voz en periodos cortos

De los antecedentes históricos, sabemos que una de las técnicas que más prometen es la caracterización de la envolvente espectral del habla. Esta se logra mediante el análisis espectral de corto plazo basado en los coeficientes espectrales y sus coeficientes de regresión, típicamente los coeficientes de primer y segundo orden (coeficientes delta y delta-delta), para obtener la envolvente se emplea un banco de filtros, este banco de filtros tiene como función simplificar la señal que obtenemos después de su transformación por cada ventana.

En general una envolvente como su nombre lo dice, es simplemente una función que rodea nuestra señal de interés, por ejemplo, si obtuviéramos una envolvente de una forma de onda, esta se vería similar a lo que tenemos en la figura 2.14

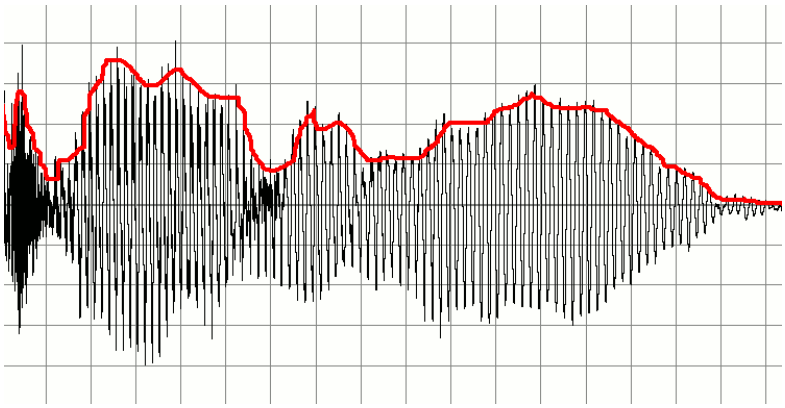


Figura 2.14: Envolvente de una forma de onda

Lo que nosotros tenemos en realidad es una transformada instantánea de Fourier como la que se ve en la figura 2.15, como se puede observar, ésta tiene mucho ruido, motivo por el cual se obtiene su envolvente, ésta es como la que se muestra en la figura 2.12, los filtros concentraran una porción de este espectro en un solo valor que representara a una banda del espectro, logrando así reducir el número de datos de manera considerable, por ejemplo, si tenemos una ventana de 512 puntos esto producirá un espectro de 256 datos (la mitad es redundante pues es simétrica), y de ésta podemos obtener digamos 20 o 30 representantes mediante el banco de filtros.

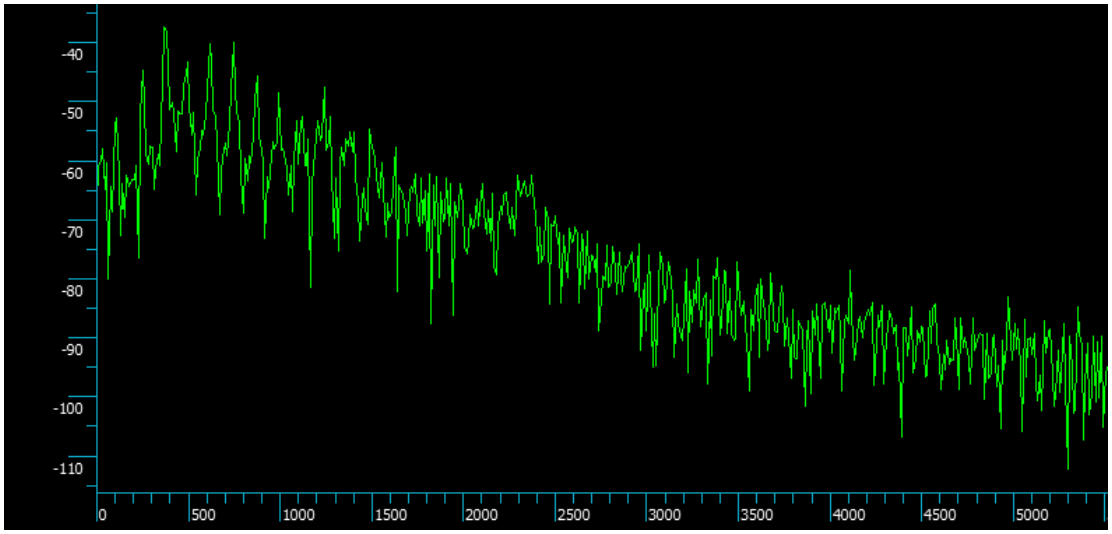


Figura 2.15: Espectro instantáneo de habla

Para ejemplificar mejor como se aplica el banco de filtros, en la figura 2.16 podemos apreciar una superposición del espectro con un banco de 12 filtros de Hamming con un solapamiento del 50% y una escala Mel, estos parámetros serán explicados más adelante a detalle.

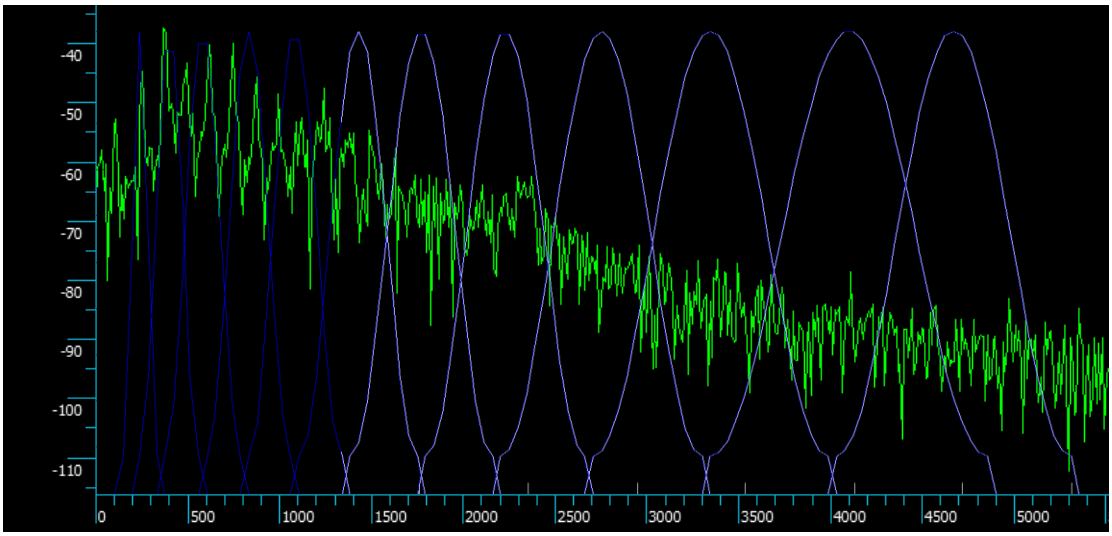


Figura 2.16: Aplicación del banco de filtros

Una vez que tenemos la envolvente del espectro se aplicara una segunda transformación de Fourier, misma que nos entregará los ya comentados coeficientes cepstrales en escala Mel o MFCC, esto también puede ser graficado y quedaría como se muestra en la figura 2.17.

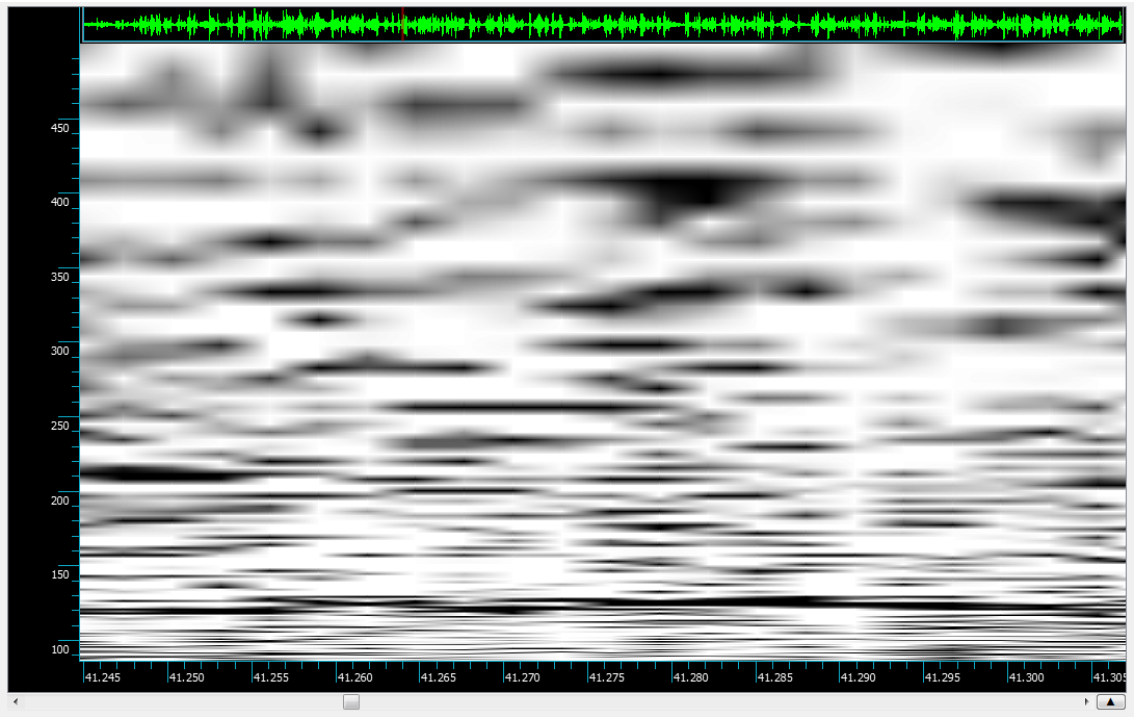


Figura 2.17: Ceptrograma de la señal de voz

Como estos coeficientes no contienen información de cómo se comporta dinámicamente la señal, se añaden las derivadas de primer y segundo orden de la función cepstral en el dominio del tiempo que se extraen por cada cuadro o ventana y que representan la dinámica espectral.

Estos coeficientes son comúnmente conocidos como coeficientes cepstrales-delta y cepstrales-delta-delta, sin embargo, más delante de manera empírica veremos si nos es provechoso emplearlos, así como otras opciones existentes para tener una mejor comprensión de las técnicas más recientes.

Una vez que se tienen los coeficientes veremos que aún tenemos mucha información, miles de datos por cada muestra, por lo que este material no es susceptible de compararse y aún menos de dar un buen resultado, después de todo, se tiene información relativa al mensaje entre otras cosas y eso no es nuestro interés por el momento, por esto veremos un poco del paradigma del reconocimiento de patrones.

El objetivo de un reconocedor de patrones es, dada una serie de atributos de una clase desconocida (entiéndase clase como el elemento a identificar por el reconocedor, voces, formas, letras, caras, etc.), identificarla dentro de las clases conocidas por el reconocedor. Así, esta metodología consta de dos etapas: un entrenamiento del reconocedor para caracterizar las distintas clases con sus peculiaridades (entrenamiento) y el proceso de identificar características desconocidas en este universo entrenado (pruebas).

El reconocimiento automatizado de locutores comienza con la definición de si el sistema será para identificar o para verificar, cómo se explicará un poco después. El primero corresponde a un reconocimiento dentro de N clases o individuos, pudiendo ser este conjunto un set abierto o cerrado. Un verificador de locutor representa la aceptación o rechazo sobre una clase en particular. También el sistema puede ser dependiente o independiente de texto.

Ahora bien, para alimentar este reconocimiento de patrones se requiere reconocer y adquirir los atributos que mejor representen los rasgos de interés. La señal de voz se descompone en dos tipos de información: uno referente a los sonidos que se emiten y otro que caracteriza el locutor por sus resonancias del tracto vocal, a nosotros nos interesa por supuesto la segunda.

Una vez que se extraen las características de cada individuo, la siguiente etapa es clasificarlos y agrupándolos de tal forma que su separación pueda ser medida y comparada. Si se considera el cálculo de al menos 38 coeficientes cepstrales por cada 20 milisegundos de señal de la voz, entonces se tiene 1,900 coeficientes por cada segundo, que pueden ser agrupados usando diversas técnicas, siendo las principales las que se definen a continuación.

Cuantización Vectorial (VQ)

Este método ha sido utilizado como algoritmo de clasificación de las características del locutor. Se basa en aprendizaje no supervisado, es decir, un algoritmo (K-means) agrupa

automáticamente cada clase conocida (set de locutores). Existen muchos algoritmos de agrupamiento, siendo los más conocidos Linde-Buzo-Gray (LBG) y learning vector quantization (LVQ).

En esencia, las características de cada uno de los distintos individuos se dividen en un set de regiones convexas mutuamente exclusivas, computándose el centroide de cada región. La colección de centroides se denomina libro código, siendo típico en reconocimiento del hablante el cálculo entre 32 y 64 centroides por libro código. Para el proceso de test, el vector de características de una voz desconocida se compara con las distancias a los distintos libros código, siendo la similitud su distancia acumulada mínima.

Redes Neuronales (NN)

Otra categoría de algoritmos son los clasificadores supervisados, siendo el más popular el de las redes neuronales, en particular del perceptrón multicapa (como el de la figura 18). El vector de coeficientes cepstrales se multiplica por los respectivos pesos y los resultados se suman y se evalúa en la función de activación, del tipo sigmoidal, escalón u otra. En este caso, la salida puede ser 1 ó -1, simbolizando la pertenencia o no a alguna clase.

La esencia principal del aprendizaje supervisado responde a la existencia de una serie de ejemplares de entrenamiento, los cuales pasan sucesivamente por esta red neuronal, en este caso forzando uno de los dos valores en la salida, lo que trae consigo que el error sea asumido por la modificación de los valores de los pesos (retropropagación del error). Múltiples perceptrones son utilizados para separar más clases, siendo usual que la salida de un perceptrón sea la entrada a otro.

Gráficamente, el perceptrón multicapa modela una línea de decisión de las clases entrenadas, lo que trae consigo que para incorporar una nueva clase, se deba entrenar nuevamente toda la red. En el caso del reconocimiento del locutor, el entrenamiento se realiza con las voces de todos los individuos de interés. En la prueba, el vector de características de la voz desconocida es sometido al paso de esta red ya entrenada (con todos los pesos determinados) y su salida se asociará con alguna de las clases.

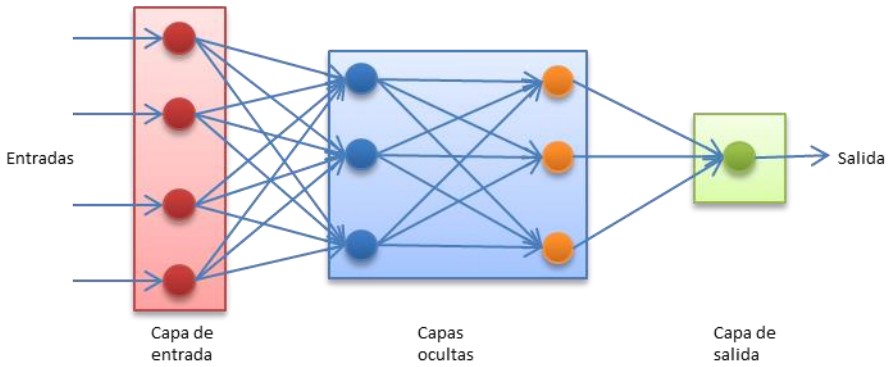


Figura 2.18: Perceptrón multicapa

Modelos Ocultos de Markov (HMM)

Los modelos ocultos de Markov (HMM) corresponden a modelos estocásticos que han sido usados con éxito en el ámbito de reconocimiento del locutor dependiente de texto. Cada palabra de un locutor determinado es generada por un modelo de Markov, el que consiste en

una serie finita de estados interconectados por probabilidades de transición. Cada uno de los vectores de características tiene cierta probabilidad de mantenerse en el estado actual o avanzar al siguiente. Por su parte, cada uno de los estados tiene una probabilidad (o densidad de probabilidad) de presenciar una cierta observación, es decir, observar un vector de características. En este caso, solamente son visibles las observaciones, desconociéndose la secuencia de estados.

En la fase de entrenamiento, se genera el modelo de cada hablante que corresponde a la siguiente ecuación, tal que se maximice la probabilidad de la observación dado el modelo:

$$\boldsymbol{\lambda_j = (A, B, \pi)} \qquad \text{Ecuación 3}$$

Donde A = $\{a_{ij}\}$ es la matriz de probabilidades de transición, B = $b_j(k)$ es la matriz de probabilidades de la observación y π es la probabilidad que cada estado sea el primero (figura 1.18, lado derecho del perceptrón). El cálculo de las matrices A, B y π se realiza a través del algoritmo Baum-Welch o métodos de gradiente, pero no se obtendrá el óptimo, sino máximos locales.

En las pruebas, el problema de empatado o similitud entre una voz desconocida y una clase conocida puede ser formulada por la distancia entre una observación y un modelo de hablante conocido. La observación corresponde a un vector de características, es decir, un vector de coeficientes cepstrales y el modelo del hablante conocido es el modelo oculto de Markov. La distancia antes citada corresponde a una densidad de probabilidad dada por:

$$\boldsymbol{P(S_i = S_j/O, \lambda_j)} \qquad \text{Ecuación 4}$$

Donde S_i es el hablante desconocido, S_j el hablante conocido, O es el vector de observaciones, siendo cada una de ellas una serie de coeficientes cepstrales y λ es el modelo del hablante conocido.

Modelo de Mezclas Gaussianas (GMM)

El modelo de mezclas Gaussianas (GMM) fue introducido por primera vez para reconocimiento de locutores por Reynolds en "Speaker identification and verification using Gaussian mixture speaker models" en 1995 y es el algoritmo estadístico de clasificación más exitoso en la actualidad. El conjunto de vectores de coeficientes cepstrales de un hablante se representa mediante una distribución híper dimensional, la cual es modelada por múltiples Gaussianas, tal y como se muestra en la figura 2.19.

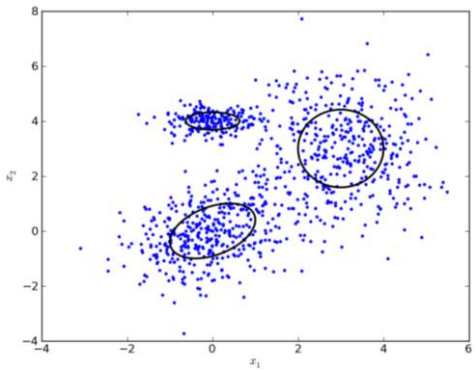


Figura 2.19: Modelado de mezclas Gaussianas

Típicamente en reconocimiento de locutores, se utilizan entre 512 y 1,024 mezclas para una óptima caracterización. Diversos algoritmos se utilizan para generar estas Gaussianas, destacándose Expectation-Maximitation (EM) y Maximum a posteriori probability (MAP).

En ellos, iterativamente se refina la ubicación, anchura y máximo de las Gaussianas hasta un umbral de convergencia. Entonces, en la fase de entrenamiento de una voz conocida se genera un modelo basado en sus propias mezclas de Gaussianas. Luego en la fase de pruebas, se evalúa cada vector de características en la distribución del modelo.

En suma, la puntuación final será la suma de todas estas evaluaciones. A mayor cercanía entre los vectores cepstrales de la voz desconocida y el modelo, la evaluación en la distribución caerá en los valores más altos de las Gaussianas respectivas, por lo tanto su puntuación será más alta. Algunas sesiones de entrenamiento bastaran para determinar el óptimo valor de puntuación para el umbral de discriminación.

Vectores de soporte

El método de vectores de soporte es un método general para la resolución de problemas de clasificación, regresión y estimación, se basa en la teoría estadística del aprendizaje. El principal objetivo es transformar los vectores de entrada, en este caso los vectores de características n-dimensionales en vectores de dimensión más alta en los que el problema puede solucionarse linealmente. Esta frontera de decisión lineal corresponde a un híper plano óptimo que separa solamente dos clases, por lo que es una decisión binaria. Por lo anterior, en el reconocimiento de locutores este método se utiliza en verificación de locutores.

El híper plano elegido es el que mejor separa las clases, esto es, que maximiza la distancia euclídea a los vectores de características más cercanos de cada clase, por eso se denomina clasificadores de margen máximo. Estos vectores corresponden a los vectores de soporte.

Para ellos se construyen 2 planos paralelos que pasan por los vectores de soporte. Cuando las clases no son linealmente separables, se busca el híper plano que minimice los errores a través de una función de costo. La extensión a límites no lineales se realiza a través de un núcleo o kernel que satisfaga las condiciones de Mercer.

En esencia, cada vector de características es mapeado a un espacio de alta dimensionalidad en que los datos son separables linealmente. El producto escalar de los vectores transformados, se puede escribir como el núcleo, por lo que no es necesario trabajar en el espacio extendido. Al igual que las redes neuronales, SVM trabaja en bloque, por lo que un cambio en los datos supone la obtención de una nueva máquina.

La etapa de pruebas verifica a qué lado de la línea de separación están los vectores de características de la voz desconocida, siendo un proceso de discriminación. A un lado están los vectores que caracterizan al individuo y al otro el resto de la población.

De manera general, usemos un método u otro de clasificación, la puntuación o score se define como el valor que entrega la evaluación de características vocales de una voz desconocida dentro de las clases conocidas y que han sido entrenadas. Como ejemplo, la figura siguiente

muestra la prueba entre características de una clase desconocida y dos clases conocidas entrenadas con el método de Cuantización Vectorial o VQ.

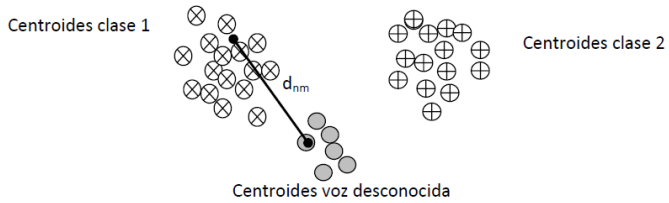


Figura 2.20: Esquema de cálculo de puntuación para dos clases entrenadas y una clase desconocida

Las figuras corresponden a los libros código de cada clase obtenidos con la técnica de cuantización vectorial o VQ, mientras que d_{nm} corresponde a la distancia euclídea entre el centroide desconocido n y el centroide N_m de la clase 1.

En el ejemplo de la figura 2.20, se calcula en la etapa prueba todas las distancias desde los centroides desconocidos a los centroides de las respectivas clases, almacenando la suma de estas distancias para cada una de ellas. Este valor será la puntuación. En el caso de set cerrado, la clasificación de la clase desconocida será aquella cuya puntuación sea la menor, es decir, la suma de todas sus distancias es más pequeña, lo que se traduce en una cercanía entre ambas clases. Sin embargo, en el caso de set abierto, se debe establecer un umbral de decisión para asociar las características vocales desconocidas con alguna de las clases entrenadas.

Si la distancia acumulada para todas las clases es mayor que ese umbral, significa que la voz desconocida no está lo suficientemente cerca de alguna de las clases, por lo que no hay identificación positiva. Para establecer ese umbral, se utilizan muchas muestras conocidas que pertenezcan a las clases entrenadas, que progresivamente van refinando el valor del umbral.

2.5 Determinación de efectividad

En la identificación de locutores la determinación de la efectividad de la solución es básica, pero para comprender mejor ésto veremos unos pocos puntos que nos ayudaran a adentrarnos en cómo realizar esta métrica.

Verificación e identificación de locutores

La verificación de locutores es el proceso de aceptar o rechazar una supuesta identidad en un sistema, esto es, comparar puntualmente dos modelos, uno que está en el sistema y otro que se obtiene de la producción de una contraseña, ya sea ésta fija, cambiante o aleatoria.

La identificación de un locutor consiste en crear un modelo para un locutor desconocido y analizar este modelo contra modelos de locutores conocidos. El locutor es identificado como el modelo que más se parece a la voz analizada, siempre y cuando se tenga un resultado que supere un margen aceptable que se fije en el sistema, éste es determinado por las curvas de los dos errores posibles en una identificación biométrica, la falsa aceptación y el falso rechazo.

La diferencia básica entre verificación e identificación es el número de alternativas de decisión, en la identificación el número de posibles resultados es tan grande como la base de datos, en cambio en la verificación sólo se tienen dos opciones, aceptar o rechazar sin importar el tamaño de la base de datos, en este tenor en todo sistema de identificación de locutores el desempeño se decrementará conforme el tamaño de la base de datos aumenta.

En una identificación de locutores vale la pena también hacer mención de que existen pruebas de set abierto y cerrado, en las pruebas de set abierto el individuo a identificar puede o no estar en la base de datos, en este sentido también hay que tener en cuenta que existe el resultado de que el locutor objetivo no se puede identificar, mientras que en los sets cerrados el locutor objetivo siempre estará en la base de datos.

Tipo de alimentación

La identificación de locutores se puede dividir en tres ramas a partir de la señal de voz que alimenta al sistema, éstos pueden ser dependientes del texto, donde se tiene que tener el mismo contenido textual en las dos o más grabaciones o producciones vocálicas que se desean comparar, esta técnica normalmente hace uso de plantillas o modelos de la secuencia de voz que se desea trabajar, se alineara en el eje del tiempo la muestra para tener un mejor desempeño, un método posible consiste en ir sumando las diferencias encontradas y en base a esto disparar una aceptación o rechazo, a pesar de sus limitantes este método puede ser interesante en algunas aplicaciones debido a su buen desempeño.

El método independiente del texto como su nombre lo dice, no depende del contenido textual de la grabación o producción vocal, esta técnica presenta ventajas considerables sobre la anterior, pero también retos y problemas adicionales, un nivel de dificultad más elevado, una menor eficiencia y normalmente un mayor costo de cálculo.

La tercera vertiente es más bien una modificación del método dependiente del texto, en éste se tiene un texto que se solicita al hablante y éste reproduce esta habla para intentar verificar su voz, concediendo acceso a un sistema, en esta variante sólo se tiene la ventaja de que la contraseña o palabra que se tiene que pronunciar no es fija, ésta varía con cada uso del sistema, no es tan versátil como el sistema independiente del texto pero presenta sus propias ventajas, más rápido, más confiable que el independiente del texto, más seguro que el dependiente del texto, pero al mismo tiempo tiene como principal limitante que solo puede usar el texto predefinido que espera recibir.

Posibles resultados

Entre los sistemas biométricos como los de voz, retina y huellas dactilares se utiliza una misma manera de ofrecer resultados o de realizar su tarea de identificación o verificación, ésto se realiza al ofrecer las dos opciones posibles de decisión: aceptar o rechazar al locutor o sujeto que se está identificando. Ésta lleva a cuatro posibles resultados, dos correctos y dos errados:

- Aceptar un locutor registrado.
- Rechazar un impostor.
- Aceptar un impostor.
- Rechazar un locutor registrado.

Los dos primeros casos corresponden a respuestas correctas por parte del sistema, mientras que las dos últimas opciones son erradas. Estos errores corresponden, respectivamente, a los comúnmente denominados errores de falsa aceptación y de falso rechazo. La tasa donde estos errores se igualan EER (Equal Error Rate) es comúnmente utilizada como medida de desempeño en los sistemas biométricos.

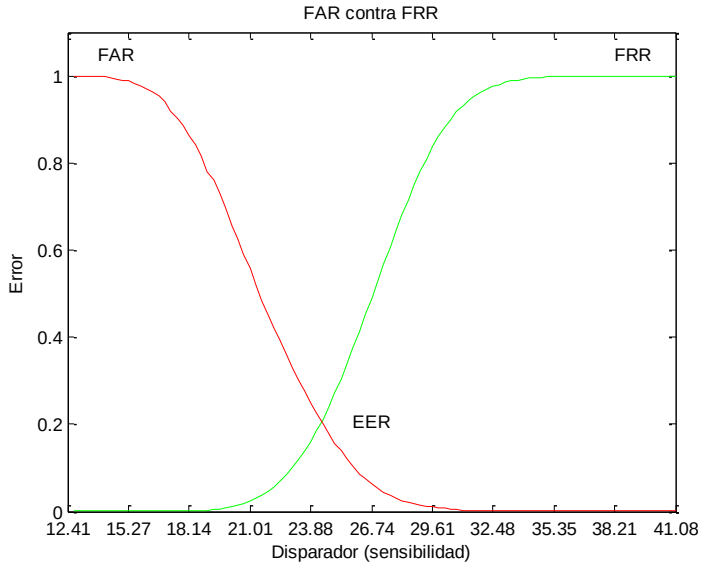


Figura 2.21: Curva FRR vs. FAR

Los errores en un sistema se producen al aceptar un impostor o al rechazar a un cliente. Es importante mencionar que todo sistema posee un umbral de decisión, si éste es demasiado estricto tenderá a rechazar clientes y si es demasiado permisivo tenderá a aceptar impostores, por lo que para aumentar el desempeño del sistema se deben minimizar conjuntamente ambos tipos de errores.

Para encontrar el umbral de decisión óptimo se definen dos curvas: la curva de Falsa Aceptación y la curva de Falso Rechazo. La curva de Falso Rechazo se construye moviendo el umbral de decisión en el rango de clasificación del sistema, identificando para cada uno de estos puntos, qué porcentaje de las ocasiones el cliente trató de verificarse y fue rechazado.

Por su parte, la curva de falsa aceptación se obtiene desplazando el umbral de decisión e identificando para cada punto qué porcentaje de los impostores fue aceptado. La intersección de estas curvas indica el umbral óptimo para el cual se minimiza el error del sistema. Este umbral óptimo se denomina TEER por la abreviación en inglés de Threshold of Equal Error Rate o simplemente EER y es definido a posteriori, es decir, luego de una serie de intentos de verificación tanto por parte del cliente como por impostores.

A partir del TEER se puede encontrar el porcentaje de error del sistema (EER) al evaluar este valor en algunas de las curvas definidas anteriormente. La figura 2.21 muestra gráficamente la situación descrita; en ella se observan las curvas de falsa aceptación y falso rechazo cuya intersección corresponde al TEER. La diferencia principal es que el TEER marca el umbral de trabajo óptimo y el EER marca el porcentaje de error mínimo del sistema, pero se refieren en si al mismo punto en la gráfica.

Efectividad

Ahora que ya vimos que tipos de pruebas existen, que resultados podemos obtener y el punto de operación ideal para un sistema biométrico, podemos decir que en la detección de eventos existen dos tipos comunes de error, el sistema puede dar una falsa alarma o una detección fallida. La forma adecuada para mostrar el funcionamiento del sistema es generar una curva que permita ver el comportamiento en los diferentes puntos de operación. El punto de

operación de un sistema es un parámetro que puede ser ajustado para llevar a cabo la toma de decisiones.

La curva ROC (Receiver Operating Characteristics) ha sido usada tradicionalmente para ajustar el punto de operación de un sistema o para decir cuál es el mejor para la toma de decisiones. Generalmente se grafica la tasa de falsas alarmas en el eje horizontal, mientras la detección correcta se grafica en el eje vertical.

La curva DET (Detection Error Tradeoff) ha sido desarrollada como otra alternativa para apreciar el funcionamiento de un sistema de manera sencilla, esto debido a que se grafica la desviación normal en ambos ejes, dando un tratamiento uniforme para ambos tipos de error y emplea una escala logarítmica en ambos ejes, lo anterior mejora la forma de observar el funcionamiento de los sistemas o aproximaciones y produce gráficas que son cercanas a líneas rectas

- Curvas ROC: Es una de las formas tradicionales de representar los errores en problemas de clasificación. Como se puede observar en la figura 2.22.a, en estas curvas se muestra la variación de la Tasa de Falsos Positivos (eje horizontal) y la tasa de "verdaderos positivos" (muestras del cliente correctamente clasificadas, es decir, 1-"Tasa de Falsos Negativos" en el eje vertical) con respecto al umbral de decisión. El eje 'Y' y la recta y=1 pueden ser consideradas las asíntotas de esta curva, de manera que cuanto más se acerque la curva a las asíntotas, mejor es el comportamiento del sistema.
- Curvas DET: Es una modificación de la anterior que al lograr una representación cercana a la lineal, como se puede observar en la figura 2.22.b, permite una comparación visual entre sistemas más clara y fácil de realizar. Además, la distancia entre las curvas expresa las diferencias entre rendimientos de manera más significativa.

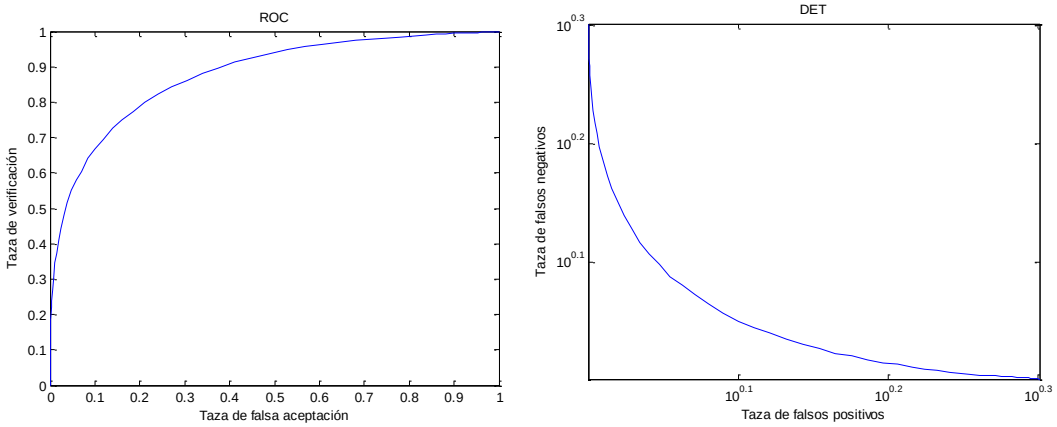


Figura 2.22: Curvas ROC y DET

En las curvas DET se logra una representación cercana a la lineal, mediante el uso de una escala no lineal en los ejes, se basa en el modelo de Gaussianas asociado a las distribuciones de las salidas del sistema. Es decir, que en vez de representar las Tasas de Falsos Positivos y Tasas de Falsos Negativos asociadas a un determinado umbral, como en las curvas ROC, se representa el número de desviaciones normales en la distribución normal estandarizada (media 0 y varianza 1) correspondientes a esas probabilidades.

La escala original en cada eje sería el número de desviaciones estándar que es lineal, sin embargo, tal y como se ve en la figura 2.22, esos valores son sustituidos por la probabilidad de error asociada, que da como resultado una escala no lineal. Cuanto más cercana esté la representación a una recta, más se acercará la distribución de resultados del sistema a la normal.

2.6 Propuesta inicial

Basándonos en lo que hemos visto sabemos que el problema es amplio, pero aunado a lo anterior existen aún problemáticas no tratadas como son:

- Grabaciones con ruido: el ruido enmascara la señal de voz de manera irremediable, el detectar que parte de la señal es voz y que parte es ruido es difícil o imposible, en ocasiones el algoritmo puede confundirse e identificar la fuente de ruido y no al locutor

Para ejemplificar ésto presentare un pequeño ejemplo “armado”, en éste se presentara la palabra “Teléfono” en un espectrograma de banda estrecha, por ahora no es muy importante saber distinguir formantes o armónicos, basta con poner atención a las bandas más negras de la figura 2.23, estas estructuras son las que permiten identificar a un locutor.

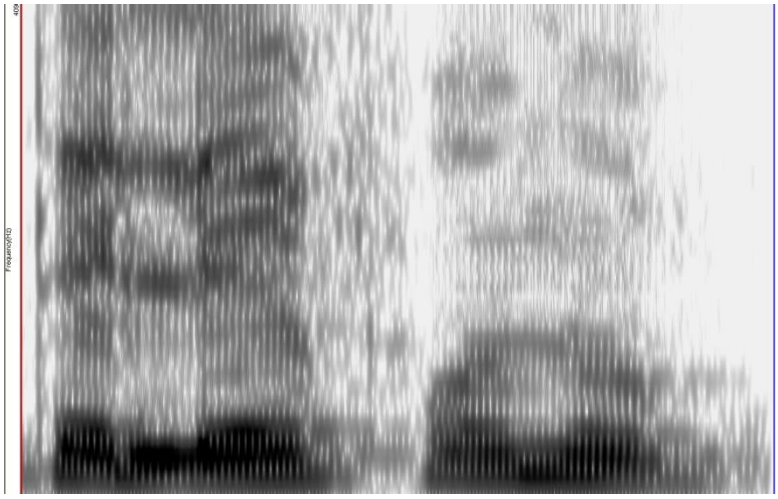
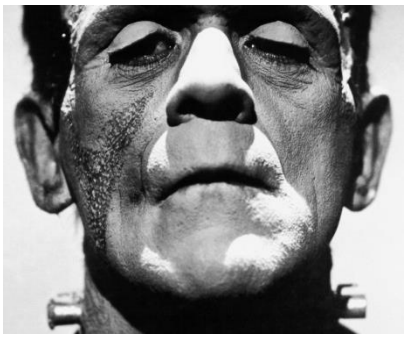


Figura 2.23: Espectrograma de la palabra "Teléfono"

Ahora presentaremos la misma palabra pero con un segundo locutor simultáneamente diciendo la palabra “Casa”, como se puede ver y se resalta con los círculos rojos de la figura 2.24, no es posible distinguir cuando habla una persona u otra, menos aún que porción corresponde a cada locutor, podría mejorarse un poco la distinción entre ambos locutores por medio de un cepstograma o por técnicas aún más complejas, pero en general este material no sería útil, ya que se modelarían características de ambos locutores en un sólo modelo, creando un Frankenstein biométrico, algo similar ocurre con el ruido.



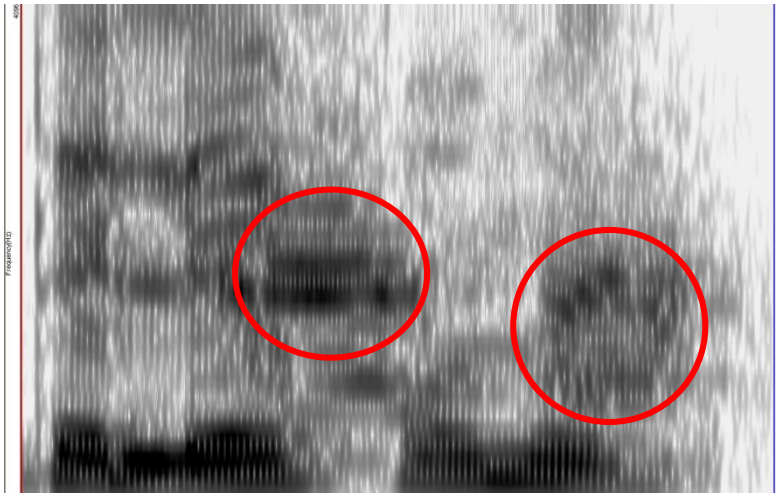


Figura 2.24: Palabra "Teléfono" con la palabra "Casa" de fondo superpuesta

En la figura 2.25 se muestra de nuevo la misma palabra "Teléfono" de la figura 2.23, pero esta vez se suma a la grabación música en un nivel apenas audible desde el punto de vista perceptual, sin embargo la música enmascara toda la información, no es fácil distinguir algunas de las estructuras del habla y otras más son totalmente destruidas u opacadas bajo la música, este es el efecto adverso que causa el ruido.

La identificación forense de locutores presenta grandes retos, ruidos de autos, motos, perros, pájaros, otras personas, música y un sinnúmero de fuentes no controlables de ruido, que se suman a las grabaciones que se trabajaran, ésto dificulta o hace imposible en ocasiones la correcta interpretación de un material, incluso cuando se trata de un análisis pericial ejecutado por un perito con años de experiencia.

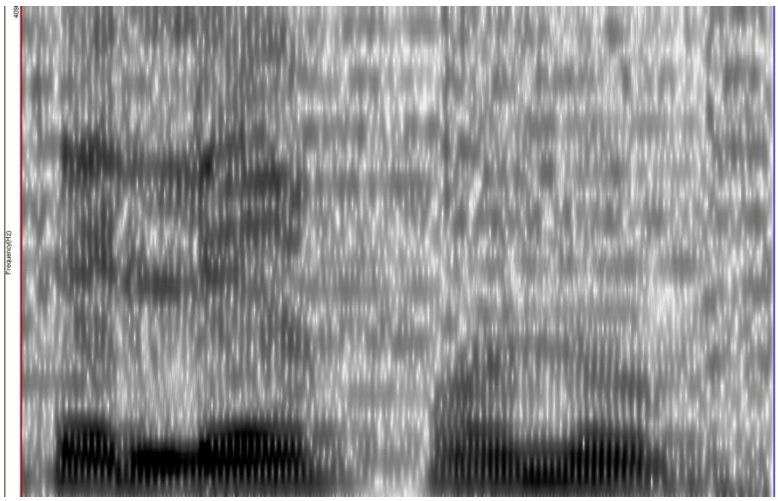


Figura 2.25: Palabra "Teléfono" con una muy tenue música de fondo

En este sentido se sugiere enfáticamente el segmentar las grabaciones en miras de remover la mayor cantidad de ruidos o voces ajenas, ya que acciones tales como el filtrado pueden resultar destructivas, imaginemos tratar de remover el velo de la mujer con Photoshop, podremos hacer un muy buen esfuerzo, pero el velo tapa características básicas que permitirán distinguirla, como la nariz y la boca, por lo tanto el filtrado de la fotografía solo "inventaría" información que no estaba en la muestra original y que no sabemos qué tan apegada esté a la realidad.

Lo que debemos hacer es intentar preservar las mejores porciones para su análisis, en los casos donde esto no sea posible por cuestiones de la calidad o cantidad de material, éste podrá usarse ya sea en el análisis biométrico o en el estudio por expertos, pero siempre teniendo en cuenta que la posibilidad de error se incrementaré significativamente.

- Cambio de canal: el cambio de canal altera la voz misma, éste es un filtro digital que puede llegar a modificar de buena manera la voz humana, presentándose al igual que en el caso pasado la falsa identificación del canal en lugar del locutor.

En este caso, se tiene que explicar un poco como funciona un canal de comunicación y como afecta éste a la calidad del audio, en la ingeniería la salida de un sistema lineal está dada por la convolución de la entrada con la respuesta del sistema

Recordemos que la convolución es un operador matemático que transforma dos funciones en una tercera función que en cierto sentido representa la magnitud en la que se superponen la entrada del sistema y una versión invertida y trasladada de la segunda.

$$x(t) * h(t) = y(t)$$

Figura 2.26: Sistema de comunicaciones

En este diagrama, x(t) es la voz que produce el locutor sin ningún canal de por medio, como cuando charlamos con otra persona frente a frente, y(t) es la voz que se transmite después de pasar por un sistema h(t), ésta es una combinación de ambos elementos, la voz original y los efectos del canal, h(t) representa al canal de comunicación, sea éste muy simple o muy complejo, para mostrar ésto usaremos un ejemplo muy simple.

Tomaremos una sola ventana de voz, digamos con una longitud de 256 datos, el resultado de tal transformación se puede apreciar en la figura 2.27, recordemos, el resultado es simétrico, solo se toman de 1 a 128 para la gráfica.

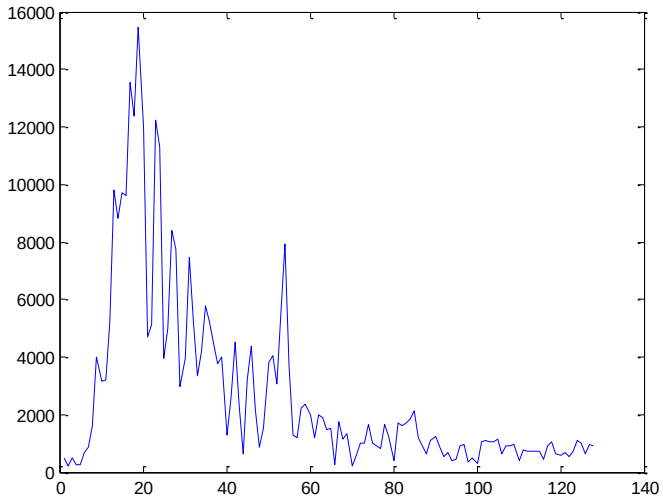


Figura 2.27: Espectro instantáneo de habla

Para este ejemplo también nos apoyaremos del teorema de la convolución, mismo que se muestra en la ecuación 5, ésta nos dice que la transformada de Fourier de la convolución de dos

funciones es igual a una constante k que multiplica a la transformada de Fourier de cada una de las funciones, para simplificar pensemos que k=1 y esto queda fuera, por lo que podemos decir lo siguiente, lo que es una convolución en el dominio del tiempo, es una multiplicación en el dominio de la frecuencia.

$$\mathcal{F}(f * g) = k \cdot \mathcal{F}(f) \cdot \mathcal{F}(g) \quad \text{Ecuación 5}$$

Haciendo uso del teorema de la convolución y sabiendo que el espectro que se muestra en la figura 2.27 ya está en el dominio de la frecuencia, sólo nos falta definir un sistema h(t) y transformarlo también al dominio de la frecuencia.

Si también simplificamos un poco la complejidad del canal de transmisión y lo planteamos como un filtro paso banda que sólo transmite la parte de la señal más "necesaria" nos aproximamos de manera idealizada a cómo se comporta un sistema real, usaremos una ventana de Tukey para representar al sistema ya en el dominio de la frecuencia, ésto quedaría como se ve en la figura 2.28.

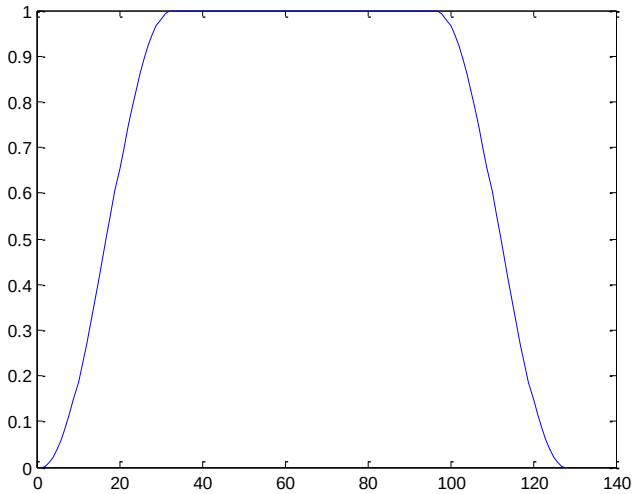


Figura 2.28: Modelado de la respuesta de un sistema de comunicaciones idealizado

Ahora basta con multiplicar ambas funciones para tener un resultado de su convolución.

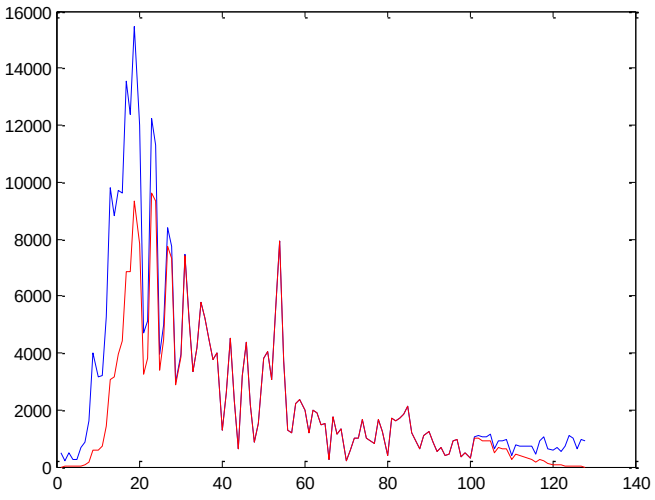


Figura 2.29: Resultado de la convolución

Aquí vemos en rojo el resultado de la convolución y en azul la señal original, a simple vista se ve que la señal no es igual, para complicar las cosas un poco más sabemos que un sistema de comunicaciones no será ideal, para darnos una mejor idea veremos en la figura 2.30 la gráfica de comportamiento en frecuencia de dos micrófonos, el PG48 y el SM7B de la marca Shure.

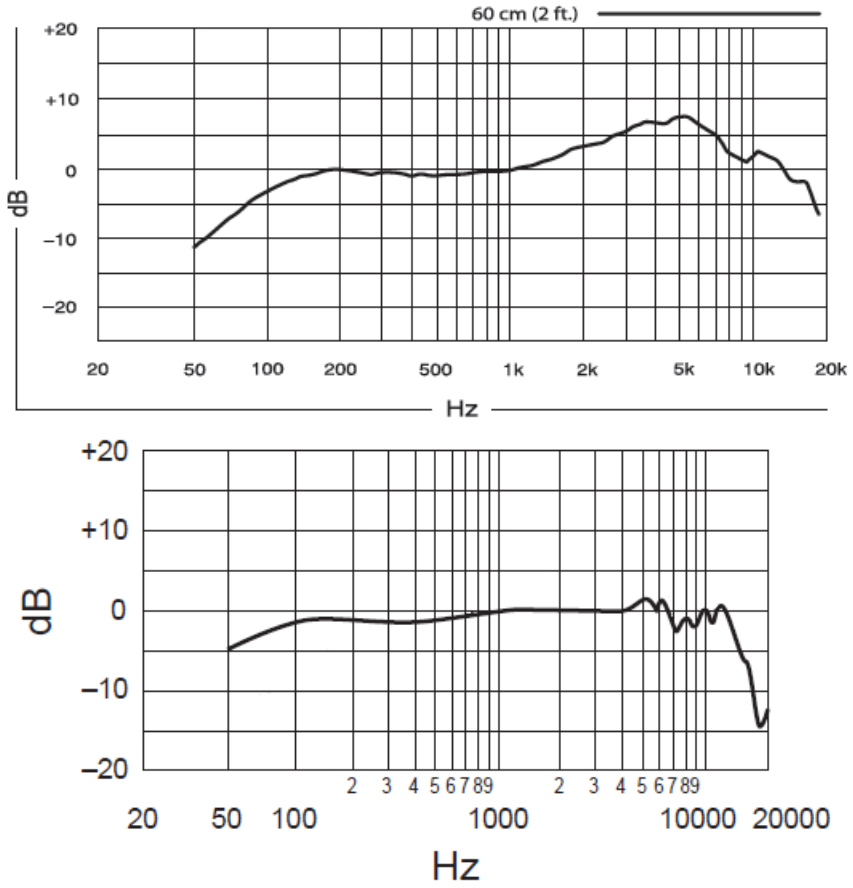


Figura 2.30: Gráfica de comportamiento en frecuencia de dos micrófonos

El primero es un micrófono de poco menos de 100 dólares, el segundo es un micrófono de más de 500 dólares, es el tipo de micrófono que se usa en cabinas de grabación profesionales, como se puede apreciar todos los micrófonos generan un efecto paso banda.

Este efecto es por limitantes técnicas, recoger señales por debajo de 50 [Hz] es sumamente difícil y no se justifica ya que hay “poca” información en dicha región, por encima de los 5 [kHz] pasa lo mismo, un micrófono sube mucho de precio al tener una gráfica de respuesta en frecuencia más “plana” y que tenga mejor respuesta arriba de los 5 [kHz].

Con esto imaginemos la calidad del micrófono de nuestro teléfono casero o de nuestro celular, esto sin tener en cuenta caídas y desgaste por el uso normal, además agreguemos un esquema más “real” de comunicación, convertidores analógico-digital, redes de comunicaciones, ruido eléctrico, pérdidas de información, convertidores digital-analógico y el equipo con el que grabamos nuestra conversación, algo similar al esquema que se muestra en la figura 2.31.

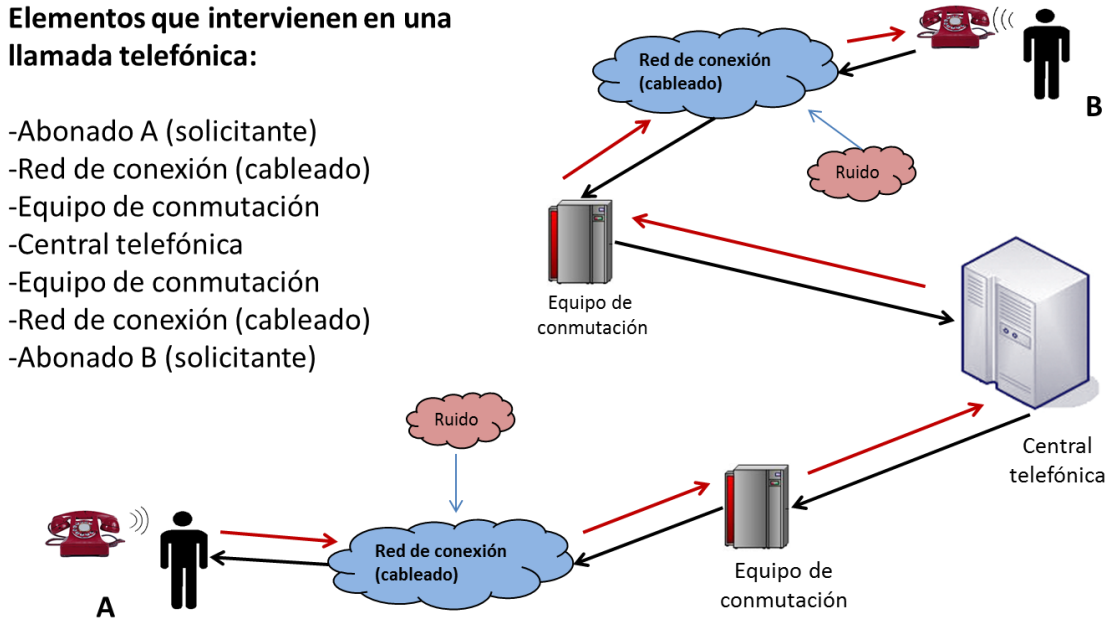


Figura 2.31: Esquema simplificado de una red telefónica

Todo lo que está entre el locutor que queremos identificar y nuestro archivo de audio ya digitalizado es el sistema h(t), además tenemos canales de comunicación como los de la telefonía celular, todos pensamos que escuchamos en nuestro celular a la persona que habla en el otro extremo, pero esto no es enteramente cierto.

En un esquema sobre simplificado de cómo trabajan los CODECS (abreviatura de codificador-decodificador) de telefonía celular, podríamos decir que se obtiene en el codificador un filtro inverso por técnicas como LPC para "anular" la mayoría de la señal de voz con una simple resta (similar a los sistemas de cancelación de ruido de algunos audífonos, automóviles, aviones, etc.), el restante ruido es entonces codificado en MP3 (sistema de compresión de audio con pérdidas), para después ser empaquetado y transmitido. (No se toman en cuenta mecanismos como los de detección del habla).

En el extremo receptor llegan algunos paquetes, casi nunca todos, y a partir de esto que llega se reconstruyen los faltantes, una vez que se tienen los paquetes se extra el filtro LPC, se le suma el ruido que se codifico en MP3 y listo, tenemos todos los elementos para reconstruir la voz, por ejemplo, en el algoritmo CELP de 4.8 [kbps] se usan 144 [bits] por cada 30 [ms] de voz en un esquema un poco más complejo de lo descrito anteriormente.

Esto parece una pesadilla desde el punto de vista biométrico, la señal es destrozada y modificada severamente, pero recordemos que el propósito de los canales de comunicación no es la biometría, es transmitir una conversación con la mejor relación inteligibilidad/costo para maximizar las ganancias de la empresa de comunicaciones.

Existen extensos estudios para determinar cuanta información es la mínima que debo transmitir para que aún sea inteligible el mensaje, recordemos que a mayor transmisión mayor costo, por eso se emplean detectores de actividad vocal para no transmitir nada cuando la persona no habla, por eso cuando nuestra contraparte no habla no sabemos si se cortó la comunicación o la

persona sigue ahí, no se transmite el ruido ambiental, esto es substituido por un silencio absoluto.

Por último, en este punto se muestra una grabación realizada directamente con un micrófono de calidad telefónica y después un par de ejemplos de lo que un códec hace a esta señal, que es sólo parte del procesamiento que sufre.

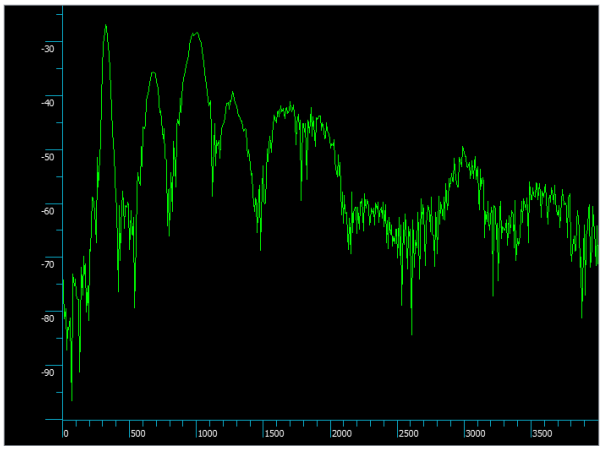


Figura 2.32: Señal original en grabación con un teléfono

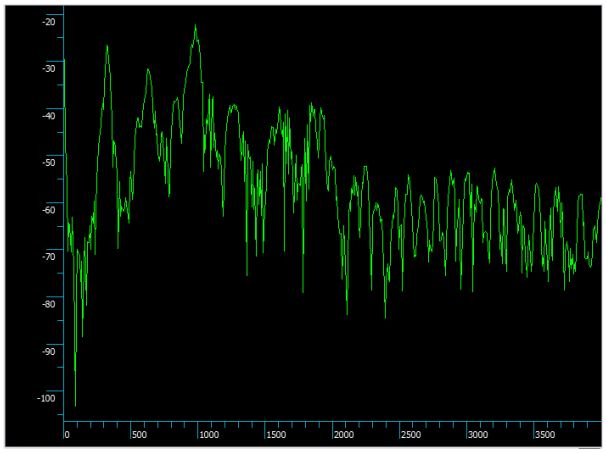


Figura 2.33: Señal después de un códec LPC10

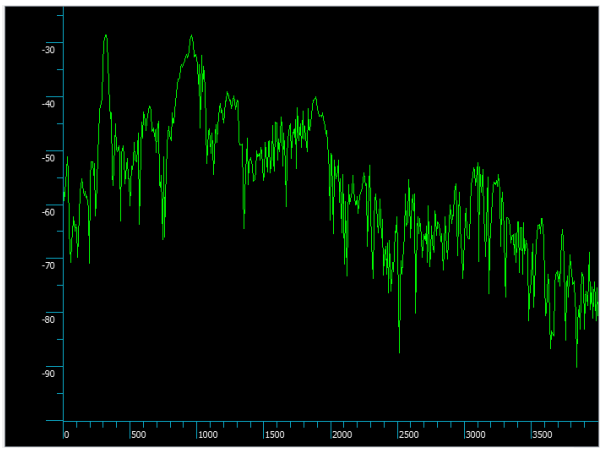


Figura 2.34: Señal después de un códec CELP

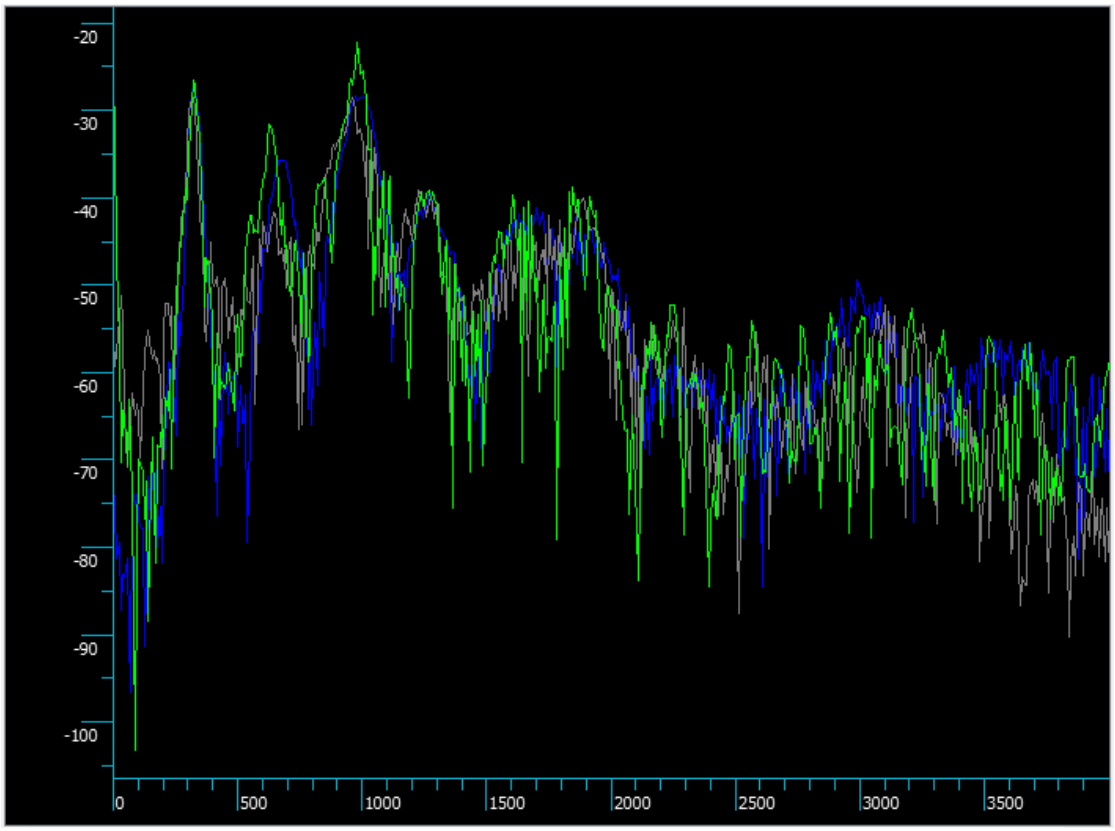


Figura 2.35: Comparativa de las tres señales

Podría ser que en las figuras 2.32, 2.33 y 2.34 no sea muy fácil notar las diferencias, en la figura 2.35 pongo las tres gráficas simultáneas en distintos colores, en azul tenemos la grabación original, en verde LPC10 y en gris CELP, como se puede ver la señal sufre cambios importantes, por este motivo un sistema biométrico podría equivocadamente identificar al canal y no al locutor.

- Incremento del error en la identificación con el crecimiento de la base de datos: todos los sistemas tienen un poder de discriminación limitado, si la tasa de error es del 1% quiere decir que en 100 audios identificara erradamente a una persona, pero en 100 millones de personas (población de un país) podría identificar erradamente a un millón de personas, algunas corrientes sugieren la segmentación de la población en subgrupos más pequeños para disminuir este impacto ya que el problema no tiene solución real

Como se puede consultar en diversas fuentes, aún los mejores sistemas del mundo en reconocimiento de locutores no logran una tasa de error igual (EER) menor al 2% en condiciones idóneas, al empeorar la situación rondan el 5% de error y en condiciones verdaderamente malas pueden llegar a un 20% de error, a nuestro pesar, las pésimas condiciones suelen ser lo común, es un locutor que no quiere ser atrapado, cambiará su voz hasta donde pueda, el estado de ánimo es totalmente distinto cuando comete el delito a cuando es capturado y emplea celulares o teléfonos públicos desde ubicaciones con mucho ruido, de la calle, automóviles, etc.

Aún tomando un EER de un 1% que no es realista, en una población de 100 millones de personas (menor a la población de México) esto significaría más de 1 millón de candidatos para

una identificación, como lo muestra la figura 2.36, un sistema que pudiera identificar a cualquier Mexicano ya sea por su voz, huellas dactilares o rostro requeriría un EER del 0.000001 % que es imposible en la actualidad para cualquier biometría.

Figura 2.36: Segmentación de la población

En este tenor una sugerencia general es segmentar la población, lo cual suena sumamente lógico, de acuerdo al INEGI (Instituto Nacional de Estadística y Geografía), aproximadamente el 50% de la población son mujeres y otro 50% son hombres, por lo tanto si requiero identificar a un hombre para que buscar contra las mujeres, no es lógico.

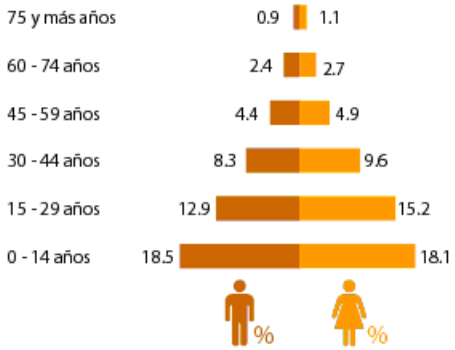


Figura 2.37: Distribución de la población

Entrando un poco más en esto, si requiero identificar a un hombre de aproximadamente 35 años de edad para que buscar en la población de niños y de adultos mayores, como se ve en la figura 2.37, la población se distribuye por edades, en el rango de 30 a 44 años solo hay 8.3 millones de hombres en todo el país, esto entre la superficie del país nos da 4.2 hombres de entre 30 y 44 años por kilómetro cuadrado, por lo que si lo limitamos a un solo estado o una sola ciudad podría ser que un EER del 2% sea suficiente.

El problema del razonamiento anterior es cómo segmentar la población, no siempre se puede saber por seguro si se trata de un hombre o de una mujer sin necesidad de una investigación más exhaustiva, como en el caso de la atleta "Semenya" que compitió como mujer en eventos deportivos, lo mismo se puede decir de la edad y más aún de la región de origen de la persona, el acento puede ser falso.

Otros expertos sugieren la segmentación automática de la base de datos en grupos de personas "similares", esto es conocido como la generación de grupos o clusters por su término en inglés,

sin embargo a mi personal punto de vista esto es riesgoso, podríamos perder por completo no sólo a un locutor sino a toda una porción de la población si nuestro algoritmo de segmentación falla, después de todo este proceso de agrupamiento no es otra cosa que una discriminación automatizada y esto es muy similar a lo que hace el motor biométrico, por lo tanto es engañarnos a nosotros mismos, es correr dos veces el mismo proceso con distintos umbrales y a distintos tiempos, pero el problema permanece igual.

De todo esto, es evidente que no es factible crear un sistema de identificación biométrica por voz que compita con sus similares que llevan hasta 20 años de ventaja en la investigación y el desarrollo en cuestión de pocos meses, dato que es importante mencionar y recalcar para tener expectativas reales.

La propuesta concreta es crear un sistema que esté compuesto al menos de los siguientes módulos que pueden ser interconectados entre sí:

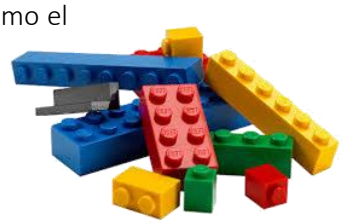
- Módulo de detección de actividad vocal: éste tiene la funcionalidad de remover las porciones con pausas o silencios dejando únicamente la voz pura para su estudio, este módulo debe contemplar al menos dos metodologías como podrían ser:
    - Detección por energía
    - Detección por frecuencias
- Módulo de filtrado digital de pre-énfasis: este módulo tiene la función de disminuir la diferencia energéticamente entre los distintos componentes espectrales de la voz, ya que por su naturaleza misma ésta tiende a caer con el aumento en frecuencia, esto por diversos factores que generan pérdidas, el módulo debe contemplar una configuración básica al menos con:
    - Selección del parámetro alfa variable
- Módulo de ventaneo: éste tiene la función de segmentar la señal de voz en pequeñas porciones o ventanas de acuerdo a las tendencias actuales que indican que el análisis de corto tiempo es mejor, ya que la señal tiende a ser cuasi-estable, debe contemplar al menos:
    - Tamaño de la ventana variable
    - Traslape de ventanas variable
    - Diversos tipos de ventanas matemáticas
- Módulo de extracción de energía: el módulo obtendrá la energía de cada ventana previa su transformación al dominio de la frecuencia, este parámetro nos indica la fuerza de la señal y su comportamiento en el tiempo, este módulo no tiene parámetros propios ya que depende de las ventanas mismas.
- Módulo de transformación al dominio de la frecuencia: el módulo realizará la conversión del dominio del tiempo al dominio de la frecuencia de la señal previamente separada en ventanas, este módulo carece de parámetros propios ya que depende de los parámetros de la ventana misma.
- Módulo de banco de filtros: módulo previo a la obtención de los coeficientes cepstrales, se requiere aplicar a la señal un banco de filtros que separara la señal auditiva en bandas frecuenciales, este banco debe poder aplicar diversos tipos de filtros, contemplando al menos los siguientes parámetros:

  - Número de coeficientes variable
  - Escalas lineales y logarítmicas
  - Traslape de los filtros variable
  - Filtro de corte paso banda
  - Diversos tipos de filtros
- **Módulo de transformación cepstral:** módulo que realiza la segunda transformación de la señal para obtener los coeficientes cepstrales (cambio del dominio de la frecuencia al dominio cepstral) es sumamente similar a la anterior transformada de Fourier, sin embargo, existen también tendencias que indican que es más eficiente el uso de otras transformaciones, razón por la que es diferente y debe de incorporar al menos:
  - Transformada discreta de Fourier
  - Transformada coseno discreta
- **Módulo de aproximación polinomial:** es una recomendación a nivel general el tener en cuenta la información dinámica de la señal auditiva, ya que ésta no está presente en los coeficientes cepstrales, en este sentido se debe implementar un módulo que obtenga la primera derivada que es la razón de cambio de la voz (velocidad) y la razón de cambio de ésta (aceleración), estos componentes son conocidas como coeficientes delta y delta-delta, esto se llevará a cabo mediante una aproximación polinomial, ésta debe contemplar al menos:
  - Orden de la aproximación variable
- **Módulo de ensamble de los coeficientes:** los coeficientes pueden tener componentes opcionales, se tiene la recomendación general de que los coeficientes delta y la energía benefician al proceso de identificación, sin embargo, puede resultar útil poner esto a prueba, se debe contemplar:
  - Agregar coeficientes delta
  - Agregar coeficientes delta-delta
  - Agregar energía
- **Modelado:** en general el resultado de la obtención de la información del habla es demasiado grande, contiene mucha información perteneciente al mensaje, al acento, al idioma y a diversos factores que por el momento no nos interesan, el modelado pretende remover esta "paja" para lograr discernir quien habla, se pretende implementar una técnica, como podría ser:
  - Cuantización vectorial

Con los mencionados módulos se pueden construir múltiples modelos para la identificación de locutores, incluso se puede utilizar más de un método a la vez en vías de lograr la mayor eficiencia posible, con lo ya propuesto, el número de combinaciones de los parámetros de trabajo a optimizar puede rebasar los miles de millones de configuraciones.

Puesta a punto

El sistema modular que se describió antes se pondrá a prueba con una base de datos experimental en la que se sabe de antemano cuales archivos pertenecen a una misma persona, en ésta existen al menos dos grabaciones de cada locutor, esta base será conocida como el corpus de prueba y calibración, mismo que será utilizado de manera reiterativa en distintas configuraciones.



La finalidad de realizar estas identificaciones en archivos de los que se conoce de antemano el resultado, es obtener las gráficas de calibración de la solución, mismas que permitirían comparar objetivamente una configuración contra otra, posibilitando evaluar cuál es la mejor configuración o mezcla de éstas para mejorar la eficiencia de la solución.

De antemano, es prácticamente imposible saber cuál sería la configuración o módulos que se deben desarrollar y emplear, si esto fuera conocido sería relativamente fácil desarrollar una aplicación más rápida y quizás más eficiente que la que aquí se propone, por esto se creará un juego de bloques intercambiables en un esquema tipo Lego, que es el juguete infantil que nos permite crear todo tipo de objetos con bloques genéricos.

Esta prueba pretende ser la aportación original de la presente tesis, existe una buena cantidad de tutoriales y de proyectos sobre la identificación de locutores en internet, pero no existe ningún trabajo que incorpore una explicación detallada de la historia, el estado del arte, las problemáticas, el desarrollo y la calibración de un motor biométrico por voz.

3. PARAMETRIZACIÓN DE LA VOZ

Ahora que tenemos una buena idea de cómo se pueden extraer parámetros de la voz humana y sobre todo, que sabemos que representan esos parámetros y como son usados en la representación del locutor mediante un modelo, pasaremos a la implementación de un extractor de parámetros de acuerdo a las tendencias a nivel mundial que se tienen en la materia (Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier Merlin, Ortega-Garcia, Petrovska-Delacrétaz y Reynolds, 2004).

3.1 Parametrización por coeficientes cepstrales

La parametrización es la transformación de una señal de voz, que como ya hemos mencionado contiene una gran cantidad de información, en una nueva representación más compacta, con menor redundancia y por lo tanto, más adecuada para ser modelada, mismo que se implementará por medio de la metodología más recomendada actualmente y que ha logrado los mejores resultados de acuerdo al NIST (National Institute of Standards and Technology).

3.1.1 Filtro pre-énfasis

Para dar inicio a la obtención de estos parámetros o MFCC (Mel-frequency cepstral coefficients), como mejor se les conoce, se recomienda aplicar un filtro de pre-énfasis, este filtrado no destructivo aplicado a la voz humana, pretende aumentar o amplificar en el dominio de la frecuencia aquellas componentes de menor magnitud con respecto a las de mayor nivel energético.

De manera natural, las componentes de frecuencias altas tienden a tener menor energía que las componentes de frecuencias bajas debido a las características del aparato fonador humano, se compensa esto mediante un filtrado que eleva las frecuencias altas para tratar de colocarlas a un nivel semejante de aquellas componentes localizadas en las frecuencias bajas.

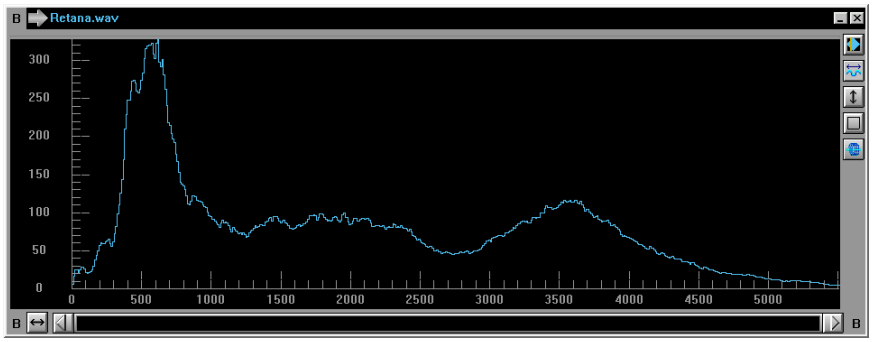


Figura 3.1: Señal de voz sin procesar

Para ejemplificar esto, se muestra en la figura 3.1 el comportamiento promedio del espectro con un locutor cualquiera que habla por poco más de dos minutos, como podemos ver, la energía en los rangos de frecuencia de 300 [Hz] a poco más de 1,000 [Hz] es prácticamente tres veces mayor que cualquier contenido después de dicha franja.

Para compensar este efecto adverso a los fines de identificar al locutor, se aplica un filtro paso altas de primer orden definido por $H(z) = 1 - \alpha z^{-1}$, donde α estará en el rango [0,1], en este caso y sólo para ejemplificar usaré un valor de 0.95 y un filtro FIR (Finite Impulse Response) mediante MATLAB para su implementación.

```
Y=filter([1 -0.95],1,x);
```

Mediante esta simple instrucción MATLAB, creará para nosotros un filtro paso altas que se aplicará a la señal de audio contenida en la variable 'x', pero lo realmente interesante será ver los resultados sobre el archivo previamente mostrado.

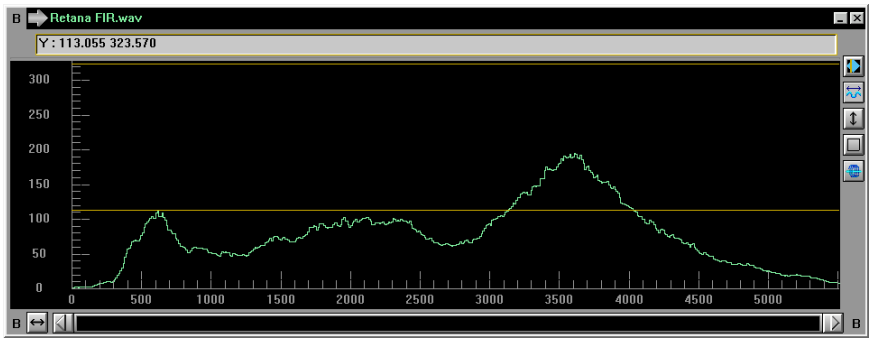


Figura 3.2: Señal de voz después del pre-énfasis

En la figura 3.2 vemos que ahora al contrario la parte más energética está en las frecuencias altas, alrededor de los 3,500 [Hz], pero éste es meramente un ejemplo, no quiere decir que usaremos un valor de 0.95 para todo este trabajo. A continuación, se muestran ambas gráficas para poder observar que la señal es la misma, sólo cambia su distribución energética en el espectro.

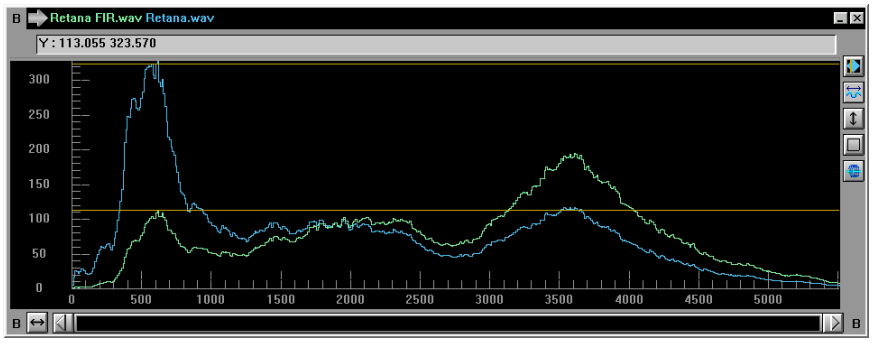


Figura 3.3: Señal de voz con y sin pre-énfasis

Ahora que tenemos el filtro de pre-énfasis, tenemos ya el primer paso para lograr un sistema automatizado de identificación de locutores, por lo que mostraré el primer bloque del diagrama:

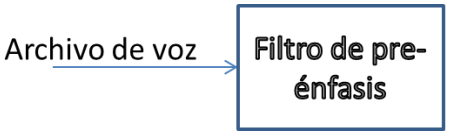


Figura 3.4: Diagrama del pre-énfasis

Durante el resto de este trabajo se utilizará una función de MATLAB muy simple que facilita el uso de este filtro en esquemas repetitivos, esta función se puede encontrar en el anexo 1.

De lo anterior, debe surgir la primera duda, tenemos el valor de α, pero no es claro que valor de 0 a 1 debemos tomar o cuál sería el valor óptimo para los fines que nos interesan, motivo por el

cual por ahora lo dejaremos como una incógnita, ya que seguiremos un proceso científico para determinar su valor y que tanto puede afectar los resultados su selección.

Debe también tenerse claro que puede ser una opción el no aplicar el filtro si éste no resultara útil, mismo que también se verificara más adelante, ya que después de todo, no es indispensable este proceso de homologación energética de las bandas del espectro.

3.1.2 Ventanas

Continuando con la obtención de los MFCC tenemos el aspecto de las ventanas que fue cubierto en mi tesis de licenciatura, sin embargo, veremos un poco más de este aspecto, lo que es claro es que una transformada de Fourier discreta de toda nuestra señal no es viable y tampoco sería representativa de nada en realidad.

La señal de voz es sumamente cambiante, esto lo podemos apreciar en la figura 3.5 que nos muestra poco más de dos segundos de habla, ésta cambia su comportamiento de manera continua, por lo que si tomará como ventana toda esta información y obtengo el comportamiento de mi señal en el dominio de la frecuencia la pregunta sería ¿Qué significa esto?

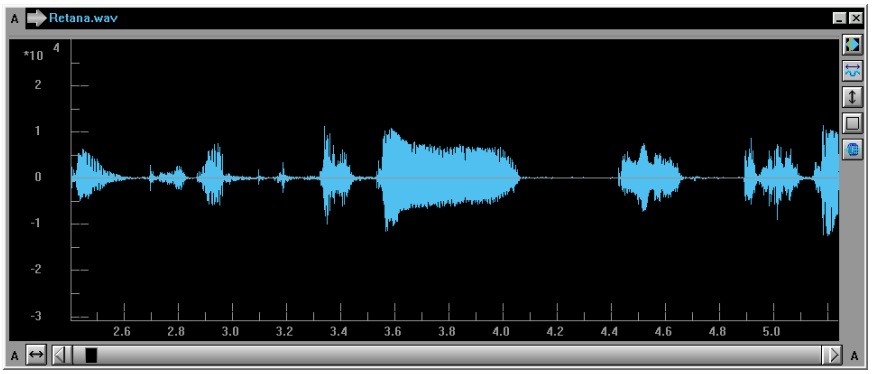


Figura 3.5: Gráfica de forma de onda de 11.89 segundos de habla

La respuesta es simple, significa poco o nada, para ejemplificar esto en la figura 3.6 mostramos el espectro de 11.89 segundos de habla, aquí vemos la distribución de frecuencias de este periodo en la señal como si fuera una sola pieza, pero en realidad esto no significa nada, el porqué es simple, pero sin embargo lo explicaré a través de una analogía.

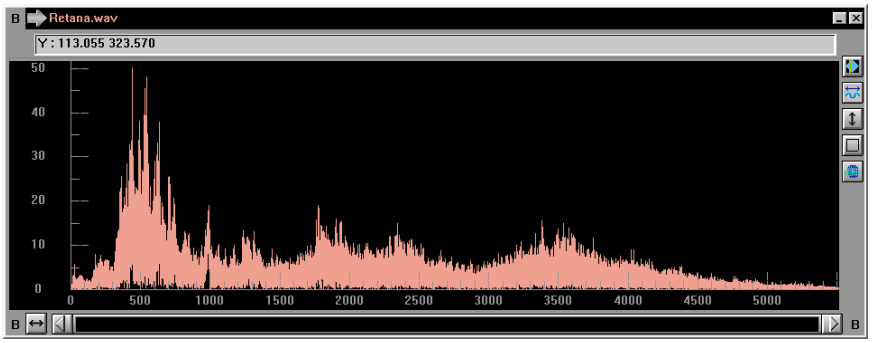


Figura 3.6: Espectro de los 11.89 segundos de habla

Si queremos comprender que materias representan mayor dificultad para un alumno de primaria podemos simplemente analizar sus calificaciones, podremos ver quizás muchos 10 en

Español, significan que no tiene problema en dicha área, pero calificaciones de 6 y 7 en Matemáticas pueden significar una dificultad sobre dicha materia.

Ahora pensemos que no podemos ver las calificaciones de un niño o niña, sino únicamente la información relativa a todo un grupo de 20 niños, aquí un promedio de calificaciones bajas en Matemáticas puede ser más un indicativo de que el maestro no se desempeña adecuadamente contra un problema de aprendizaje de todos los niños del grupo, después de todo, en un grupo de 20, no todos tendrán problemas con la misma materia.

Si elevamos esto a que sólo podemos ver el total de 5 grupos del mismo grado escolar con 100 niños esto podría quizás atribuirse a un material educativo no adecuado en esa escuela, si de nuevo lo elevamos a todos los grupos de 5º de primaria de todo el estado podría verse como un problema con el nivel educativo requerido por las autoridades, pero de ese análisis de miles de niños es prácticamente imposible saber si un niño en particular tiene problemas para aprender una materia.

Lo mismo sucede con el habla, en periodos largos nos puede orientar sobre aspectos tales como el canal de grabación, sobre la calidad del micrófono y algunos otros datos, pero revela poco de lo que queremos en este momento, nosotros requerimos modelar como habla un locutor y sus puntos que lo individualizan o caracterizan.

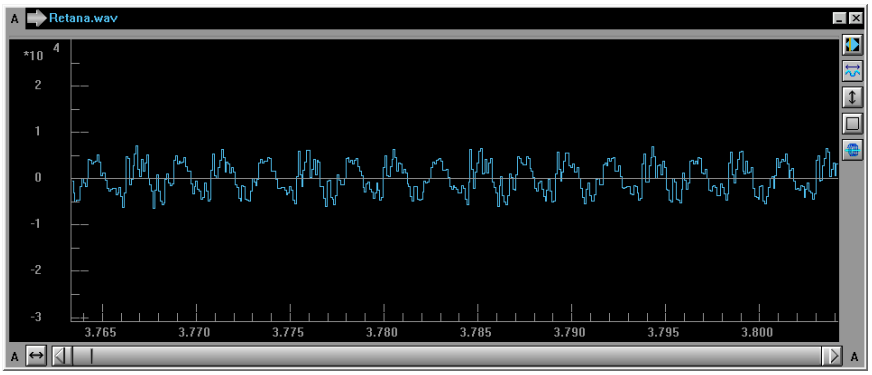


Figura 3.7: Gráfica de forma de onda de la señal de voz en un periodo corto

En este sentido y como no es difícil apreciar si vemos la figura 3.7, el habla humana es mucho más estable en periodos cortos, incluso se puede decir que ésta es cuasi-estática (Reynolds, 2002), mismo que la hace más útil para caracterizar el habla humana y por ende para poder crear su modelo de manera eficiente.

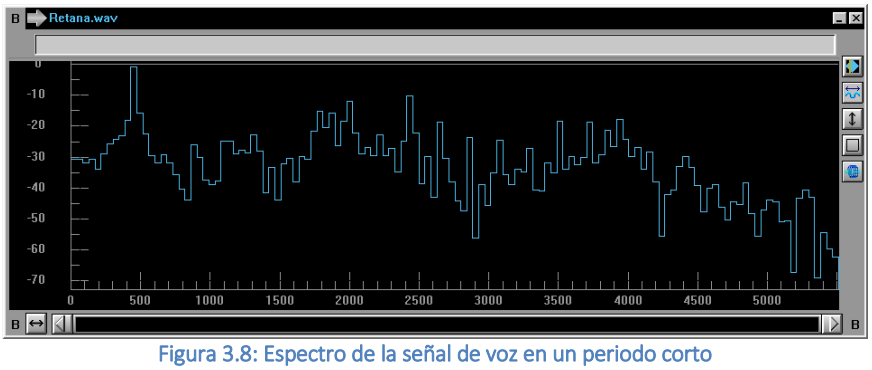


Figura 3.8: Espectro de la señal de voz en un periodo corto

Como se ve en la figura 3.8, esta gráfica desde el simple punto de vista visual contiene mucho menos "ruido" y se aprecia claramente que la señal armónica produce una representación visual donde se pueden apreciar cuatro o cinco formantes o máximos relativos, que son aún más claras al introducir en la misma gráfica la envolvente del espectro a partir del audio con pre-énfasis, como se muestra en la figura 3.9.

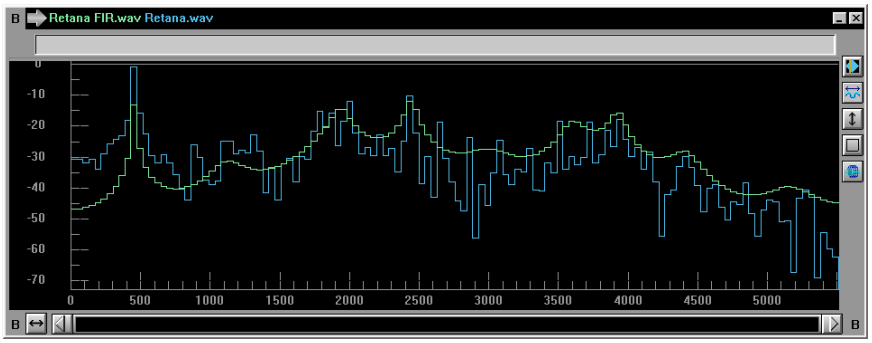


Figura 3.9: Espectro de la señal de voz en un periodo corto con envolvente

Ahora bien, una ventana matemática es una función que vale cero fuera de un intervalo determinado por el usuario, siendo el ejemplo más claro una ventana rectangular, misma que vale cero fuera del intervalo y un valor que normalmente es 1 dentro de la misma, por lo que no "afecta" a la señal a la que se le aplica fuera de que recorta la señal a sólo los datos que quedan dentro de la ventana, su función está dada por la ecuación:

$$w(n) = 1 \qquad \text{Ecuación 6}$$

Su gráfica está dada por:

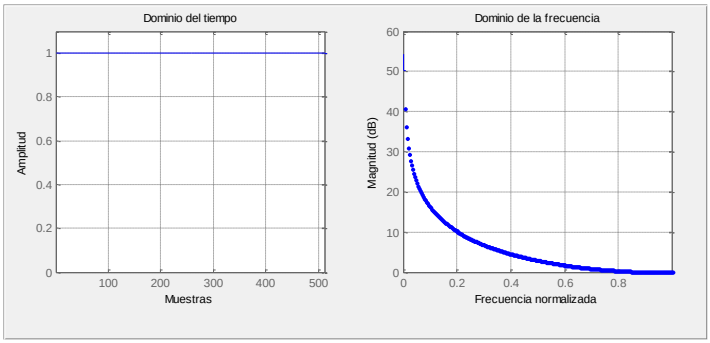


Figura 3.10: Ventana rectangular

A manera ilustrativa se mostrarán las gráficas que caracterizan el comportamiento de las principales ventanas en el dominio del tiempo y de la frecuencia, mismas que serán probadas a lo largo de este trabajo para analizar su efecto para fines de la identificación del locutor, pero no ahondando más en éstas, ya que ese no es el objetivo principal del trabajo.

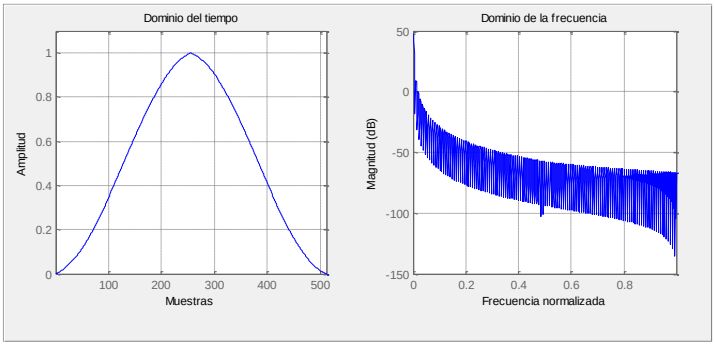


Figura 3.11: Ventana modificada Bartlett-Hanning

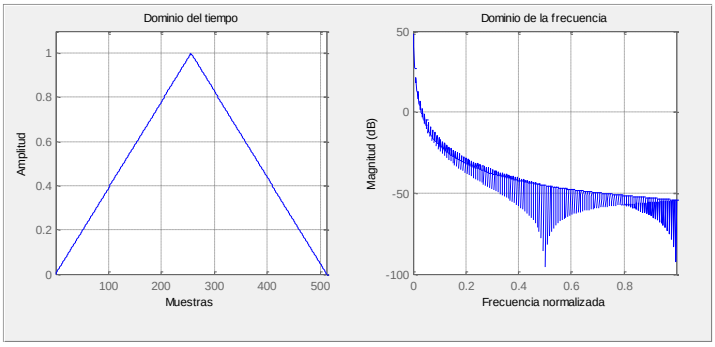


Figura 3.12: Ventana Bartlett

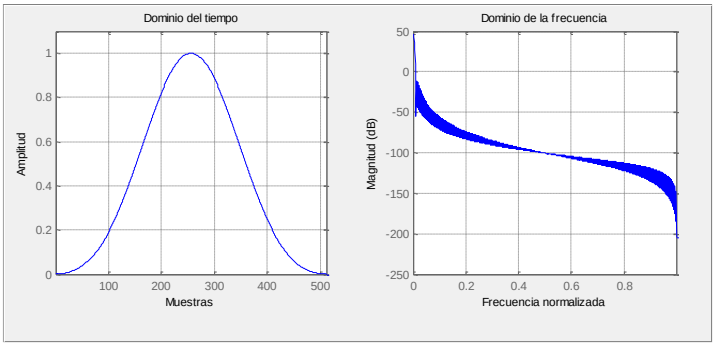


Figura 3.13: Ventana Blackman

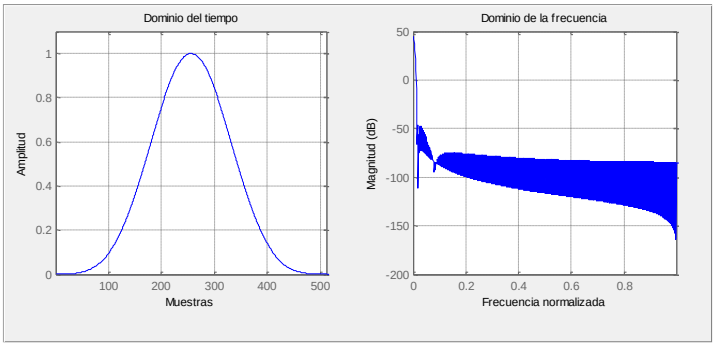


Figura 3.14: Ventana de mínimo 4 términos Blackman-Harris

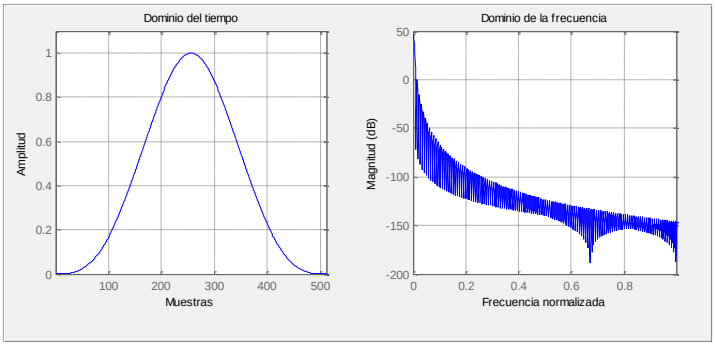


Figura 3.15: Ventana de Bohman

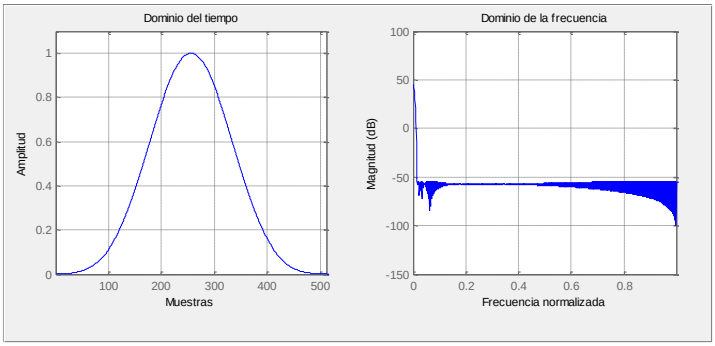


Figura 3.16: Ventana de Chebyshev

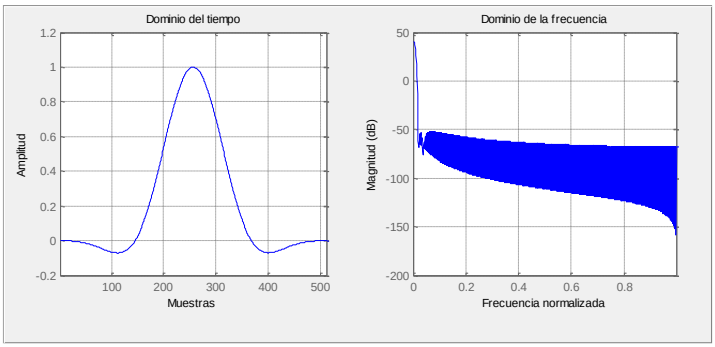


Figura 3.17: Ventana de parte alta plana

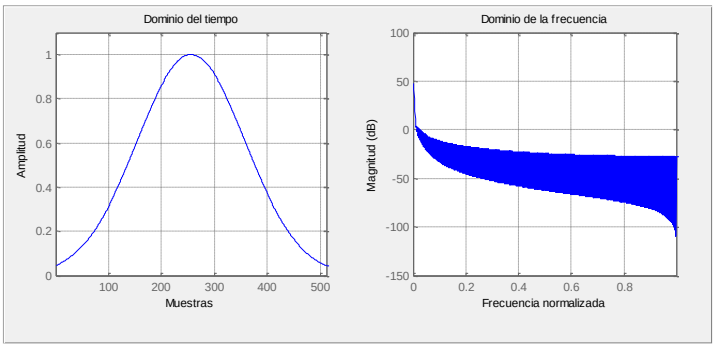


Figura 3.18: Ventana Gaussiana

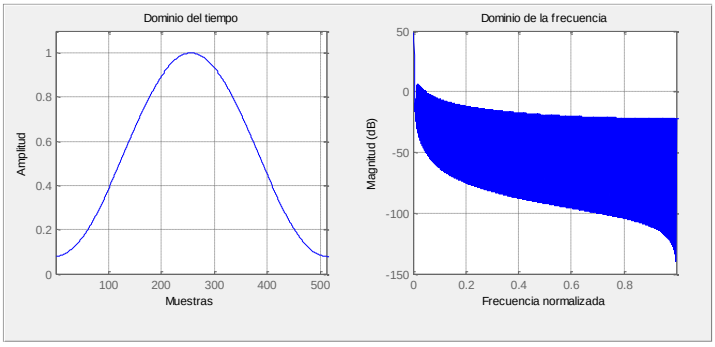


Figura 3.19: Ventana de Hamming

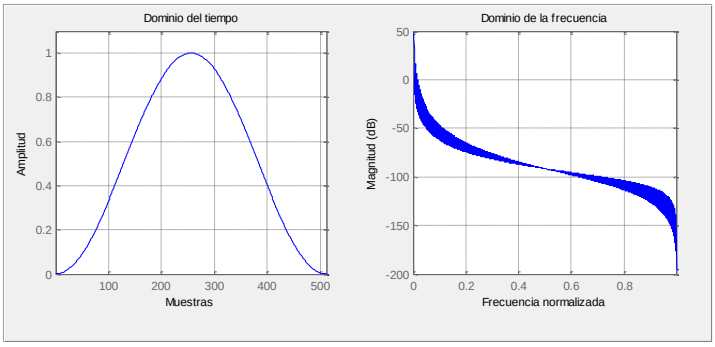


Figura 3.20: Ventana de Hann

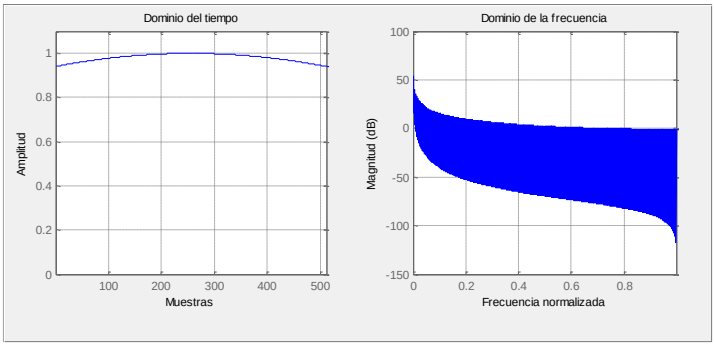


Figura 3.21: Ventana de Kaiser

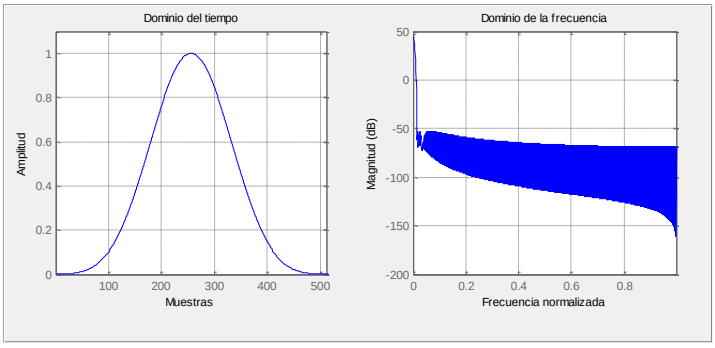


Figura 3.22: Ventana de Nuttall (Blackman-Harris de N puntos)

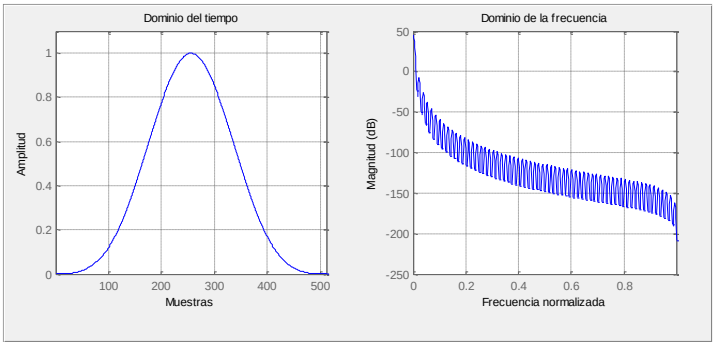


Figura 3.23: Ventana de Parzen

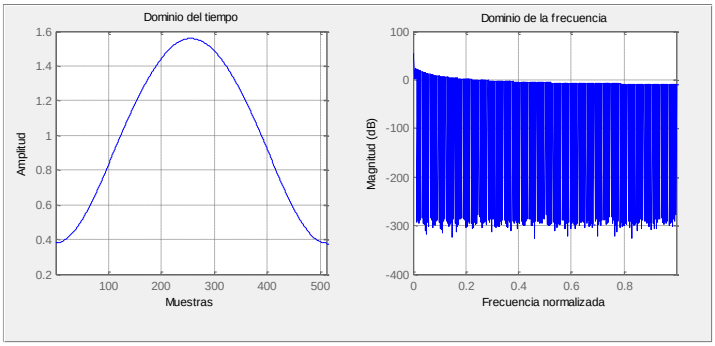


Figura 3.24: Ventana de Taylor

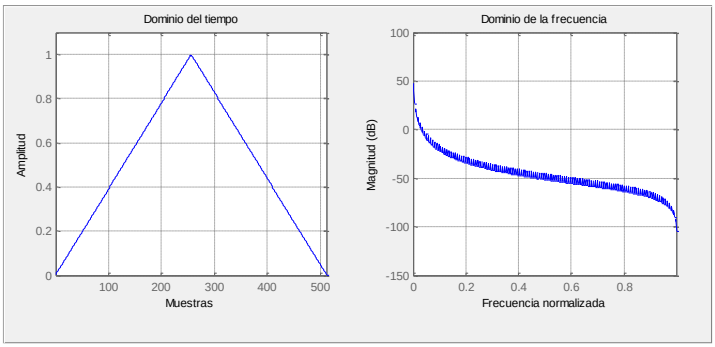


Figura 3.25: Ventana triangular

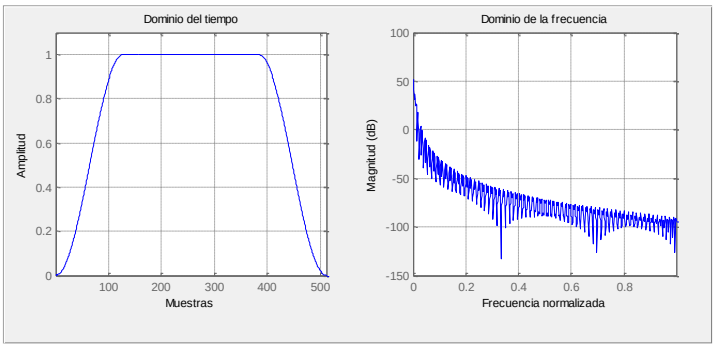


Figura 3.26: Ventana de Tukey

Como podemos ver, cada ventana ofrece un comportamiento distinto de la anterior, siendo regularmente la más recomendada la de Hamming o la de Hann para la identificación de locutores, pero se probará la eficiencia del sistema con todas las ventanas mencionadas para ver cuál nos ofrece los mejores resultados de manera empírica, ya que aunado a el gran número

de ventanas que se podrían emplear, algunas de ellas tienen parámetros que podemos modificar para que ésta se comporte ligeramente distinto.

Para visualizar mejor la diferencia entre las distintas ventanas, se muestra en la figura 3.27 una gráfica donde se tiene a un mismo tiempo las 17 ventanas antes mencionadas.

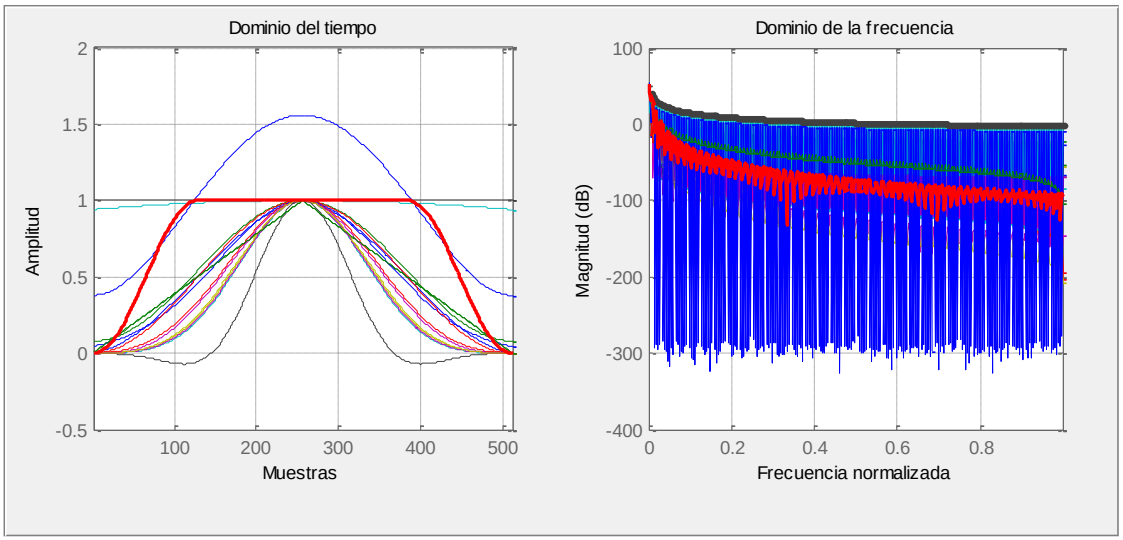


Figura 3.27: Múltiples ventanas

Cada ventana afectará el resultado final de maneras distintas, ya que esta información será la que transformaremos al dominio de la frecuencia, para ejemplificar esto, veremos la transformada de una simple suma de dos señales sinusoidales a 110 y 220 [Hz] como se muestra en la figura 3.28.

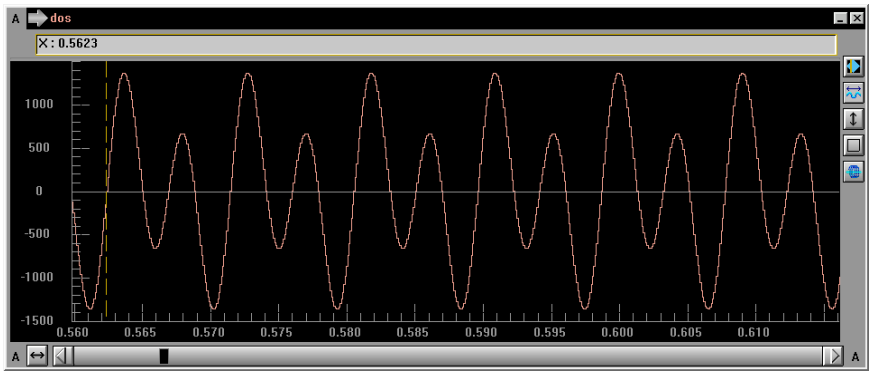


Figura 3.28: Suma de dos señales sinusoidales de 110 y 220 [Hz]

En la figura 3.29 se aprecia que la ventana cuadrada (lo que equivale a no usar ninguna ventana) introduce una mayor distorsión en la parte baja del espectro como se aprecia en la gráfica morada, las restantes gráficas azul, verde y rosa, son ventanas tipo Hamming, Hann y Nuttall respectivamente, éstas causan distintos comportamientos, pero en general, una menor distorsión y una mejor delimitación de las frecuencias que componen la señal, ya que se elimina o aminora el efecto borde al tomar sólo porciones de la señal para su análisis.

Recordemos que la distorsión que genera la ventana, es debido a que su aplicación es el resultado de la convolución de la transformada de la señal con la transformada de la ventana, ya que la multiplicación es aplicada en el dominio del tiempo.

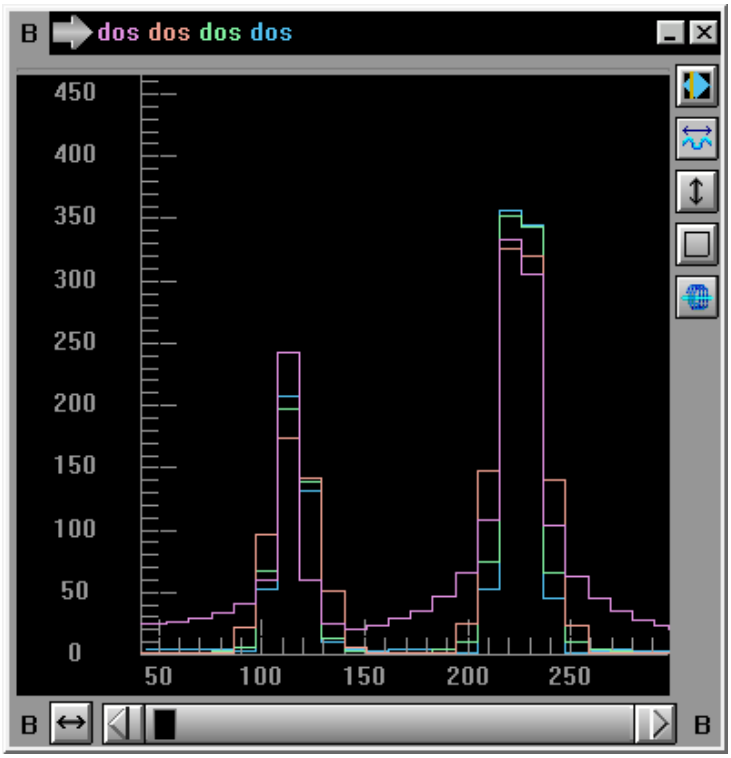


Figura 3.29: Análisis FFT de la señal anterior con distintas ventanas

La función que empleo para aplicar estas ventanas se encuentra en el anexo 1. Con esto podemos decir que tenemos el segundo paso, ahora nuestro diagrama de bloques luce así:

Figura 3.30: Diagrama de bloques con ventana

3.1.3 Transformación

Como el siguiente paso se requiere transformar la información del habla a un dominio que nos facilite su análisis, la transformada de Fourier es una herramienta muy utilizada en el campo científico y en la acústica en general, ya que transforma una señal de audio del dominio del tiempo al de la frecuencia sin alterar la información, sólo nos da una representación alternativa que nos permite descomponer una señal compleja en un conjunto de componentes de frecuencia, mismo que es sumamente útil en el estudio de señales estacionarias o cuasi-estáticas (Bernal, 1999).

En el caso de la voz que claramente no es una señal estacionaria, nos vemos obligados a tomar segmentos cortos donde se pueda considerar a la señal como estacionaria o cuasi-estática, mismo que nos posibilita su análisis mediante la descomposición de Fourier con un resultado coherente, pero como es evidente para analizar todo el audio completo se requiere la utilización de un análisis por segmentos subsecuentes de la señal de audio para poder observar la evolución de las frecuencias en la señal original, mismo que nos deja con el problema del tamaño de la ventana.

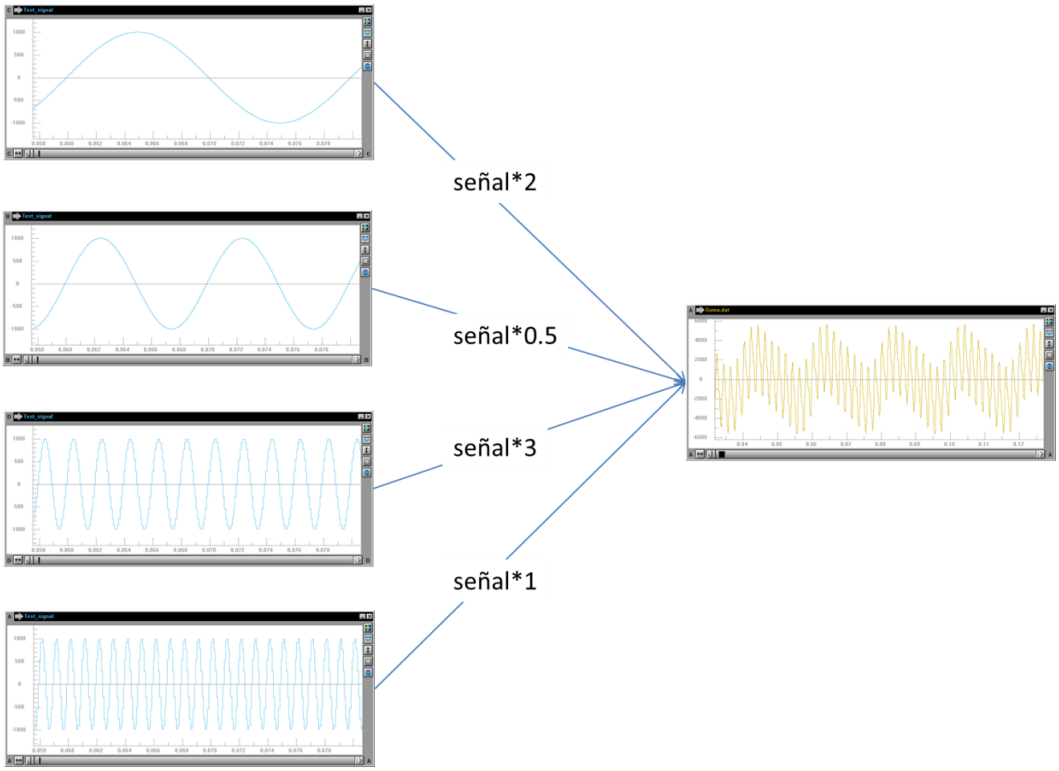


Figura 3.31: Ejemplo de formación de una señal compleja

En la figura 3.31 vemos el concepto inverso a la transformada de Fourier, una señal compleja se construye a partir de la sumatoria de señales simples, en este caso sinusoidales, mismas que son multiplicadas por un factor de "afectación" o un peso y que forman una señal compleja más similar a la voz humana.

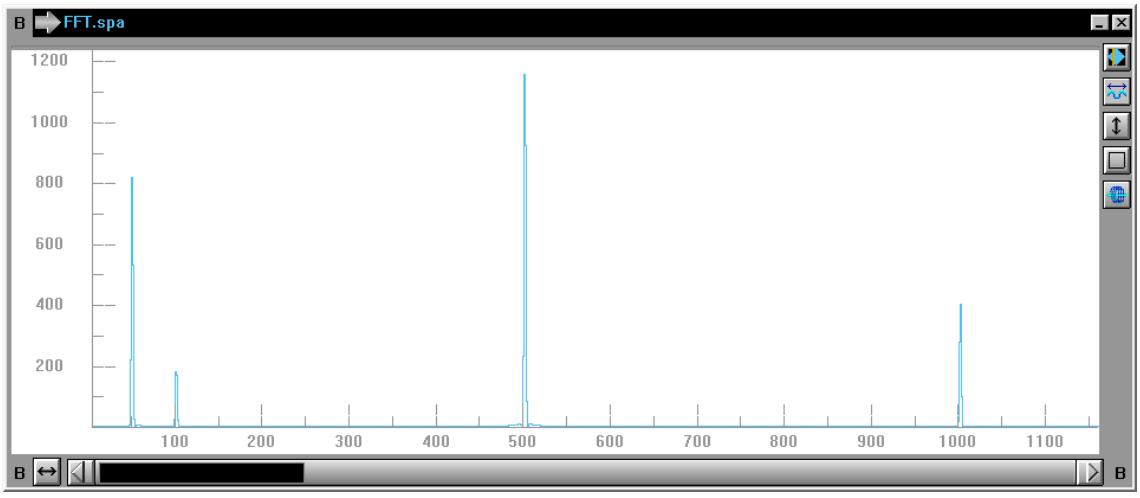


Figura 3.32: FFT de la señal compleja de la figura 2.31

En la figura 3.32, que no es otra cosa que una transformada de Fourier por la técnica de la transformada rápida o FFT (Fast Fourier Transform) por sus siglas en inglés, corroboramos que las frecuencias de las señales simples son 50, 100, 500 y 1,000 [Hz] y que sus pesos son 2, 0.5, 3 y 1, valores que coinciden con aquellos de la figura 3.31, mismo que nos demuestra la importancia y utilidad de esta herramienta de análisis al mostrar claramente información que no se puede discernir de la forma de onda en el lado derecho de la figura 3.31.

Ahora la principal duda sería ¿Qué tamaño de ventana emplear? Para responder esto bastara con explicar el efecto de este tipo de análisis y sus consecuencias. Como ya se mencionó, el análisis por la transformada de Fourier debe realizarse sobre porciones pequeñas de la voz humana, pero al emplear ventanas pequeñas la resolución en frecuencia es pobre, dado que el máximo número de frecuencias a estudiar va de la mano con el tamaño de la ventana misma.

Para explicar esto mejor basta con recordar que la amplitud en frecuencias de la transformada discreta de Fourier o DFT está dada por $^{1}/_{N\tau}$ donde N es el número de puntos de la ventana y τ es el periodo de la señal que es el inverso de la frecuencia, por lo que por ejemplo para una señal muestreada a 11,025 [Hz] y con una ventana digamos de 256 puntos, tendríamos una amplitud en frecuencia o Δf=43.07 [Hz], esto es, sólo analizaremos frecuencias cada Δf, por lo que una frecuencia por ejemplo de 150 [Hz] que no es múltiplo de Δf se apreciara con un cierto esparcimiento (fenómeno de Gibb).

Este parámetro definirá si trabajaremos con espectrogramas de banda ancha o de banda estrecha, cada uno tiene sus ventajas y desventajas, esto se debe a que va ligada de manera inversa la resolución en frecuencia y la resolución en tiempo, a muy buena resolución en frecuencia muy mala resolución en tiempo dado que la ventana es muy grande, a muy buena resolución en tiempo muy mala resolución en frecuencia ya que la ventana es muy pequeña.

Con esto concluido, podemos decir que tenemos el tercer paso, ahora nuestro diagrama de bloques luce así:

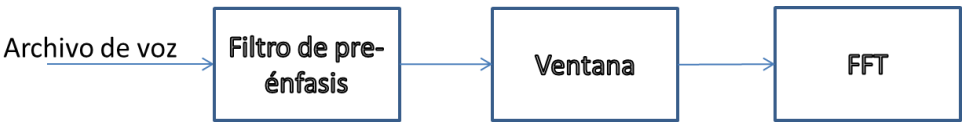


Figura 3.33: Diagrama de bloques con la transformada de Fourier

3.1.4 Envolvente

Como siguiente paso para obtener un modelo veremos que la transformada de Fourier aunque muy útil, produce información que no usaremos, la transformada es simétrica, aunado a que está compuesta de valores complejos, como no nos interesa saber que parte de la señal corresponde a una serie senoidal y cual a una parte cosenoidal, trabajaremos con los valores absolutos o elevado al cuadrado, mismo que nos proporcionará cuánta información está contenida en cada frecuencia. En la figura 3.34 vemos la gráfica de la transformación sin obtener el valor absoluto de los resultados.

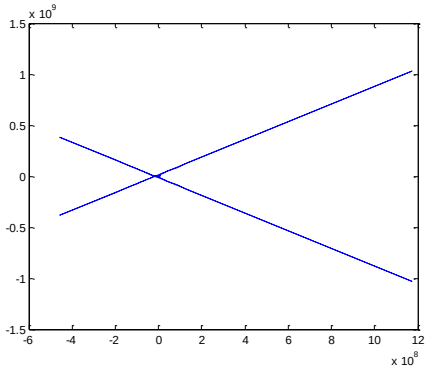


Figura 3.34: Espectro de una señal a una sola frecuencia con valores complejos

Aún después de obtener el espectro de potencia (elevando al cuadrado) que se aprecia en la figura 3.35 seguimos teniendo información no útil, esto porque la simetría antes mencionada, en una transformada con 512 puntos a los 256 puntos se empieza a repetir la misma información formando un espejo completo de los primeros 256 datos, motivo por el cual solo trabajaremos con la mitad de los datos, esto está dado por $^{n}/_{2}$ que es la mitad del tamaño de la ventana de cálculo.

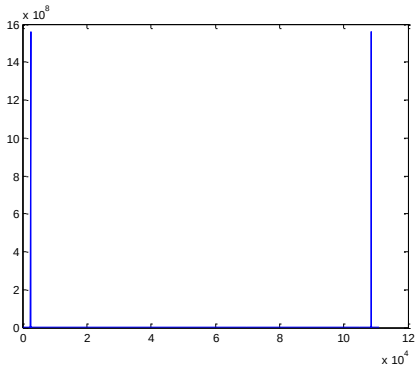


Figura 3.35: Mismo espectro pero con valores absolutos

Aún en este punto el espectro presenta un gran número de fluctuaciones que podrían ser consideradas como información no útil para los fines que perseguimos, resulta más conveniente trabajar con la envolvente de dicha señal (como la de la figura 3.9), que además tiene la gran ventaja de disminuir considerablemente la cantidad de información requerida para su almacenamiento, en la siguiente figura se aprecia un espectro de potencia compuesto de 1,024 datos, mismo que puede ser remplazado por unas pocas decenas de números para representar a su envolvente.

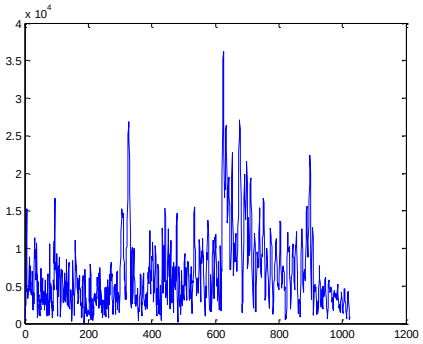


Figura 3.36: Espectro de potencia de una muestra de voz con N=2048

Este proceso de representar una información por medio de su envolvente puede llevarse a cabo a través de un banco de filtros (Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-García, Petrovska-Delacrétaz, Reynolds, 2004), en particular para el habla humana se recomienda además de los filtros, el uso de una escala alternativa conocida como Mel, sin embargo algunos autores sugieren una escala conocida como Bark y otros más la simple escala lineal, para comprender mejor esto explicare de manera breve la escala Mel y la escala Bark.

La escala Mel fue nombrada así por Stevenson, Volkman y Newman en 1937, ésta es una escala perceptual basada en el tono, ésta es equidistante entre un punto y otro desde el punto de vista

perceptual, la referencia de esta escala corresponde a asignar un valor de 1,000 [Mels] a una frecuencia de 1,000 [Hz] 40 [dB] por arriba del límite auditivo, el nombre Mel procede de la palabra inglesa “Melody”.

Realmente no existe una única fórmula para la escala Mel, depende su formulación de aspectos tales como la base logarítmica empleada, así como de las constantes y otros pequeños ajustes, a continuación se muestra una formulación considerablemente popular que se encuentra en el libro de O’Shaugnessy y que es como sigue:

$$m = 2,595 log_{10}\left(1 + \frac{f}{700}\right) = 1,127 log_e\left(1 + \frac{f}{700}\right) \qquad \text{Ecuación 7}$$

La fórmula inversa de la anterior seria:

$$f = 700\left(10^{m/2,595} - 1\right) = 700\left(e^{m/1,127} - 1\right) \qquad \text{Ecuación 8}$$

Emplearemos esta fórmula para convertir nuestra escala de Hertz a una en Mels, para apreciar de manera gráfica como se ve una escala en Mel contra una en Hertz se muestra en la figura 3.37 la gráfica de la fórmula anterior, para archivos con una frecuencia de 11,025 [Hz], que es lo que trabajaremos durante esta tesis, tenemos que por el teorema de Nyquist-Shannon nos da una frecuencia de corte de 5,512 [Hz].

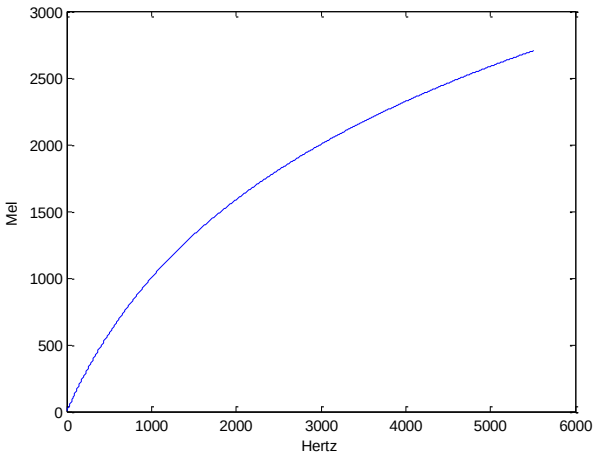


Figura 3.37: Escala Mel-Hertz

Ahora bien, la escala de Bark es una escala psicoacústica propuesta en 1961 por Eberhard Zwicker, fue nombrada así en honor a Heinrich Barkhausen, quien propuso la primera medida subjetiva del volumen. La escala va del 1 al 24 que corresponden a las primeras 24 bandas criticas de audición.

Las bandas de la escala de Bark son fijas, teniendo sus extremos en 20, 100, 200, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000 y 15500 [Hz] respectivamente; sus centros están localizados en 50, 150, 250, 350, 450, 570, 700, 840, 1000, 1170, 1370, 1600, 1850, 2150, 2500, 2900, 3400, 4000, 4800, 5800, 7000, 8500, 10500 y 13500 [Hz].

La formulación para obtener la conversión de Hertz a Barks es la siguiente (propuesta por Traunmüller en 1990) y se muestra su comportamiento en la figura 3.38:

$$Bark = \frac{26.81}{1+\frac{1960}{f}} - 0.53 \qquad \text{Ecuación 9}$$

La inversa de esta formulación es:

$$f = \frac{1960}{\left[\frac{26.81}{z+0.53} - 1\right]} \qquad \text{Ecuación 10}$$

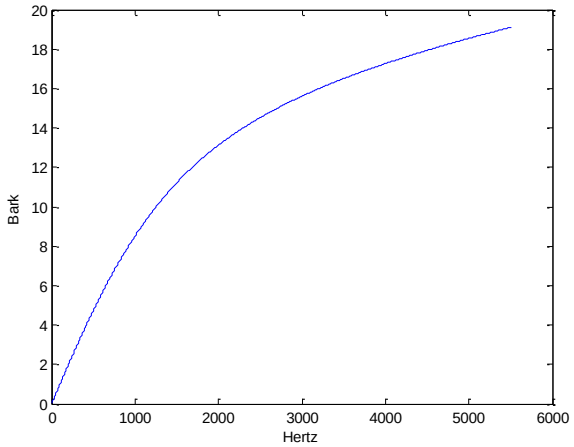


Figura 3.38: Escala Bark-Hertz

En la figura 3.39 se muestra una gráfica comparativa con las tres escalas que se utilizaran, por supuesto para poderlas apreciarlas mejor, éstas se normalizaron para su adecuada comparación, como se puede apreciar la escala Mel y la Bark son similares, pero la escala Mel es más popular.

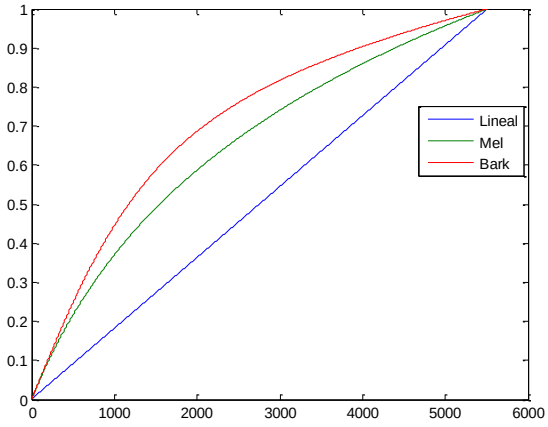


Figura 3.39: Escala Bark, Mel y lineal

En lo que respecta al banco de filtros, cuando se emplea la escala Mel se le conoce como banco de filtros Mel, banco de filtros Bark para dicha escala o simplemente como banco de filtros para la escala lineal, la salida de dicho filtro será un vector con tantos elementos como el orden del filtro empleado, este filtro está compuesto por una serie de filtros individuales, cada filtro se define por la forma del filtro y por su localización (frecuencias izquierda, central y derecha), esta forma puede variar, pero de manera común se emplean filtros triangulares.

En el presente trabajo tendremos para elegir los mismos 17 filtros que se pueden seleccionar para el tipo de ventana y que son aquellos disponibles por default en MATLAB, para definir estos filtros se colocara la frecuencia central en una de las frecuencias críticas y se colocaran los

extremos con un solapamiento variable con el filtro adyacente, teniéndose sólo como casos especiales el primer y último filtro que colocaran su extremo izquierdo en la frecuencia mínima y su extremo derecho en la frecuencia máxima respectivamente.

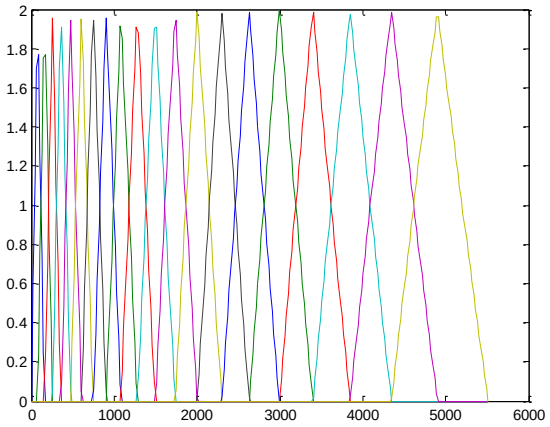


Figura 3.40: Ejemplo de un banco de 20 filtros Mel

Estos extremos de las frecuencias mínimas y máximas pueden ser empleados para remover o filtrar influencias en bandas en las que no esperamos contenido de habla pero que si puede existir otro tipo de información ajena al locutor, por ejemplo, en bandas de frecuencia tales como 50-60 [Hz] podemos encontrar la componente de alterna del sistema eléctrico, pero no información relevante al locutor, motivo por el cual puede ser muy recomendable su filtrado.

Para la selección de las frecuencias y al no tenerse pre-establecido cuales son críticas ya que la naturaleza de la señal vocálica varia de manera constante, serán colocados de manera equidistante en la escala correspondiente, definiéndose el número de filtros por el número K, el mínimo recomendado de filtros es de 12, motivo por el cual se limitará el sistema a no aceptar menos de 8 y no más de 50 para este parámetro, ya que como vemos, el número de combinaciones y permutaciones de las posibles configuraciones va creciendo rápidamente.

De manera adicional no usaremos los filtros pre-establecidos para Bark, ya que sólo usaríamos los primeros 10 y los restantes no serían aprovechados por la limitante del canal que se tiene prevista en el presente trabajo, motivo por el cual se usará el mismo método de K filtros equidistantes planteados para Mel y para la escala lineal, pero estarán equidistantes en escala Bark.

Como último detalle que concierne al filtrado, resta explicar su aplicación, tendremos K filtros independientes por los que tendría que ser multiplicada la señal, recordemos que una propiedad de la transformada de Fourier es que la convolución de dos señales en el dominio del tiempo, que sería como coloquialmente se aplicaría un filtro, es sustituida por la multiplicación de la señal por el filtro, así como su concatenación de resultados para obtener cada uno de los K coeficientes de acuerdo a la siguiente formula.

$$S(l) = \sum_{n=0}^{N/2} S(n) M_l(n) \text{ para l=0, 1,...,L-1} \qquad \text{Ecuación 11}$$

S(n) – Es el espectro de potencia
N – Es el tamaño de la ventana
L – Es el número de filtros K

La función para crear el banco de filtros se encuentra en el anexo 1, esta función puede producir filtros en escala Mel, Bark y lineales, así como emplear formas de filtro Bartlett-Hanning, Bartlett, Blackman, Blackman-Harris, Bohman, Chebyshev, parte alta plana, Gaussiana, Hamming, Hann, Kaiser, Nuttall (Blackman-Harris de N puntos), Parzen, rectangular, Taylor, triangular y Tukey, pudiéndose también establecer los límites mínimos y máximos de corte y el porcentaje de traslape de los filtros, así como el orden K del banco. En la figura 3.41 se muestran algunos ejemplos del banco en mención y en la 3.42 se muestra el resultado del filtrado.

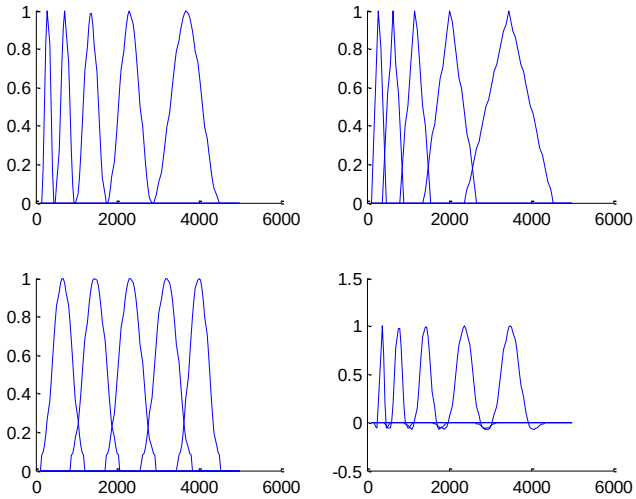


Figura 3.41: Ejemplos de filtros con distintas escalas, formas y solapamientos con K=5

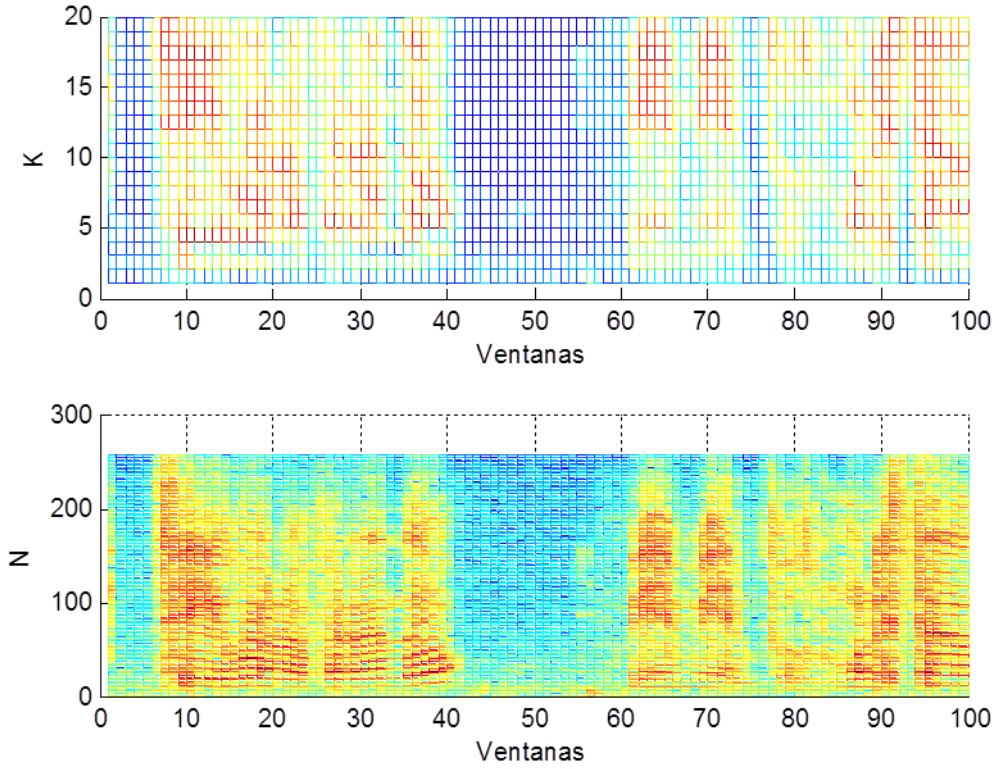


Figura 3.42: Ejemplo de la envolvente en escala Mel del espectro de potencia que se muestra en la parte inferior

Con esto podemos decir que tenemos el cuarto paso, ahora nuestro diagrama de bloques luce así:

Figura 3.43: Diagrama de bloques con el banco de filtros

3.1.5 Coeficientes

Como siguiente paso simplemente corresponde pasar a dB el resultado del filtro anterior, por su definición misma, que sería obtener el logaritmo del filtro y multiplicarlo por 20, pero como nuestro interés son los coeficientes cepstrales, es común aplicar una transformada coseno discreto (DCT por sus siglas en inglés (Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-García, Petrovska-Delacrétaz, Reynolds, 2004)) al resultado anterior, ya que ésta es comúnmente utilizada en la compresión de datos, esto dado que la transformada de Fourier (DFT por sus siglas en inglés) y su implícita periodicidad resulta en que las discontinuidades ocurren en las fronteras, siendo que en contraste en la DCT donde ambas fronteras son iguales esto no sucede, produciendo una extensión continua en las fronteras, aunado a la reducción del costo computacional o su mayor simpleza de cálculo (Wikipedia, DCT).

A pesar de la recomendación ofreceremos ambas opciones, transformada discreta de Fourier o transformada coseno discreta, ya que después de todo, la DCT es simplemente un atajo computacional de la DFT según otros autores (math.stackexchange.com), motivo por el cual se permitirá comprobar si esta tiene algún efecto en la calidad del resultado.

Como resultado de este último paso obtenemos los coeficientes cepstrales y podemos decir que tenemos nuestro diagrama de bloques está completo.

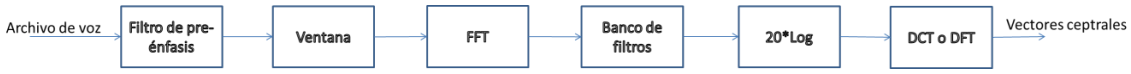


Figura 3.44: Diagrama de bloques completo

3.2 Procesamiento de los coeficientes cepstrales

Los coeficientes obtenidos son la materia prima con la que trabajaremos, pero de nuevo estos pueden ser acondicionados y tratados en miras de mejorar los resultados obtenidos, por este motivo veremos algunos puntos adicionales que pueden ser de utilidad.

3.2.1 Centrado y reducción de los vectores

El centrado de los vectores se refiere simplemente restar la media de todos los valores obtenidos a cada uno de estos, este método se conoce como la resta cepstral de la media o CMS por sus siglas en ingles.

La reducción de estos valores se refiere a la normalización que se puede realizar dato por dato, estos dos procesos son comúnmente empleados en la verificación de locutores, motivo por el cual pueden resultar útiles en la identificación de locutores.

3.2.2 Información dinámica y energía

Una vez que se tienen los coeficientes cepstrales, opcionalmente centrados y reducidos, se pueden incorporar al vector que caracterizará al locutor otra información, puede resultar útil incluir información sobre la energía de la señal, ya que los coeficientes cepstrales no captan esta información, así mismo, sabemos que la voz humana no es constante, motivo por el cual puede ser sumamente conveniente añadir información sobre cómo cambia la envolvente de la señal o lo que es lo mismo, información dinámica (Dan Jurafsky, 2009), que representa como estos vectores varían en el tiempo.

De manera común muchos autores (Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-García, Petrovska-Delacrétaz, Reynolds, 2004), realizan esto mediante los parámetros delta y delta-delta que se calculan mediante una aproximación polinomial de la primera y segunda derivada, siendo su cálculo realizado mediante la siguiente formula:

$$d_i = \frac{\sum_{n=1}^{N} n(c_{i+n} - c_{i-n})}{2\sum_{n=1}^{N} n^2} \qquad \text{Ecuación 12}$$

Esta formulación no es la más simple, ya que la primera aproximación es simplemente la siguiente:

$$d_i = \frac{c_{i+1} - c_{i-1}}{2} \qquad \text{Ecuación 13}$$

Sin embargo, aquí se usará la aproximación conocida como deltas desplazados o SDC (Shifted Delta Coefficients) que es una técnica que suma bloques de N vectores para el cálculo de cada delta, donde la opción más simple es la segunda fórmula con N=1, pero que puede aplicarse con un N diferente en miras de mejorar la apreciación de la información dinámica que nos permite una aproximación simple a la extracción de factores acústicos prosódicos (Jonathan J. Lareau, 2006).

Este método es una aproximación al estudio pseudo-prosódico del habla sin necesariamente tener que encontrar un modelo o cálculo de los parámetros prosódicos, ya que nos permite tomar en consideración aspectos de la entonación de la frase, la coarticulación y acentuación al observar aspectos supra segméntales cuanto mayor sea N, siendo estos los aspectos clave que estudia la prosodia desde su punto de vista físico.

Para la energía, ésta se obtiene de la señal original una vez aplicada la ventana correspondiente por medio de la fórmula siguiente:

$$E = log \sum_{n=1}^{N} s_n^2 \qquad \text{Ecuación 14}$$

Se recomienda en general aumentar al vector que contiene la información relativa a los coeficientes cepstrales la información de la energía, la información correspondiente a los delta coeficientes cepstrales, el delta de la energía y de nueva cuenta los coeficientes delta-delta de aceleración que no son otra cosa que el empleo de la misma aproximación de manera recursiva sobre los coeficientes delta, empleándose incluso la misma aproximación polinomial para el delta de la energía.

MFCC	E	ΔMFCC	ΔE	ΔΔMFCC	ΔΔE

Figura 3.45: Vector de características

Con esta adición nuestro diagrama de bloques que nos entregaba coeficientes MFCC se modifica ligeramente, mismo que queda como a continuación se indica:

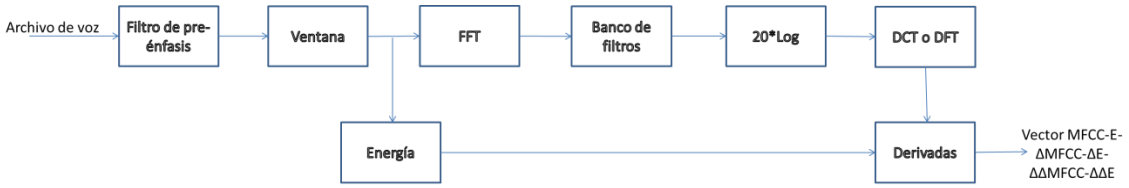


Figura 3.46: Diagrama de bloques de obtención del vector de características

Como una última consideración en la formación del vector que representara al locutor, vale la pena mencionar que el coeficiente cero (primer coeficiente) corresponde a la fuente de excitación, que en el caso de habla es el soplo fonatorio que excita los pliegues vocálicos y que comúnmente se conoce como F_0 o tono fundamental, mismo que tiene valores identificativos, pero que en general está sumamente ligado a los estados emocionales e incluso a aspectos prosódicos del habla, factor que lo hace sumamente variante, motivo por el cual se omite de manera generalizada en el vector final que se estudiara, aislando así la fuente de excitación del filtro.

3.3 Detección de actividad vocálica

Como un elemento de pre-procesamiento adicional (Ville Hautamäki, Marko Tuononen, Tuija Niemi-Laitinen and Pasi Fränti, 2007) se recomienda implementar un algoritmo que permita separar al habla del ruido o porciones de silencio, ya que estos no auxilian en nada a la identificación del locutor y pueden llegar a incrementar el error en los sistemas de identificación de locutores al identificar el ruido ambiental y no al locutor, aunado a la pérdida de tiempo empleado en la extracción de características de las porciones de audio donde no hay voz.

Ésta es una técnica utilizada en el procesamiento del habla en general, por ejemplo para la codificación del habla, para el reconocimiento del habla o para activar o desactivar un proceso, motivo por el cual es ampliamente utilizado en la telefonía celular, por este motivo existen múltiples aproximaciones a la solución de este problema.

Algunas de las variantes pueden incluir aquellos sistemas VAD (por sus siglas en inglés Voice Activity Detection) que trabajan en tiempo real, por ejemplo para la telefonía celular, y aquellos algoritmos donde la calidad es más importante que la velocidad, como lo es la identificación de locutores.

En nuestro caso en particular emplearemos un método combinado de detección por energía y por el comportamiento en frecuencia de la señal, aunado a que será aplicado en periodos cortos de tiempo como ya se ha venido manejando, mas sin embargo no es por el mismo motivo, aquí lo que nos interesa es no dejar pasar la señal que no es voz y si dejar pasar toda señal que sea voz, motivo por el cual la ventana debe ser lo suficientemente pequeña.

Para la energía la aproximación es muy simple, se toma una ventana rectangular de la señal a estudiar sin traslapes en esta ventana y se obtiene la energía de este segmento de la señal mediante la aplicación de la fórmula:

$$E(n) = \frac{1}{N}\sum_{k=0}^{N-1} x^2(k) \qquad \text{Ecuación 15}$$

Esto nos arrojara un simple número que representa la energía promedio de la señal en ese periodo, como se puede ver en la figura 3.47 la energía en los periodos con voz es mucho más alta que en aquellas ventanas donde no hay voz como se muestra en la figura 3.48, si presentáramos ambas gráficas en una misma ventana, la gráfica azul de la porción sin habla no sería visible por la diferencia en escala, en la figura 3.49 se puede apreciar esta gráfica comparativa con ambas señales pero en escala logarítmica.

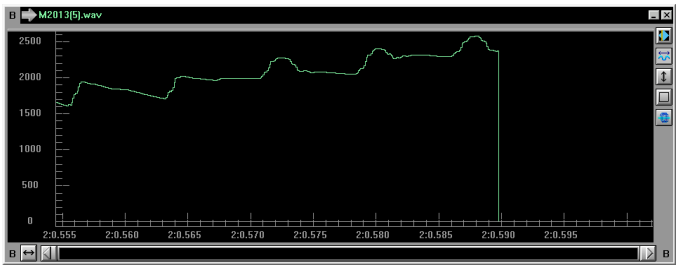


Figura 3.47: Energía en una ventana con voz

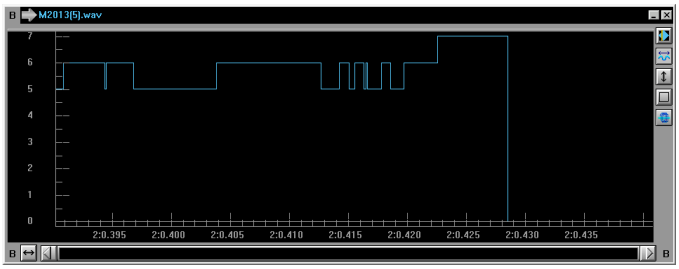


Figura 3.48: Energía en una ventana sin voz

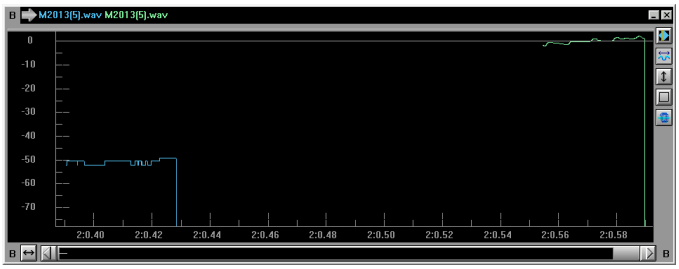


Figura 3.49: Energía de señal con y sin voz en escala logarítmica

Para lo que respecta al análisis en frecuencia, esto requiere de una transformada rápida de Fourier que obedece a los mismos razonamientos ya expuestos, el cambio de la señal al dominio de la frecuencia nos permite estudiar la descomposición del audio en sus componentes de frecuencia, una vez hecho esto bastara con obtener el máximo de este análisis de Fourier, esto es, la frecuencia que más contribuye en el segmento.

La obtención de esta frecuencia nos es útil ya que en teoría el ruido tiene una distribución energética similar en todas las frecuencias como se puede apreciar en la figura 3.50, caso

contrario cuando existe voz, ésta se comporta como lo vimos en la parte del pre-énfasis y como se aprecia en la figura 3.51, ambas gráficas se pueden apreciar juntas en la figura 3.52 que muestra, que tanto la distribución energética como la separación entre los máximos y los mínimos locales es mucho más importante.

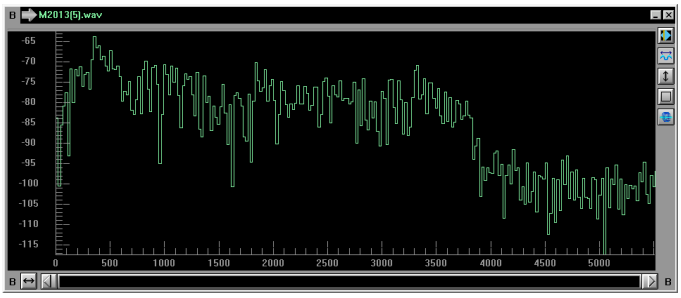


Figura 3.50: Espectro de una porción de señal con ruido

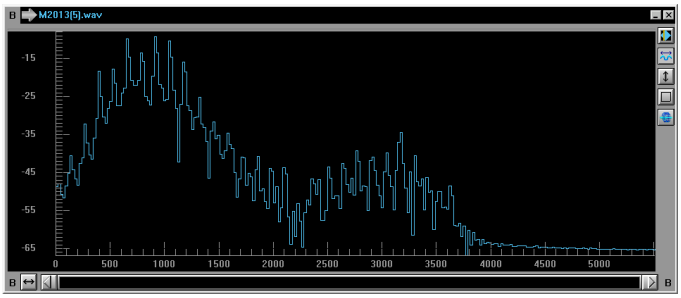


Figura 3.51: Espectro de una porción de señal con voz

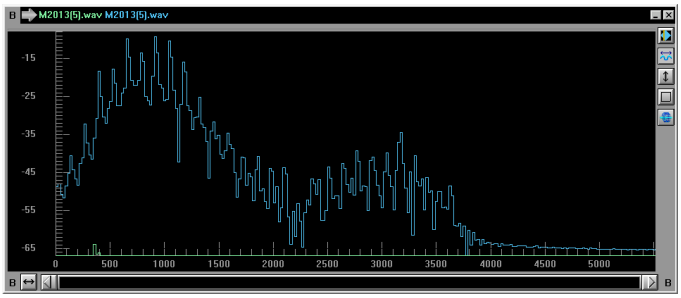


Figura 3.52: Comparativa del espectro de una voz contra la del ruido

Basándonos en lo presentado y con un algoritmo de doble criterio, que significa que si cualquiera de los dos métodos lo da como voz este segmento es tomado en consideración, se realiza la detección de actividad vocal marcando que porciones del audio tienen voz y cuáles no, sin embargo queda una pregunta: ¿Qué umbral tomar?

Podría pensarse que el umbral es trivial, sin embargo no lo es, recordemos que nosotros como seres humanos tomamos decisiones basados en información, por ejemplo en la figura 3.53 vemos una porción de audio y un recuadro verde que separa la voz de los silencios, aunado a dos gráficas más, una representando la energía de cada ventana y otra más con las frecuencias máximas de la misma ventana, por lo que se ve que los picos son el habla y lo plano es el silencio, pero en la realidad ninguna señal es plana, sólo son de escala distinta.

Existen múltiples aproximaciones para la selección de este disparador o indicador de que considerar voz y que es silencio, pero en este caso y al no ser éste el objetivo primario de la tesis se seleccionó una muy simple, al umbral de diferencia entre el mínimo y el máximo de

cada señal es dividido en 100 porciones y el usuario puede indicar cuantas de estas 100 divisiones deben ser consideradas como silencio, esto es una selección manual de la sensibilidad del sistema.

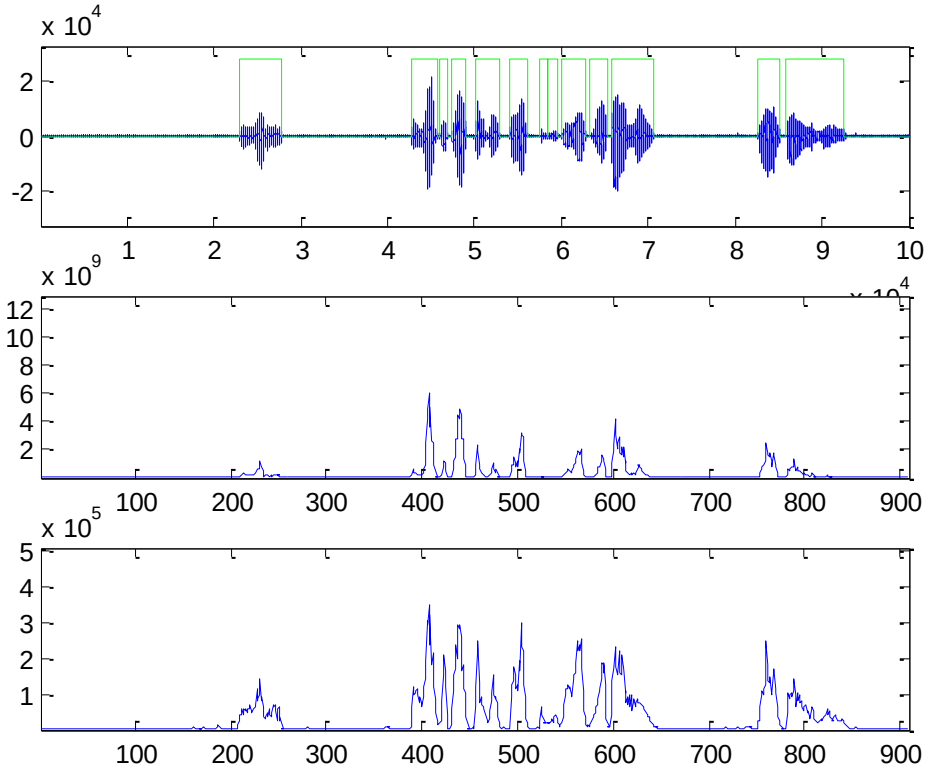


Figura 3.53: Grabación marcada en voz y no voz con grafica de energía y frecuencias

El código que realiza esta detección de actividad vocal se encuentra en el anexo 1, este código aunado a la función del VAD, permite la obtención de la relación señal a ruido o SNR que es simplemente la división de los niveles promedio de la señal que se consideró como voz sobre el promedio de los silencios, así como el cálculo de cuánto tiempo de voz pura (sin silencios o ruidos) se conservará, por último nos ofrece la opción de graficar el resultado de este proceso que es como se obtiene la gráfica de la figura 3.53 mediante la función grafVAD.m que se encuentra también en el anexo 1.

Para este módulo en concreto el diagrama de bloques quedaría como sigue:

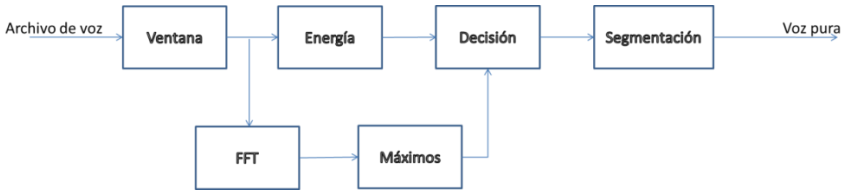


Figura 3.54: Diagrama de bloques de módulo VAD

Con este nuevo módulo nuestro diagrama de bloques general quedaría como sigue:

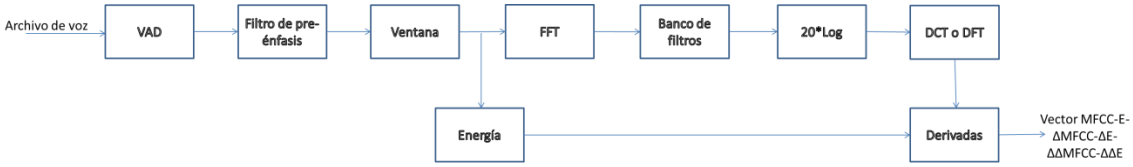


Figura 3.55: Diagrama de bloques de obtención del vector de características

3.4 Reconocimiento de patrones

El modelado es una técnica cognitiva utilizada en el reconocimiento de patrones que consiste en crear una representación ideal de un objeto real mediante un conjunto de simplificaciones y abstracciones. Su validez se puede constatar llevando a cabo comparaciones de las implicaciones predichas por el modelo contra las del objeto real.

En otras palabras, se trata de usar un modelo irreal o ideal, mismo que es similar a crear una escultura o representación del objeto real

Un modelo es una simplificación de la realidad, se recogen aquellos aspectos de gran importancia y se omiten los que no tienen relevancia para el nivel de abstracción deseado. Se modela para comprender mejor un sistema. Los sistemas complejos no se pueden comprender en toda su completitud (según el enfoque del algoritmo Dijkstra-Scholten "divide y vencerás").

Los principios de modelado son:

- Primero: la elección de los modelos tiene una profunda influencia en el acometimiento del problema y en cómo se da forma a la solución.
- Segundo: los modelos se pueden representar en distintos niveles de detalle, los analistas se suelen centrar en el ‘qué’, mientras que los diseñadores en el ‘cómo’.
- Tercero: los mejores modelos se mantienen ligados a la realidad.
- Cuarto: un único modelo no es suficiente. Cualquier sistema no trivial se aborda mejor mediante un pequeño conjunto de modelos casi independientes, es decir, que se puedan construir y estudiar por separado pero que estén interrelacionados.

De esto vemos que un modelo es una simplificación que nos es útil para los fines que nos interesan, en este caso identificar a la persona que habla, ya pasamos por un procesamiento del habla hasta obtener coeficientes que “representan” al locutor, pero sin embargo, aquí también se sigue llevando mucha información concerniente al mensaje y otros temas.

Aunado a lo anterior, tenemos vectores de gran tamaño y de diferente tamaño para cada grabación que procesamos, mismo que tampoco resulta conveniente, por esto se requiere representar a nuestro locutor con un modelo de tamaño fijo y significativamente más fácil de comparar con otro modelo.

La meta final del reconocimiento de patrones es la clasificación de datos en clases, en nuestro caso, una clase sería un locutor o algún elemento interesante perteneciente a un locutor y que lo caracteriza, de tal suerte que se pueda tratar de empatar las características de un locutor con otro distinto o en su defecto contra otra muestra de voz del mismo locutor.

Para esta tarea existen múltiples aproximaciones, como DTW o ajuste de tiempo dinámico, las cadenas ocultas de Markov, los modelos de mezclas de gaussianas, sin embargo ya tenemos una gran cantidad de "parámetros" que podemos manipular y que no existe literatura alguna que garantice o muestre cual es la configuración o juego de parámetros ideal, en este entendido, el alcance del presente trabajo pretende centrarse en la ubicación de esa configuración más que en detalles específicos de como robustecer un algoritmo de detección de actividad vocal o de aceleración de tareas para aplicaciones que requieren análisis en tiempo real.

Con esto en mente se seleccionó un algoritmo simple pero a la vez poderoso para la creación de un modelo, la cuantización vectorial, ésta es un proceso de mapeado de vectores de un espacio de gran tamaño a un número finito y mucho menor, cada región es conocida como "cluster" o grupo y se representa por una palabra código (codeword) y a la colección de palabras claves se le conoce como el libro de código (codebook).

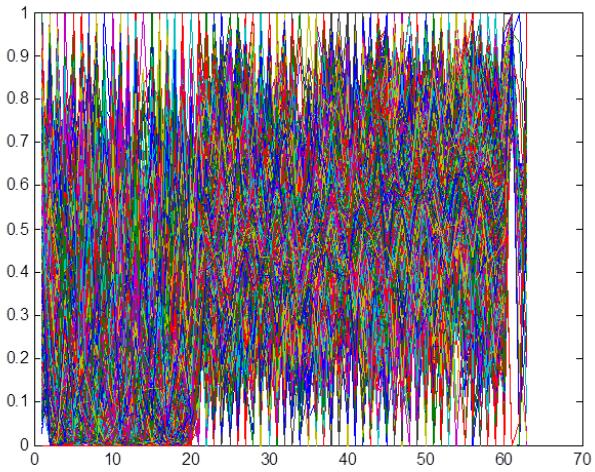


Figura 3.56: Vector de coeficientes de un locutor

En la figura 3.56 podemos apreciar un vector completo de coeficientes MFCC de un locutor cualquiera, como se puede apreciar, el número de datos es abrumador, podemos tener literalmente cientos de miles de vectores que serían difíciles de analizar.

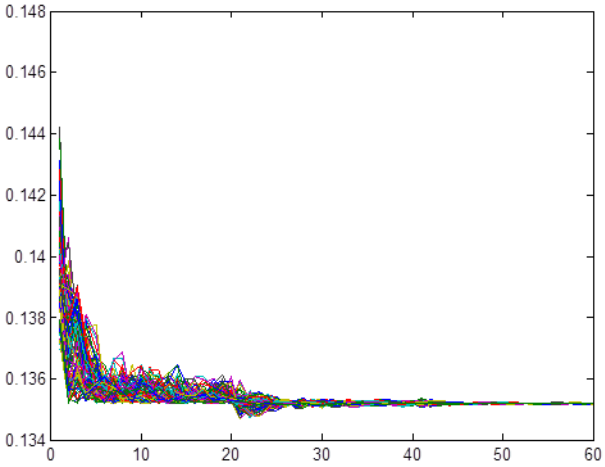


Figura 3.57: Vector de coeficientes de un locutor con VQ

Ahora, como podemos ver en la figura 3.57, esto se puede simplificar de manera significativa al eliminar la información "redundante", ya que mucha información puede ser representada por una palabra código, esto es similar al concepto del centroide como lo podemos ver en la figura 3.58, un sólo dato puede representar a varios datos mediante su media o centro de masa.

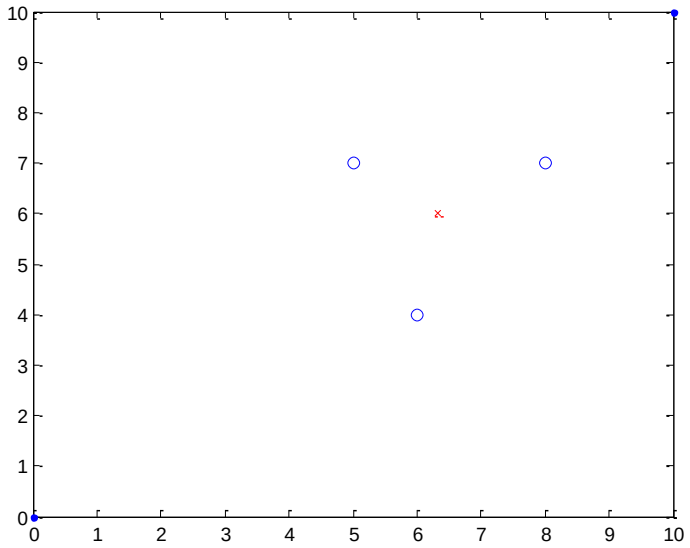


Figura 3.58: Centroide de tres datos

De similar manera pero a nivel vectorial, un sólo vector puede representar a varios vectores que están "cerca" de él, la técnica de encontrar estos vectores que representan a los datos es la generación de agrupaciones o clusters a la distancia entre un libro código y otro se le conoce como distorsión VQ.

Para el cálculo de esta cuantización vectorial basta con seguir los 6 pasos del algoritmo LBG por las iniciales de sus creadores (Linde, Buzo y Gray en 1980), éste permite agrupar un juego de L vectores a un juego de M palabras código, su implementación es la siguiente:

1- Crear la primera palabra código que es el centroide de todos los L vectores de datos
2- Dividir cada palabra código de acuerdo a la regla siguiente (ε es un factor de división, se puede experimentar con distintos valores y observar los resultados)

$$y_n^+ = y_n(1+\varepsilon)$$
$$y_n^- = y_n(1-\varepsilon)$$

3- Localizar a los vecinos más cercanos, cada vector de L deberá de ser asignado a alguna de las palabras código de acuerdo a la distancia que existe entre el punto y la palabra
4- Actualizar los centroides de cada palabra de acuerdo los vectores de L asignados a dicha célula
5- Primera iteración: repetir los pasos 3 y 4 hasta que la distancia promedio sea menor a un disparador que nosotros definimos, a menor disparador más largo será el ciclo hasta llegar a que sea infinito (ciclar la iteración)
6- Segunda iteración: repetir los pasos 2, 3 y 4 hasta que se logre un libro código de M palabras código.

La implementación de este modelado por medio del algoritmo de cuantización vectorial se encuentra disponible en el anexo 1, con este nuevo módulo nuestro diagrama de bloques general quedaría como sigue:

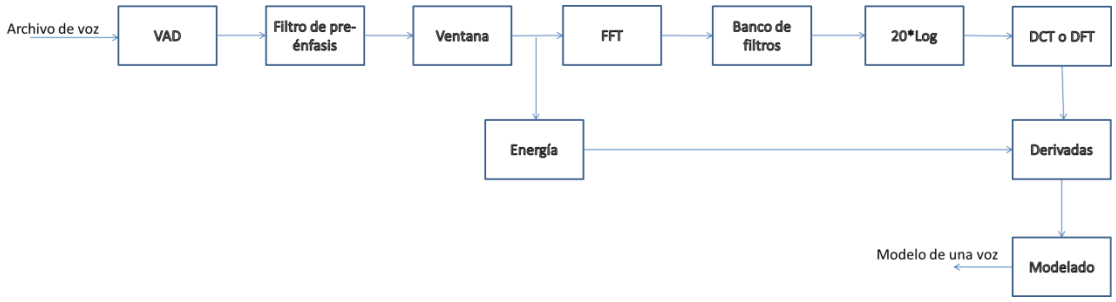


Figura 3.59: Diagrama de bloques de obtención del modelo

3.5 Distancia

Ahora que tenemos un modelo de nuestro locutor ¿Qué hacemos con él?

Nuestra meta es identificar personas por medio de su voz, en este sentido el problema es ¿cómo lograrlo?, sólo tenemos un "modelo" de nuestro locutor que son una serie de números en una forma totalmente distinta a la serie de números que teníamos originalmente, después de esta extracción de características y su agrupamiento en palabras código que conforman un libro código tenemos que "medir" que tan similar es un modelo a otro.

La distancia es una descripción numérica de que tan distantes están dos objetos, es fácil comprender la parte física si tenemos una sola dimensión como se muestra en la figura 3.60, la distancia es la diferencia en la única dimensión entre el punto A y el B, así que si el punto A esta en el punto 2 y el B en el 4 la distancia sería "2".

Figura 3.60: Distancia en una dimensión

En dos dimensiones el problema sigue siendo simple, aunque su solución quizás no sea tan aparente, en la figura 3.61 se muestra este concepto, la distancia está dada por la línea verde que une ambos puntos, y de la cual se puede obtener su largo mediante un triángulo rectángulo que se muestra en rojo, el teorema de Pitágoras nos dice que el cuadrado de la hipotenusa (línea verde) es igual a la suma de los cuadrados de los catetos, y cada cateto se define como la resta de ambos puntos, por esto la distancia seria:

$$d = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2} \qquad \text{Ecuación 16}$$

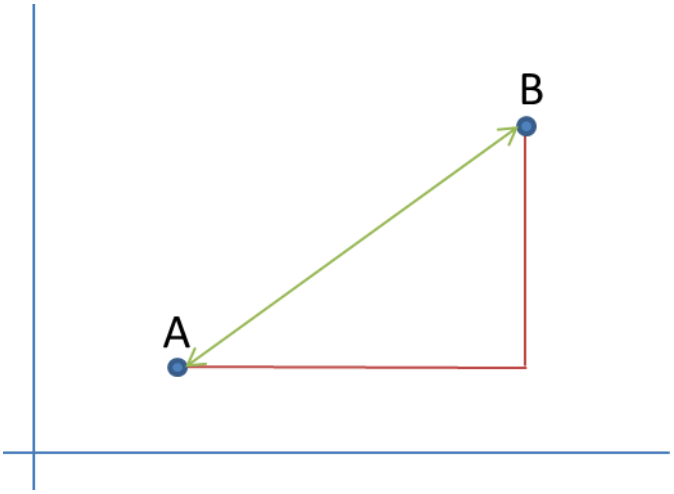


Figura 3.61: Distancia en dos dimensiones

Para el caso de tres dimensiones resulta menos aparente que la distancia seguiría siendo la misma que se definió antes, este concepto es conocido en general como distancia Euclidiana, por lo que se debe de cumplir que los puntos estén en un espacio Euclidiano, pero en general se calcularía de la misma manera para el caso mostrado en la figura 3.62 que son puntos en el espacio (tres dimensiones).

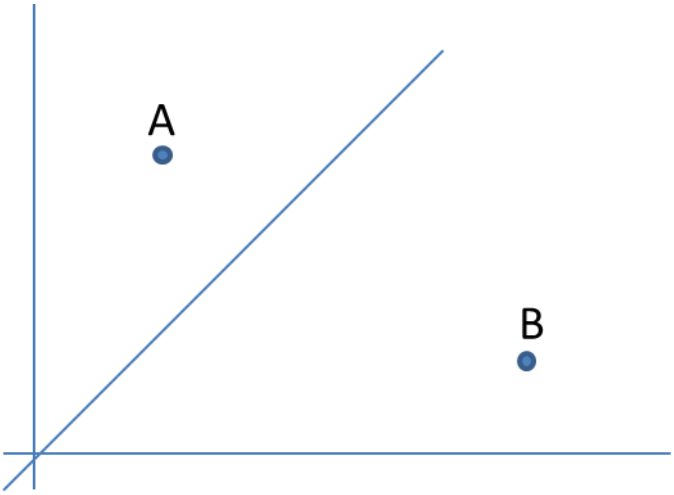


Figura 3.62: Distancia en tres dimensiones

$$d = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2} \qquad \text{Ecuación 17}$$

Cuando pasamos de tres dimensiones ya no existe analogía gráfica, entonces se dice que ya no es físico sino una generalización matemática que nos continúa brindando esa descripción numérica de que tan lejano está un objeto de otro aún cuando hablemos de 'n' dimensiones, para este caso la formulación es la siguiente.

$$d = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \qquad \text{Ecuación 18}$$

Esta fórmula nos permitiría calcular la distancia entre dos locutores o re-fraseando una comparativa de que tan lejano está un locutor de otro, que es realmente nuestro objetivo, esta

fórmula nos permite obtener una valoración de que tan cercano esta un locutor de otro, quedando ésta función en el anexo 1.

3.5.1 Qué significa la "Distancia"

Al seguir avanzando nos encontramos con un nuevo problema, ¿Qué significa la distancia que el sistema obtiene? Mismo que no es fácil responder, como ya explicamos antes, el medir la distancia de un locutor a otro da sólo un número, pero como saber que número es bueno y que número es malo o cuando el sistema logró identificar al locutor y cuando falló, para esto y como ya se explicó también en el primer capítulo, todos los sistemas biométricos tienen que ser probados o calibrados para "averiguar" cuando se trata de una identificación positiva y cuando de una negativa, y más aún de un error en la identificación.

Para esto recordemos que un sistema requiere tener un disparador o margen donde comenzará a tomar la identificación como positiva y en el sentido contrario como negativa, pero ¿Cómo averiguar esto en nuestro sistema?

La respuesta es menos complicada de lo que parece, simplemente requerimos un grupo de datos de los que conocemos a priori cuales grabaciones corresponden a un mismo locutor y cuales a uno distinto, el principal problema suele ser el grupo de datos, como no es tema de este trabajo utilizaremos simplemente una parte de un corpus ya existente y de reconocido prestigio, el corpus de las pruebas SRE (Speaker Recognition Evaluation) del NIST en su versión 2008 que incluyen varias grabaciones de cada locutor hablando en diversos idiomas y con un texto totalmente libre en un ambiente de grabación relativamente libre de ruido.

Para las pruebas en comento emplearé un total de 500 grabaciones de hombres y 500 de mujeres, teniéndose en total 1,000 grabaciones de 174 locutores distintos (89 mujeres y 85 hombres), cada grabación se compara contra las 1000 grabaciones, obteniéndose un total de 1,000,000 de comparativos biométricos por prueba donde la diagonal de la matriz corresponde a comparar un archivo consigo mismo (distancia cero) y además ambos lados de la tabla son simétricos como se aprecia en la siguiente tabla:

	F2008(1)	F2008(2)	F2008(3)	F2008(4)	F2008(5)	F2080(1)	F2080(2)
F2008(1)	0	219.7306	228.2872	211.0657	207.5297	286.0386	287.0377
F2008(2)	219.7306	0	248.8123	238.9553	232.9679	304.7251	295.6031
F2008(3)	228.2872	248.8123	0	242.156	266.9917	279.7987	275.8222
F2008(4)	211.0657	238.9553	242.156	0	230.6404	290.4721	282.4967
F2008(5)	207.5297	232.9679	266.9917	230.6404	0	299.6183	309.9867
F2080(1)	286.0386	304.7251	279.7987	290.4721	299.6183	0	248.3132
F2080(2)	287.0377	295.6031	275.8222	282.4967	309.9867	248.3132	0

Tabla 13: Resultados de un comparativo

Como se vió anteriormente, el cálculo de la distancia es simple contra el costo computacional del modelado, por este motivo es más simple comparar todos los archivos contra todos los archivos a crear una función que compara sólo aquellos archivos que nos interesan, después de todo no nos afecta en nada estos cálculos extras y nos ofrecen una mejor visualización de las tablas de resultados.

Ahora bien, ya tenemos la tabla de resultados, pero ¿qué hacer con ésta?, de entrada vemos que los resultados de la distancia son números reales, en este sentido se propone agrupar estas distancias en paquetes, en este caso en concreto se obtiene el máximo y el mínimo de todas las distancias obtenidas y se divide en 100 paquetes (el número 100 fue arbitrariamente seleccionado pero debe ser menor al número de audios a probar), lo que es claro es que cuantas más divisiones, más fino podría ser el análisis de resultados, pero por otra parte se tiene el hecho de que cuanto más grande el número de divisiones, mayor tiene que ser el número de comparaciones a trabajarse, ya que si tenemos sólo cuatro comparaciones y tenemos 100 divisiones no se podrían realizar algunos de los cálculos que veremos más adelante.

Ya que tenemos nuestros 100 rangos de distancias tenemos que clasificar cada resultado (descartando por supuesto las comparaciones de un archivo consigo mismo, ya que esto no representa nada real) dentro de los rangos calculados, para ello requerimos comparaciones donde se sabe que se trata de la misma persona y grabaciones donde se sabe que se trata de personas distintas, con esto podríamos armar una tabla similar a la siguiente:

	100-110	110.1-120	120.1-130	130.1-140	140.1-150	150.1-160
Genuino	0	0	1	1	2	2
Impostor	2	2	1	0	0	0

Tabla 14: Tabla simplificada de resultados

Esta tabla nos dice cuántos casos tenemos de comparaciones que caen dentro de una categoría o rango determinado, en nuestro caso son 100 rangos y no 6 y el mínimo y el máximo dependerán del grupo de audios del corpus.

Con estos datos, el siguiente paso es dividir cada dato entre el número total de pruebas de cada tipo, en nuestro ejemplo esto es entre 6 para las comparaciones entre la misma persona y 5 en el caso de locutores distintos, esto convertirá nuestro número de resultados en tasas de desempeño (Ted Dunstone, Neil Yager, 2009), con esto nuestra tabla quedaría como sigue:

	100-110	110.1-120	120.1-130	130.1-140	140.1-150	150.1-160
Genuino	0	0	1/6	1/6	2/6	2/6
Impostor	2/5	2/5	1/5	0	0	0

Tabla 15: Tabla de tasas de desempeño

Ahora ya tenemos toda la información que requerimos, sólo necesitamos verla de una manera ligeramente distinta para poder evaluar lo que está pasando, para esto basta con agrupar los resultados para cada rango de distancias de acuerdo a las siguientes reglas:

$$FRR: \frac{número\ de\ genuinas < rango(disparador)}{total}$$

$$FAR: \frac{número\ de\ impostores \geq rango(disparador)}{total}$$

Recordemos que FRR por sus siglas en inglés (False Rejection Rate) es la tasa de falsos rechazos y FAR (False Acceptance Rate) es la tasa de falsas aceptaciones, por lo que una vez calculado cada valor la anterior tabla de resultados quedaría ahora como sigue:

Rango	FRR	FAR
110 o menor	0%	5/5=100%
110.1-120	0%	3/5=60%
120.1-130	0%	1/5=20%
130.1-140	1/6=16%	0%
140.1-150	2/6=33%	0%
150.1-160	4/6=66%	0%
160.1 o mayor	6/6=100%	0%

Tabla 16: Tabla de FRR y FAR

Con esta tabla ya podemos construir las gráficas de desempeño de nuestro sistema bajo estándares internacionales en lo que a biometría se refiere, esto mediante gráficas ROC (Receiver Operating Characteristic) o característica operacional de receptor y en gráficas FRR vs. FAR, en la primera se grafica el FAR contra la tasa de verificación (1-FRR) y en la otra se grafica FRR contra FAR, en las abscisas se apreciarían los rangos o disparadores y en las ordenadas el porcentaje, como un ejemplo se pueden presentar las siguientes gráficas obtenidas con el sistema en una prueba de 1,000 locutores (un millón de identificaciones).

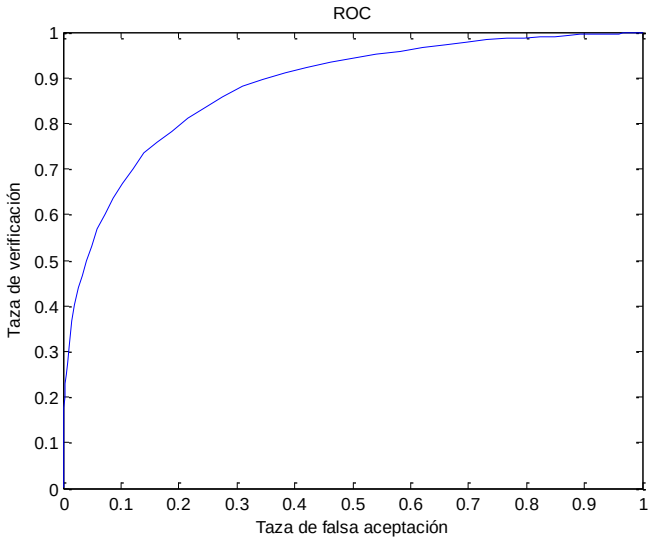


Figura 3.63: Grafica ROC

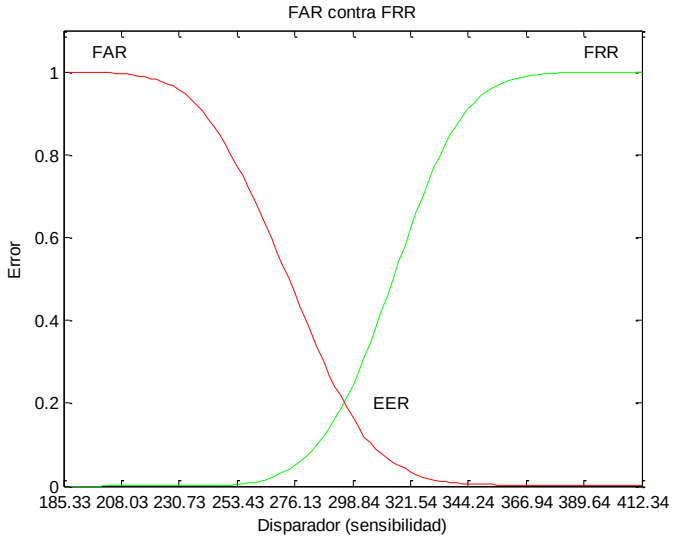


Figura 3.64: Grafica FRR vs. FAR

Un punto que vale la pena mencionar, es que debido a que en la presente tesis se analizarán cientos de combinaciones o configuraciones de parametrización del motor biométrico, se agregó una calificación simple, ésta consta del porcentaje de aciertos, el porcentaje de errores y el EER (Equal Error Rate) o tasa de error igual que es el punto donde se intersectan las dos gráficas como se ve en la figura 3.64, la función que analiza los resultados del sistema se encuentra en el anexo 1.

Ya con estos resultados podemos considerar concluido el presente capitulo que trata de la implementación del sistema de identificación biométrica por voz, el sistema es capaz de procesar muestras de voz y almacenar modelos "comparables" de distintos locutores en un esquema como se muestra a continuación:

Figura 3.65: Proceso de enrolamiento

Este diagrama simplifica muchos pasos comparativamente a como se muestra en la figura 3.59, sin embargo, representa lo mismo, entra un archivo con una voz humana, se realiza un pre-procesamiento que involucra acciones como el filtro de pre-énfasis o la detección de actividad vocal, se realiza la extracción de características que en nuestro caso son coeficientes cepstrales y sus variantes, para pasar finalmente a un modelado que en este caso se realiza por medio de una cuantización vectorial, permitiéndonos almacenar estos modelos.

Complementariamente el proceso de identificación es similar al anterior, pero involucra la búsqueda de modelos en la base de datos, la medición de distancias y la evaluación de resultados para generar un resultado, que en casos de verificación es sólo una decisión binaria, es o no es la misma persona, pero para los casos de identificaciones abiertas o cerradas es una lista de candidatos y una calificación para cada resultado.

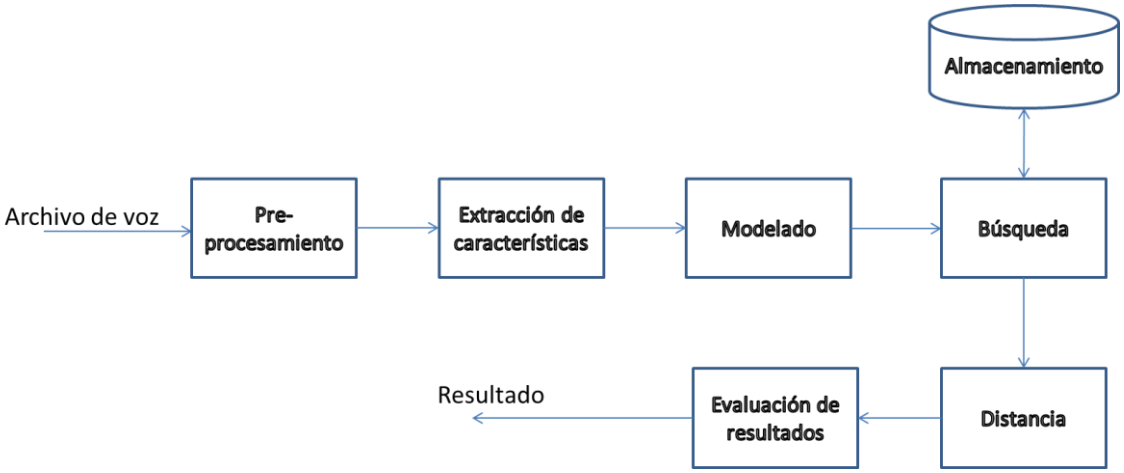


Figura 3.66: Enrolamiento y búsqueda biométrica

El sistema desarrollado en la presente tesis cuenta con los módulos para modelar un solo archivo de audio, para proceso de audios en lote, para generar el proceso de calibración con una o más configuraciones y para la identificación de un archivo de audio contra la base de datos que ya esté generada, ofreciendo como resultado una lista de candidatos conformada por

el nombre del archivo, el FRR, el FAR y un resultado ponderado, esto para cada uno de los audios que pasaron el umbral especificado por el usuario.

Con este punto se considera que el sistema está completo, dejando paso a la selección de los parámetros óptimos de trabajo, la solución permite seleccionar 24 parámetros de trabajo que en conjunto pueden ofrecer más de 11,583 billones de combinaciones, por lo que si tomamos como referencia 1,000 grabaciones por prueba con un tiempo de aproximadamente 15 segundos por modelado nos daría un aproximado de 5.5 billones de años para probar todas las combinaciones posibles en una computadora con un procesador i7 de Intel.

Esto hace que la prueba de todas las configuraciones por el método de la fuerza bruta como coloquialmente se le conoce a esta aproximación en el medio informático, sea imposible de realizar, por tal motivo se propondrá una metodología para la selección de las pruebas de calibración a realizar en el capítulo 4.

4. PRUEBAS DE OPERACION

Este cuarto capítulo representa la aportación principal de la presente tesis, la porción de investigación original del trabajo, ¿Cuáles son los parámetros que maximizan la eficiencia del motor biométrico?

Existen trabajos aislados que estudian el efecto de un filtro de pre-énfasis o que postulan una mejora a un algoritmo VAD (detección de actividad vocálica), pero no existen trabajos que se aproximen al enorme problema de estudiar si están ligados los efectos del uso de algún determinado parámetro en conjunto con otro, o que muestre en un sólo trabajo el efecto de los 24 parámetros que el sistema ***akutzika*** permite estudiar, por lo que sin más nos adentraremos en el problema.

*4.1 Parámetros de trabajo de **akutzika***

Los parámetros que el sistema nos permite estudiar son los siguientes:

- Tamaño de la ventana para Fourier: El sistema ofrece las siguientes opciones
  - 64 [datos]
  - 128 [datos]
  - 256 [datos]
  - 512 [datos]
  - 1024 [datos]
  - 2048 [datos]
- Traslape de la ventana de Fourier: El sistema ofrece en sí una aproximación abierta, se puede seleccionar libremente un valor de 0 a 1, pero se sugeriría probar con las siguientes configuraciones:
  - 0 (sin traslape)
  - 0.1
  - 0.2
  - 0.3
  - 0.4
  - 0.5
  - 0.6
  - 0.7
  - 0.8
  - 0.9
- Tipo de filtro para Fourier: Aunado al traslape y para mejorar el efecto adverso que se crea en los bordes de la ventana, tenemos la implementación de un filtro que suavice los bordes de la ventana, este filtro puede ser del tipo:
  - Ventana modificada Bartlett-Hanning
  - Ventana Bartlett
  - Ventana Blackman
  - Ventana Blackman-Harris
  - Ventana de Bohman
  - Ventana de Chebyshev

  - Ventana de parte alta plana
  - Ventana Gaussiana
  - Ventana de Hamming
  - Ventana de Hann
  - Ventana de Kaiser
  - Ventana de Nuttall (Blackman-Harris de N puntos)
  - Ventana de Parzen
  - Ventana rectangular
  - Ventana de Taylor
  - Ventana triangular
  - Ventana de Tukey
- **Valor de alfa para pre-énfasis:** En esta opción el valor -1 sería no aplicar pre-énfasis, fuera de esto, el valor se puede seleccionar libremente entre 0 y 1, pero se sugiere probar al menos con los siguientes valores:
  - -1
  - 0
  - 0.1
  - 0.2
  - 0.3
  - 0.4
  - 0.5
  - 0.6
  - 0.7
  - 0.8
  - 0.9
  - 0.95
  - 1
- **Número de coeficientes MFCC:** De nueva cuenta el sistema permite seleccionar libremente este parámetro, pero por lo general es recomendado trabajar al menos 2 coeficientes por cada formante a estudiar, aunado a un par de coeficientes extras para inicio y fin de la señal, al trabajarse con audios a 11,025 [Hz] el número máximo de formantes que se podrían llegar a visualizar, sobre todo en voces femeninas es de 4, motivo por el cual se sugiere tomar al menos 10 coeficientes, una vez explicado esto se sugeriría probar al menos con las siguientes configuraciones:
  - 10
  - 20
  - 25
  - 30
  - 35
  - 40
  - 45
  - 50
- **Traslape de los filtros:** Los filtros que nos permiten obtener la envolvente como se aprecia en la figura 2.41 pueden tener también un traslape entre filtros contiguos, este

parámetro de nueva cuenta es libre en valores de 0 a 1, pero se sugiere probar con los siguientes valores:

- 0
- 0.1
- 0.2
- 0.3
- 0.4
- 0.5
- 0.6
- 0.7
- 0.8
- 0.9

- **Tipo de filtro para el MFCC:** Al igual que en la parte de la transformada de Fourier se puede elegir aunado al traslape de los filtros la forma de estos mismos, teniéndose las siguientes opciones.
  - Ventana modificada Bartlett-Hanning
  - Ventana Bartlett
  - Ventana Blackman
  - Ventana Blackman-Harris
  - Ventana de Bohman
  - Ventana de Chebyshev
  - Ventana de parte alta plana
  - Ventana Gaussiana
  - Ventana de Hamming
  - Ventana de Hann
  - Ventana de Kaiser
  - Ventana de Nuttall (Blackman-Harris de N puntos)
  - Ventana de Parzen
  - Ventana rectangular
  - Ventana de Taylor
  - Ventana triangular
  - Ventana de Tukey
- **Frecuencia mínima de corte:** Otro parámetro que se puede implementar es un filtro paso altas que elimine la parte baja del espectro, ya que en ese punto se pueden encontrar componentes de la fuente de alimentación (50-60 [Hz]) que resulten perjudiciales a la identificación, de nuevo el filtro es abierto, pero se sugiere probar con los siguientes valores:
  - 50
  - 100
  - 150
  - 200
  - 250
  - 300
- **Frecuencia máxima de corte:** De igual manera que el punto anterior se puede implementar un filtro paso bajas que elimine la parte alta de la señal, esto en miras de

eliminar las frecuencias altas que en el caso de canal telefónico pueden contener un reflejo de las ultimas frecuencias y que resultaría también dañino a la identificación, es también un parámetro abierto pero se sugiere probar al menos los siguientes valores:

  - 5000
  - 4500
  - 4000
  - 3500
  - 3000

- **Reduce vectores:** Este parámetro se refiere a la normalización opcional de los vectores MFCC obtenidos
  - Si
  - No
- **Centra vectores:** Este parámetro se refiere a un proceso de sustracción de la media de cada uno de los valores, este proceso es conocido como CMS (Cepstral Mean Subtraction) y a recomendación de algunos autores mejora la eficiencia de los sistemas de identificación.
  - Si
  - No
- **Orden de aproximación para las deltas:** Dado que los deltas son calculados por una aproximación polinomial se puede elegir el orden de dicho polinomio, es un parámetro libre, pero se sugiere probar los siguientes valores:
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8
- **Utilizar coeficientes delta:** El conjunto de valores de los MFCC pueden contener opcionalmente una evaluación de la primera derivada (velocidad) de los datos para tener información dinámica del comportamiento, ésta puede ser opcional.
  - Si
  - No
- **Utilizar coeficientes delta-delta:** El conjunto de valores de los MFCC pueden contener opcionalmente una evaluación de la segunda derivada (aceleración) de los datos para tener información dinámica del comportamiento, ésta puede ser opcional.
  - Si
  - No
- **Utilizar coeficientes de energía:** El conjunto de valores de los MFCC pueden contener opcionalmente una evaluación de la energía de la señal y de sus deltas, ésta puede ser opcional.
  - Si
  - No

- **Tipo de transformación:** Este parámetro se refiere a que si se desea obtener los coeficientes utilizando la escala lineal, Mel o Bark, en teoría solo utilizando la Mel se trataría de MFCC, pero en general se consideran otras variantes:
  - Mel
  - Bark
  - Lineal
- **Transformación secundaria:** Para la obtención del cepstrum se requiere realizar una segunda transformación a la señal que ya se tiene procesada, esta segunda transformación puede ser DCT (Discrete Cosine Transform) que es una variante de la segunda opción, la DFT (Discrete Fourier Transform), ambas son similares pero una trabaja con los números imaginarios completos y la otra sólo con la parte real.
  - DCT
  - DFT
- **Tamaño del modelo por cuantización vectorial:** Este dato se refiere al número de vectores que representara o modelara al locutor, cuanto más grande más lento el modelado, cuanto más pequeño crece el riesgo de poca representatividad, se tienen las siguientes opciones:
  - 8
  - 16
  - 32
  - 64
  - 128
  - 256
- **Desplazamiento de los vectores:** La literatura marca que se debe seleccionar un valor pequeño para desplazar los vectores en el proceso de creación del modelo, pero no existe regla para seleccionar el valor, de nuevo es abierto y se sugieren los siguientes valores:
  - 0.01
  - 0.015
  - 0.02
  - 0.025
  - 0.03
- **Aplicar detector de actividad vocal:** Este como el último pre-procesamiento que se implemento es opcional, su función es eliminar las pausas y el ruido.
  - Si
  - No
- **Tamaño de la ventana en [ms]:** El tamaño de la ventana de estudio del VAD es abierto, se sugieren ventanas pequeñas, pero no hay regla general, se sugiere probar al menos los siguientes valores:
  - 5
  - 10
  - 15
  - 20
- **Escalón de corte en energía:** Los escalones de corte del VAD (VAD con algoritmo doble, por energía y frecuencia) pueden ser manuales o automáticos, al no ser el tema

primario de la tesis decidí simplificar este paso y dejar el parámetro de “que es ruido” manual, se sugiere probar con los siguientes valores:

- 1
- 2
- 3
- 4

- **Escalón de corte en frecuencia:** Los escalones de corte del VAD (VAD con algoritmo doble, por energía y frecuencia) pueden ser manuales o automáticos, al no ser el tema primario de la tesis decidí simplificar este paso y dejar el parámetro de “que es ruido” manual, se sugiere probar con los siguientes valores:
  - 1
  - 2
  - 3
  - 4

Para las pruebas del sistema opte por crear una función que me permita crear las combinaciones posibles de acuerdo a los parámetros que se desean cambiar, en su forma general y con todas las opciones descritas se obtiene un total de 11,583 billones de combinaciones que es imposible de probar, por tal motivo más adelante se propone una manera alternativa de evaluar el sistema. La función mediante la cual se crean las combinaciones en un archivo que el sistema comprende se encuentra en el anexo 1.

4.2 Divide y vencerás

En las ciencias de la computación, el término divide y vencerás (DYV) hace referencia a uno de los más importantes paradigmas de diseño algorítmico. El método está basado en la resolución recursiva de un problema dividiéndolo en dos o más sub-problemas de igual tipo o similar. El proceso continúa hasta que éstos llegan a ser lo suficientemente sencillos como para que se resuelvan directamente. Al final, las soluciones a cada uno de los sub-problemas se combinan para dar una solución al problema original.

Figura 4.1: Divide y vencerás

La división de nuestro problema en varios sub-problemas nos puede ayudar a reducir drásticamente el número de configuraciones a probar, sólo tenemos que seleccionar con cierto

cuidado que dividir y que dejar, para ejemplificar rápidamente este proceso usaremos un postulado simple.

Digamos que queremos probar todos los tipos de ventanas que se implementaron, en total 17, deseamos ver cómo se comporta el sistema con estos 17 tipos de ventana en la transformación de Fourier, pero también en los tipos de filtros para aplicar la transformación cepstral y que además deseamos probar con digamos 10 tamaños diferentes de ventanas para la transformación de Fourier, esto significa 17x17x10=2,890 combinaciones, que si lo conjuntamos con una prueba digamos de 1,000 grabaciones significan 2.89 millones de modelos y 2,890 millones de comparaciones biométricas, por lo que a un promedio de 20 segundos por modelo nos daría algo así como 1.9 años de tiempo de procesamiento en una computadora de escritorio de buenas características (procesador i7 de Intel).

Ahora si aplicamos una división bajo el razonamiento de que es poco probable que el tipo de filtro que apliquemos al ventaneo, el tamaño de la ventana y el tipo de filtros del banco para la transformación cepstral tienen poca o nula relación entre sí, podríamos quizás realizar 17+17+10=37 combinaciones, aún más, con los resultados anteriores podríamos probar con los mejores 3 resultados de cada uno de los experimentos y hacer 3x3x3=27 combinaciones, que aunado a las 37 anteriores nos daría un tiempo aproximado de computo de 15 días, que ya es una mejora considerable contra los casi dos años originales.

4.3 Etapa de pruebas

Con la idea de dividir el problema en partes más simples empezaremos a dividir, del diagrama que ya habíamos presentado de manera global, vemos que el pre-procesamiento se ve como algo "separado", algo opcional que podemos o no realizar previo a la extracción de características, así que es un buen candidato para iniciar nuestra división.

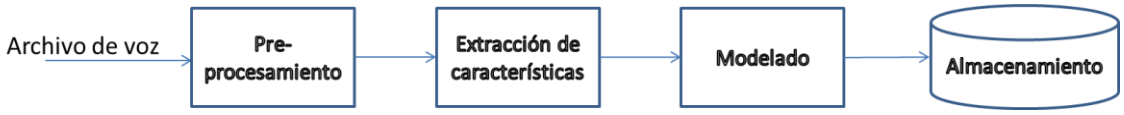


Figura 4.2: Proceso de enrolamiento

4.3.1 Pre-procesamiento

Revisando que es el pre-procesamiento del habla podemos observar en la figura 4.3 que nuestro sistema comprende dos procesos separados, el detector de actividad vocal y el filtro de pre-énfasis, su colocación sigue una regla muy simple, ¿para qué aplicar pre-énfasis a donde no hay voz?, en este tenor primero se sugiere discernir si se trata de voz o de ruido y después procesar esta voz, pero aún podemos hacer más análisis previo.

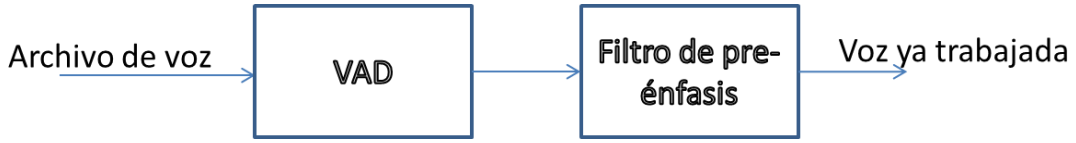


Figura 4.3: Pre-procesamiento de la señal

Estos procesos, como dijimos antes, pueden ser opcionales, podría no pasar nada si los omitimos, podría mejorar el resultado e incluso podría empeorar, pero, ¿Cómo estar seguros?, la respuesta es experimentando, iniciaremos con el módulo de detección vocal, este presenta

dos juegos de parámetros básicos, el tamaño de la ventana que usaremos y la sensibilidad del detector.

En esta ocasión el tamaño de la ventana no obedece a un precepto científico como la cuasi-estabilidad de la voz humana en periodos cortos, responde simplemente al razonamiento de que tanto deseamos cortar o ignorar, lo cual es más fácil de apreciar gráficamente, en la figura 4.4 observamos al módulo trabajando con una ventana de 10 [ms], posteriormente con una ventana de 50 [ms] y finalmente con una de 100 [ms].

Como podemos ver es bastante aparente que una ventana pequeña es lo mejor para evitar errores, disminuyendo la probabilidad de que secciones no deseadas pasen el filtro como en el caso de los 50 [ms], así como la omisión de información útil como en el caso de los 100 [ms], sin embargo, a nivel estadístico no es tan fácil tomar la decisión, motivo por el cual se realiza el experimento correspondiente.

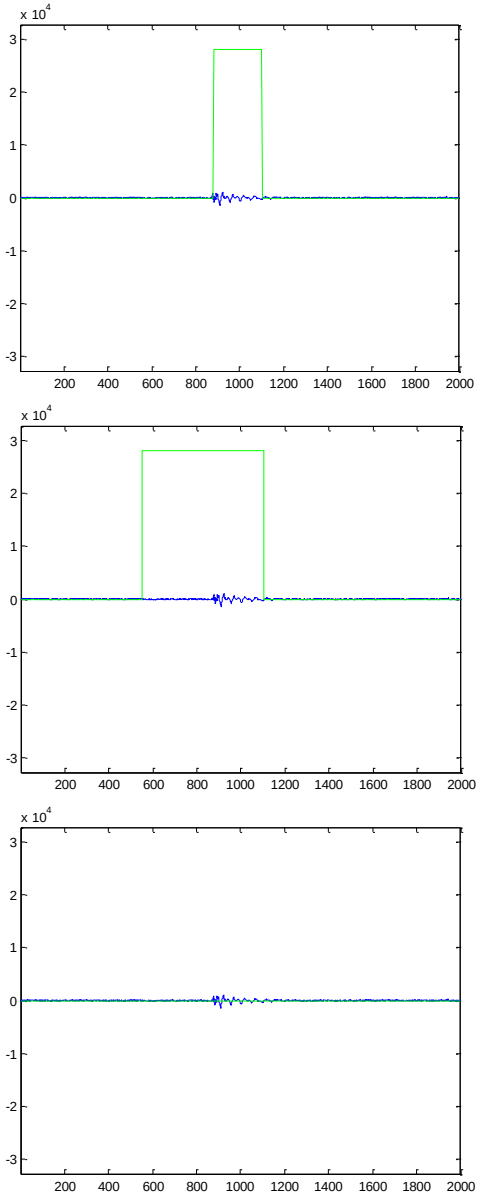


Figura 4.4: Tamaño de la ventana del VAD

En este caso se tiene que el resultado es como se había propuesto, el tamaño de la ventana no es muy importante, sin embargo, una ventana pequeña es mejor pero tampoco se obtiene una mejora significativa al disminuir fuertemente el tamaño de la ventana, pero sí un incremento en el tiempo de cómputo debido a los cálculos adicionales requeridos, en la figura 4.5 se aprecian estos resultados de manera gráfica, en el eje horizontal vemos la eficiencia y en la vertical el tamaño de la ventana.

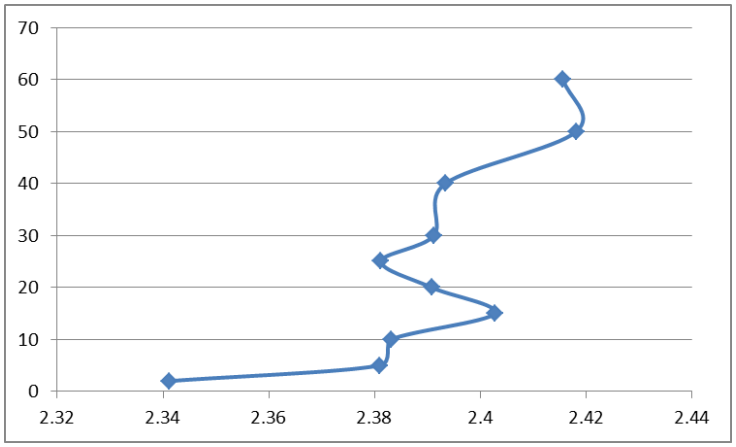


Figura 4.5: Resultados del tamaño de la ventana del VAD

Ahora que tenemos un tamaño recomendado empíricamente nos queda el asunto de la sensibilidad, algunos autores sugieren que es mejor utilizar un mecanismo automatizado para definir la sensibilidad, sin embargo, aquí prefiero usar la aproximación manual para obtener un juego de parámetros "ideales" que pueden después ser incorporados a una versión mejorada que ya tenga esta decisión incorporada, por ahora debemos marcar la sensibilidad en un nivel del 0 al 100 por cada método empleado (energía y frecuencias máximas), esto mismo se ilustra de nuevo para entender mejor el efecto.

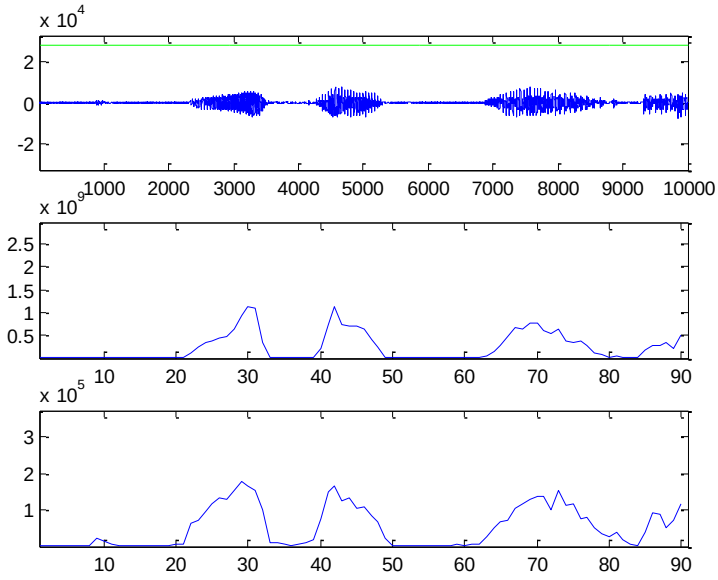


Figura 4.6: Efecto de la sensibilidad del VAD (todo pasa)

En la figura 4.6 la sensibilidad es cero, todo pasa, lo que quiere decir que es lo mismo que tener el detector de actividad vocal apagado, en la figura 4.7 vemos una parametrización de 2 y 3 respectivamente, en la figura 4.8 de 20 y 20, como vemos el sistema se hace más selectivo en

qué es voz y qué no lo es, realmente no hay una directiva que indique qué sería lo ideal, después de todo no nos interesa saber qué se dice, sino quién lo dice.

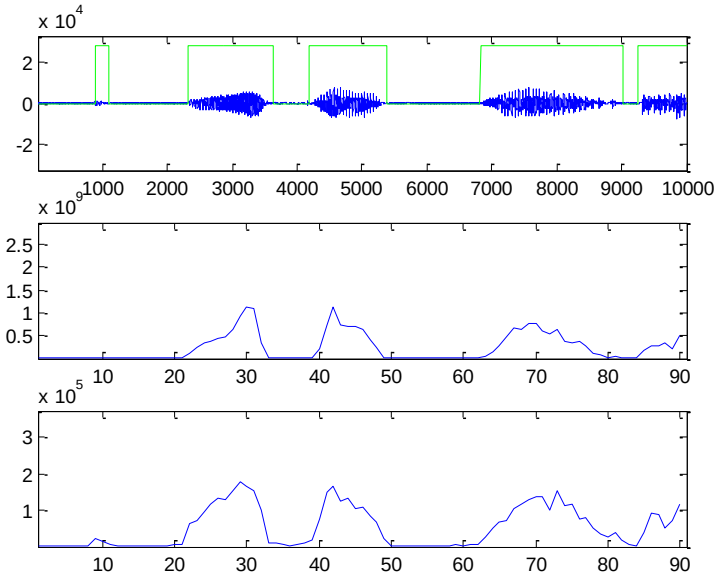


Figura 4.7: Efecto de la sensibilidad del VAD (valor 2 y 3)

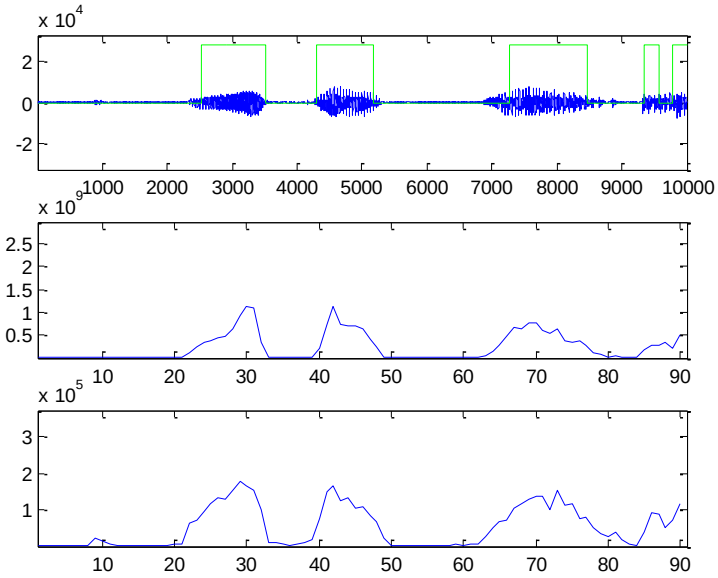


Figura 4.8: Efecto de la sensibilidad del VAD (valor 20 y 20)

Para este parámetro tenemos en realidad dos elementos que podemos configurar, la sensibilidad de la detección por energía y la sensibilidad de la detección por frecuencias máximas, en el experimento real se realizó una combinación de una sensibilidad de 1 a 6 para ambos valores, lo que resultó en 36 combinaciones que se muestran en la figura 4.9, en la horizontal se aprecia el índice de sensibilidad, en la vertical la eficiencia y en un color distinto cada serie de combinaciones de la sensibilidad por energía con sus diversas sensibilidades por frecuencia.

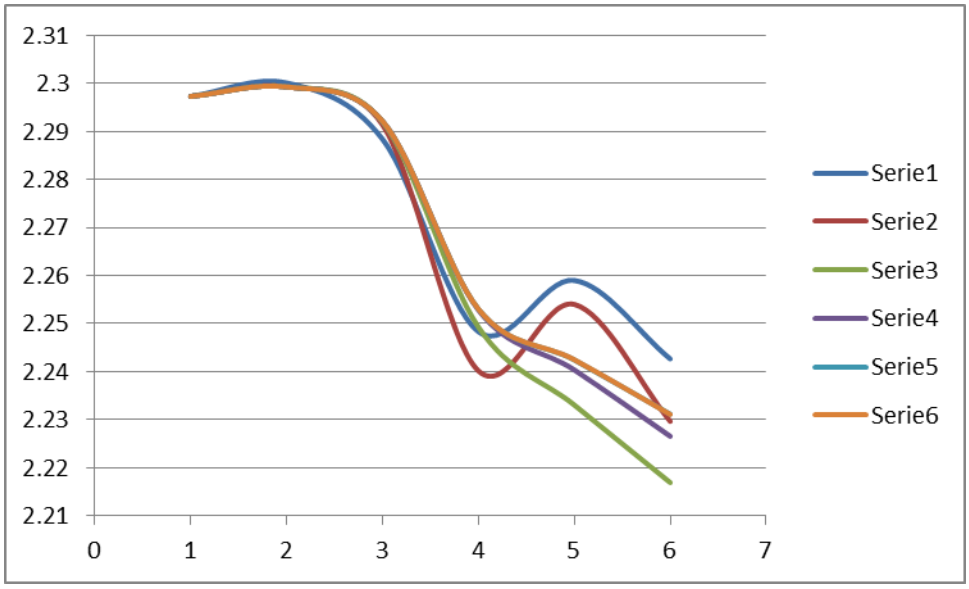


Figura 4.9: Resultados en el cambio de la sensibilidad

Con esto quedan definidos los parámetros del detector de actividad vocal, esto nos permite avanzar al filtrado de pre-énfasis, éste es un procesamiento sumamente simple, como ya se mostró en las figuras 3.1 y 3.2, la función de este filtro es disminuir las diferencias a nivel energético en la distribución frecuencial de las componentes de la voz humana, compensando así las pérdidas que sufre esta señal por su proceso natural de formación.

En la figura 4.10 se muestra de nueva instancia un nuevo ejemplo de un espectro instantáneo de una porción cualquiera sin filtrado de pre-énfasis y posterior a dicho filtrado, en la figura 4.11 se muestra de manera simultánea ambos espectros para apreciar mejor el efecto.

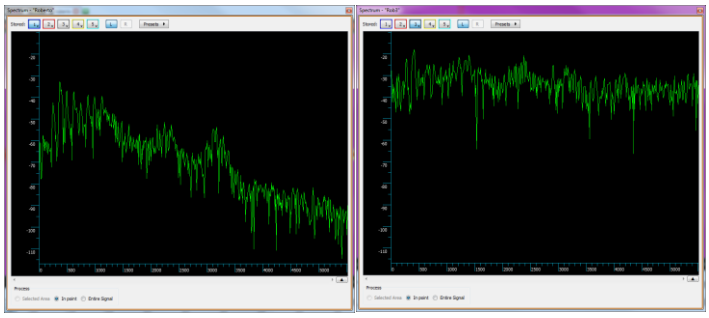


Figura 4.10: Habla antes y después del pre-énfasis

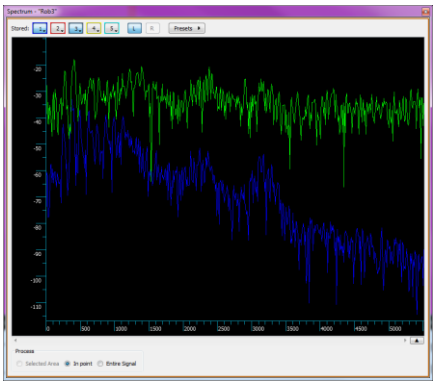


Figura 4.11: Comparativo de espectro con y sin pre-énfasis

Para las pruebas de filtrado se realizaron 23 experimentos, cuyo resultado se puede apreciar en la figura 4.12, en la horizontal se ve la eficiencia y en la vertical el valor de alfa, esto da por concluido esta primera parte de la experimentación hacia obtener la configuración ideal, aún no decidimos si es mejor aplicar o no aplicar el detector de actividad vocal o el pre-énfasis, por ahora sólo sabemos cuáles serían los mejores parámetros de cada uno en caso de decidir emplearlos que será visto más adelante.

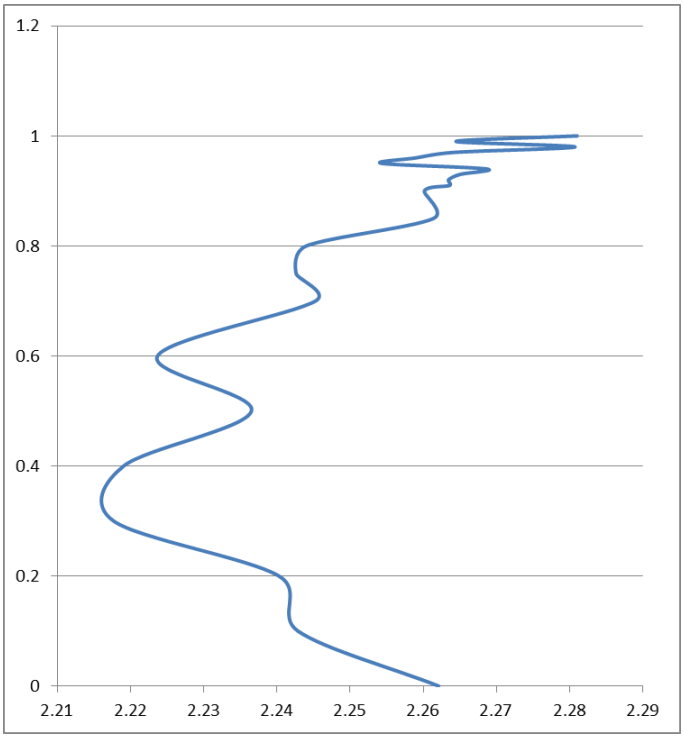


Figura 4.12: Comportamiento por el valor de alfa

4.3.2 Modelado

Esto puede parecer un salto un poco extraño si seguimos el orden de la figura 4.2, en realidad simplemente evitamos entrar al área que presenta el mayor número de parámetros que podemos ajustar, por lo tanto pasaremos al área final que lidiará con los resultados de la extracción de parámetros, tenemos un juego de parámetros que es aún muy grande como para compararlo, en este caso en específico se disminuye el tamaño del "objeto" de comparación por la técnica de la cuantización vectorial, pero ¿cuáles son los parámetros de operación más convenientes?

En este caso tenemos dos parámetros que podemos modificar, el número de palabras código que conformarán el libro código y la distancia que se utilizará para desplazar cada palabra código cuando se estén dividiendo en el ciclo iterativo, de inicio veremos el efecto que tiene el valor de la distancia que desplazamos a cada palabra código y que quedaría como se puede apreciar en la figura 4.13, en la horizontal se ve la eficiencia y en la vertical la distancia.

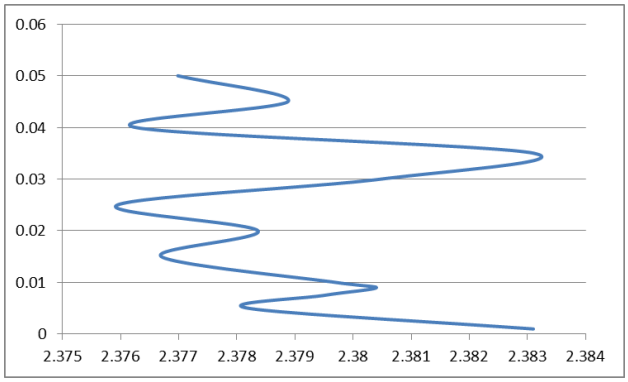


Figura 4.13: Efecto del cambio de la distancia a desplazar las palabras código

De esta figura vemos que realmente la distancia no es muy importante, la variación en la eficiencia es mínima, esto obedece a que ésta es sólo una distancia inicial por la que se desplazan los vectores, pero ésta juega un rol poco significativo en el ajuste final de cada palabra código, sin embargo, seleccionaremos aquella que maximiza la eficiencia del sistema.

Como segunda parte de este modelado nos falta definir el número de palabras por libro código, el comportamiento de este parámetro es como se muestra en la figura 4.14, aquí la horizontal muestra la eficiencia y la vertical el tamaño del libro.

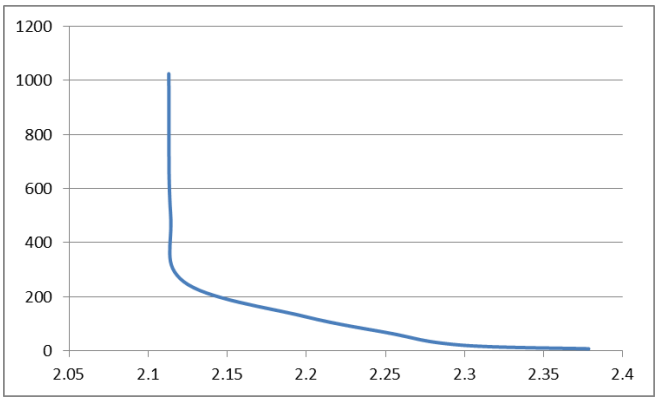


Figura 4.14: Afectación por el número de palabras código

De esta figura vemos que contrario a lo que quizás podríamos pensar, un libro compuesto de un número grande de palabras empeora el resultado y hace cada vez más lento el procesamiento, en este sentido nos beneficia este resultado, se acortará el tiempo de generación del modelo en aproximadamente la mitad, pudiendo lograrse tiempos aproximados de 12 a 15 segundos por modelo, que acortará significativamente el tiempo de procesamiento necesario para el resto de las pruebas.

Como podemos apreciar en la figura 4.15, el modelo por cuantización vectorial simplifica en gran medida la cantidad de información que representara al locutor, en este caso se muestra una extracción de parámetros con más de 6,400 vectores contra un libro código de 8 palabras.

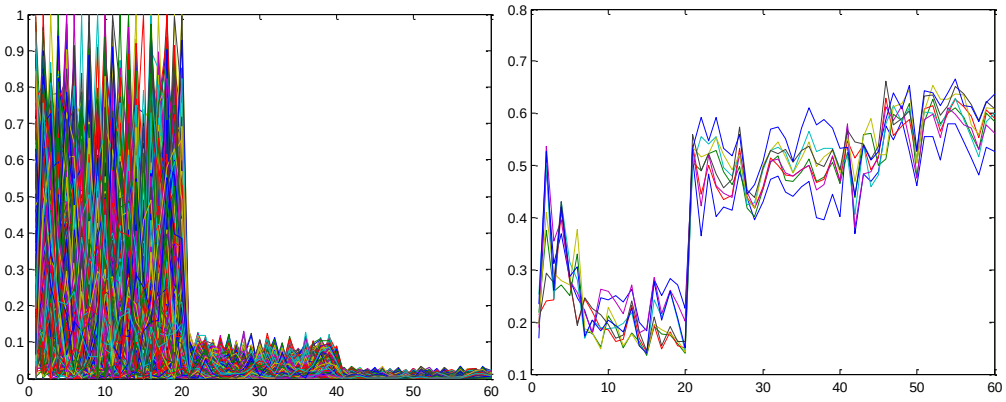


Figura 4.15: Comparativa del vector completo y el modelo por VQ

4.3.3 Extracción de parámetros

En esta última parte se llevarán a cabo la mayor parte de los experimentos de refinamiento del sistema, como podemos ver del último párrafo nos beneficiara significativamente el resultado previo, ya que el modelado consumía buena parte del tiempo de proceso.

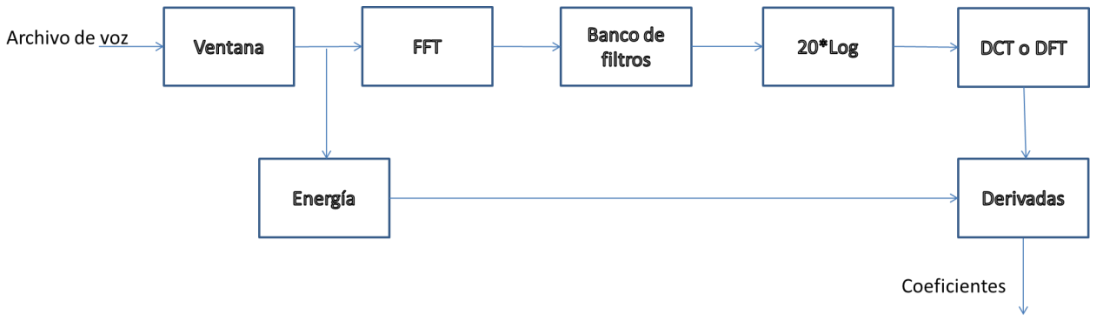


Figura 4.16: Extracción de parámetros

Ya dentro de la extracción de parámetros podemos tomar el proceso de enventanado como primer candidato a dividir, tenemos para esta parte el tamaño de la ventana, que en sí define la resolución temporal y por ende la espectral, el solapamiento de las ventanas y la forma de las ventanas, por sí solos estos tres parámetros pueden dar más de 1,300 combinaciones (17 tipos de ventanas, digamos unas 10 opciones de solapamiento y unas 8 dimensiones de la ventana), por esto veremos el tipo de ventana por separado.

Los tipos de ventanas que tenemos definidos en el sistema son los siguientes:

1. Ventana modificada Bartlett-Hanning
2. Ventana Bartlett
3. Ventana Blackman
4. Ventana Blackman-Harris
5. Ventana de Bohman
6. Ventana de Chebyshev
7. Ventana de parte alta plana
8. Ventana Gaussiana
9. Ventana de Hamming
10. Ventana de Hann
11. Ventana de Kaiser

12. Ventana de Nuttall (Blackman-Harris de N puntos)
13. Ventana de Parzen
14. Ventana rectangular
15. Ventana de Taylor
16. Ventana triangular
17. Ventana de Tukey

Su efecto para fines de la identificación de locutores, queda reflejado como sigue:

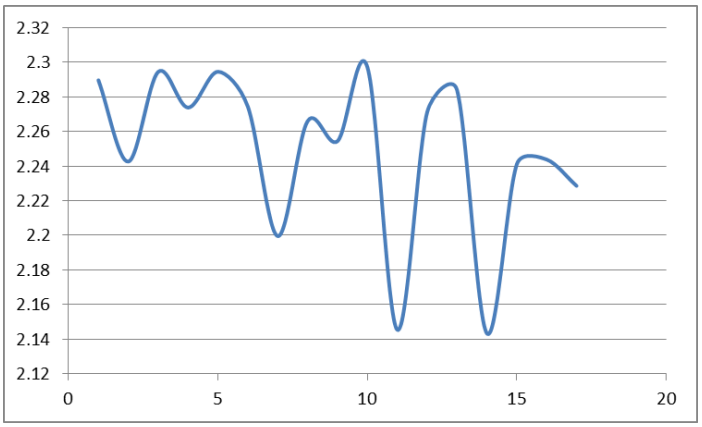


Figura 4.17: Comportamiento con los distintos tipos de ventana

En la gráfica de la figura 4.17 tenemos en el eje horizontal el número de ventana y en el vertical la calificación, como podemos ver existen varias ventanas con las que se obtienen buenos resultados, sin embargo, emplearemos sólo una en lo sucesivo para no aumentar el número de combinaciones, en este caso emplearemos la ventana de Hann para los subsecuentes cálculos.

En lo que respecta al tamaño y porcentaje de solapamiento se presentaran las gráficas individuales por cada tamaño de ventana, cada color representa un tamaño de ventana, en las horizontales se ve la calificación o eficiencia y en el vertical el porcentaje del solapamiento, en general se observa un mejor comportamiento en ventanas de 256 o 512 puntos y con solapamientos altos.

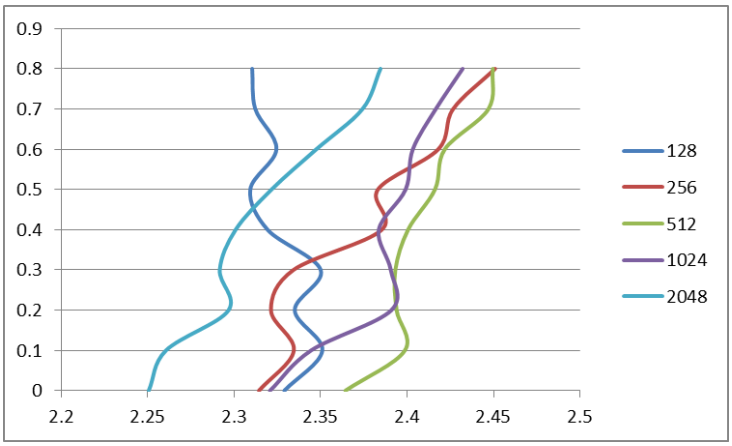


Figura 4.18: Efecto del tamaño y solapamiento de las ventanas

Ahora, en lo que respecta al banco de filtros, éste incluye bastantes parámetros, de nueva instancia empezamos por los 17 tipos de filtros, un número no determinado de coeficientes, un

solapamiento variable y un filtro paso banda opcional (combinación de los filtros paso altas y paso bajas) que elimina partes que pueden resultar perjudiciales al análisis.

En este sentido se realizaron de nueva cuenta 17 experimentos para ver cuál es el filtro que mejor se amolda a nuestras necesidades, como se puede ver en la figura 4.19 de nueva cuenta el eje horizontal se muestra el número del filtro (mismos filtros de la figura 4.17) y en el eje vertical se muestra la calificación del sistema.

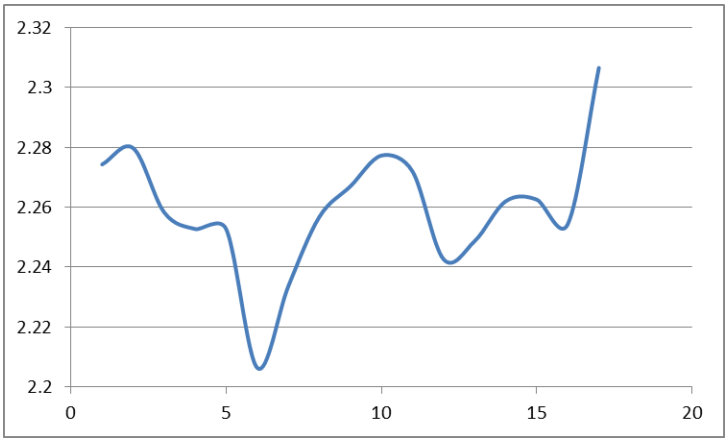


Figura 4.19: Filtros para el banco de filtrado

Como podemos ver, el comportamiento de las diferentes formas de filtros en el banco es distinto al comportamiento que teníamos en el filtrado que se aplica en el ventaneo de la señal, aquí nuestra mejor opción es el filtro de Tukey, sin embargo, hay otros puntos a considerar, mismos que veremos más adelante.

Pasando al número de coeficientes que se calcularan y el traslape de los filtros, en este punto se corrieron 80 experimentos que nos ayudaran a comprender mejor cómo se comporta el sistema, en la figura 4.20 cada color representa un porcentaje distinto de solapamiento, en el eje horizontal se puede ver la calificación del sistema y en el eje vertical el número de coeficientes.

De esta prueba se detectó un comportamiento normal pero inesperado, me percate que algunas combinaciones producían errores en la generación del modelo, vectores de ceros, lo que de manera regular no debe suceder, en este sentido investigue la causa de este vector de ceros y encontré que se producía desde el banco de filtros, el comportamiento indeseable se da cuando el número de coeficientes crece, lo que concuerda con el hecho de los 4 experimentos fallidos, dos con 45 coeficientes y dos con 50 coeficientes.

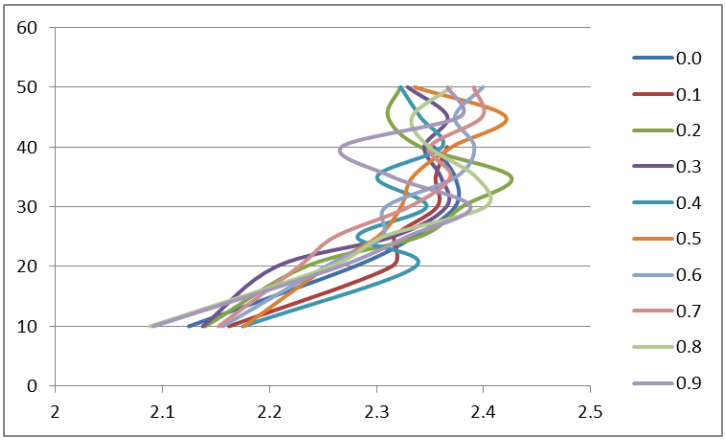


Figura 4.20: Número de coeficiente y porcentaje de solapamiento

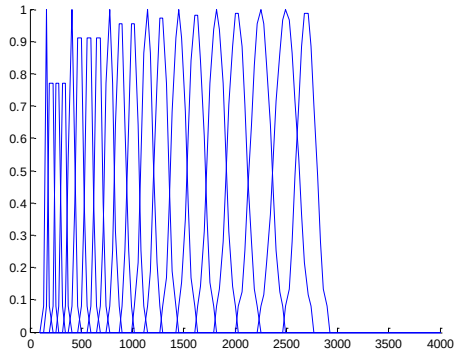


Figura 4.21: Banco de 20 coeficientes con filtro Hamming

Como se ve en la figura 4.21, los filtros tienen una forma tal que recorta la señal, pero si el número de coeficientes sube demasiado el "tamaño" de cada filtro disminuye, si el solapamiento sube el ancho del filtro sube, pero si el solapamiento baja y el número de coeficientes sube esto maximiza el adelgazamiento del filtro.

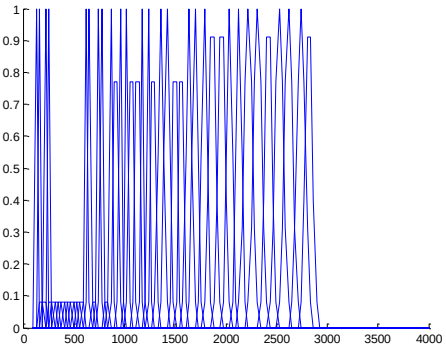


Figura 4.22: Banco de 50 coeficientes con filtro Hamming

De la figura 4.22 se ve que el ancho de cada filtro es menor, en este sentido los filtros pueden llegar a ser tan pequeños que valgan cero, lo que causaría en los sucesivos cálculos que todo llegará a valer cero, en este sentido y para evitar ese tipo de errores se probó de nueva cuenta con los resultados obtenidos, aprovechando además esto para corroborar la validez de la división del problema general.

Por lo ya expuesto se realizaron otras 64 pruebas, esta se compone de los mejores 4 resultados de cada una de las dos pruebas anteriores que versan sobre los coeficientes cepstrales, los resultados de esta prueba no concuerdan al 100% con los resultados anteriores, mismo que

indica que existe una correlación entre el tipo de filtro, el número de los coeficientes y el traslape de los mismos.

Estos resultados ligeramente distintos sugieren tomar con más cautela la división de las pruebas, la división del problema general en sub-problemas hace posible la experimentación que de otra manera no sería viable, pero tampoco se debe tomar a la ligera la correlación que guarda un elemento con otro para buscar los mejores resultados posibles.

Las gráficas de la figura 4.23 muestran por separado el comportamiento con un elemento estático, el número de coeficientes cepstrales, mostrando en distintos colores los solapamientos, el eje horizontal muestra la calificación y el vertical el tipo de ventana.

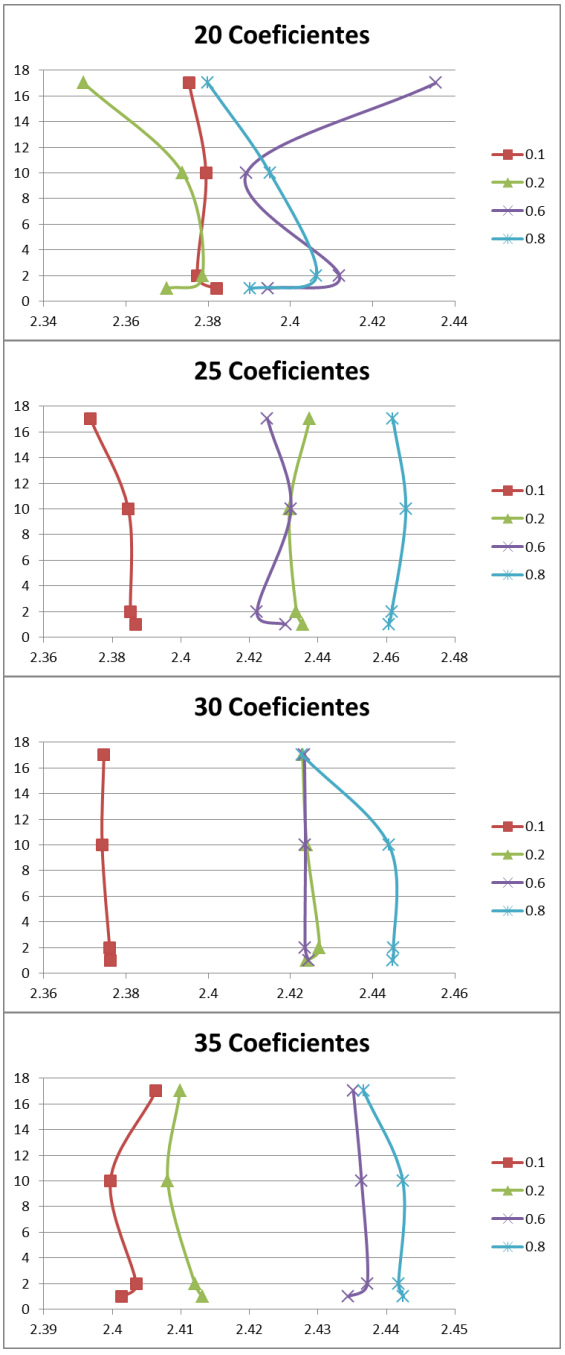


Figura 4.23: Segunda ronda de experimentos MFCC

La mejor calificación corresponde en general a un banco de filtros Hann con mucho solapamiento y un número relativamente bajo de coeficientes, pero aún así nos falta analizar un aspecto más, el filtro paso banda, aquí se recomienda cortar la parte alta y baja del audio para remover porciones que no son voz, por lo que se realizaron 42 experimentos, en la figura 4.24 cada color representa una frecuencia baja y su comportamiento con respecto a la frecuencia alta, en el eje horizontal se aprecia la calificación y en el vertical los Hertz.

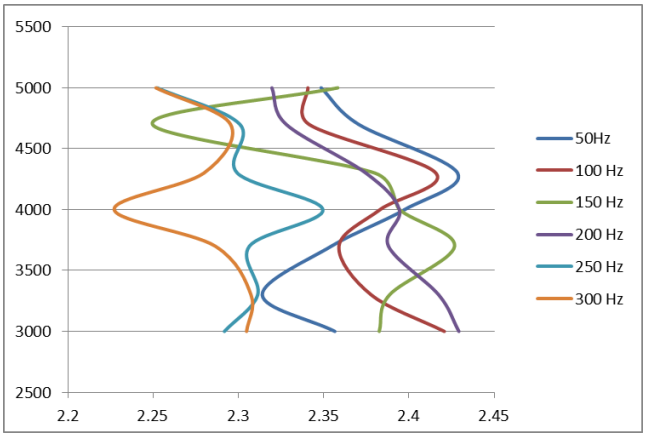


Figura 4.24: Comportamiento del filtro paso banda

De este análisis se corrobora el planteamiento inicial de que la información en la parte baja y alta del espectro poco ayuda en la identificación de locutores, mismo que nos deja prácticamente al final de nuestra experimentación, ahora veremos el efecto del orden de aproximación de los coeficientes delta y delta-delta.

En la figura 4.25 se muestra en el eje horizontal la calificación y en el eje vertical con el orden de la aproximación polinomial.

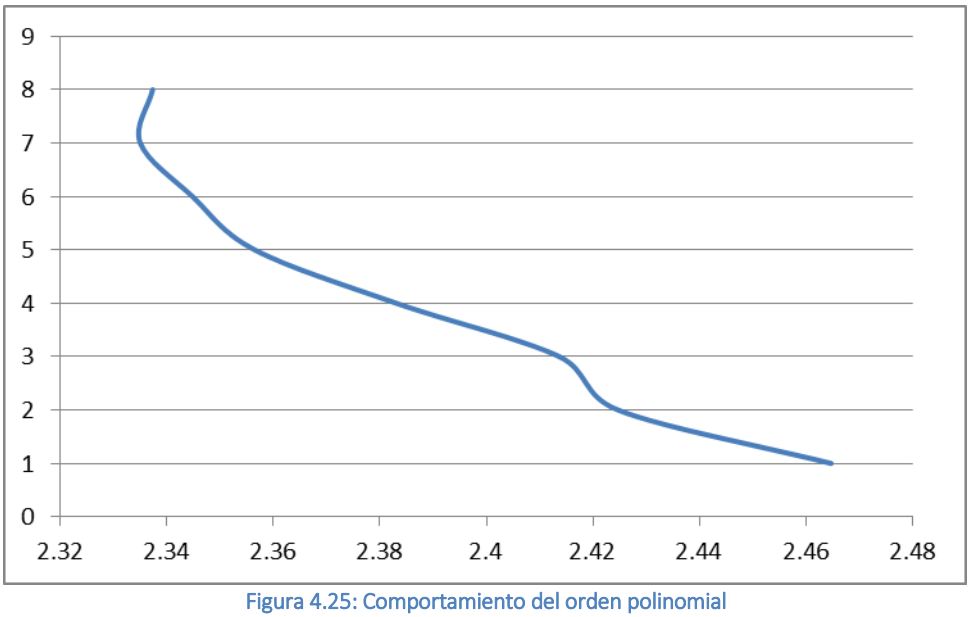


Figura 4.25: Comportamiento del orden polinomial

Aquí se ve que realmente el crecer el orden de aproximación no aporta gran cosa en términos de identificación, en general contradice el efecto esperado de que abarcar más información nos proporcionaría información valiosa sobre el comportamiento dinámico de la voz, la diferencia es poca, pero por mala suerte para esta técnica el crecer el orden no ayuda.

4.3.4 Extracción de parámetros, parte final

En esta sección final veremos la "gran prueba", como hemos visto realmente son pocos los parámetros que presentan un comportamiento estable como sería deseable, pensábamos que tendríamos muchas gráficas como la del orden de aproximación polinomial de la figura 4.25, pero la mayor parte de los comportamientos son un tanto impredecibles como es el caso del número de coeficientes cepstrales, su solapamiento y el tipo de filtro que los compone.

De hecho, como se vió en el experimento para los coeficientes cepstrales, si incluimos un tercer factor en una prueba que antes sólo contemplaba dos variables el resultado puede variar, esto nos dice que la correlación entre un parámetro y otro es significativa, motivo por el cual decidí ya no continuar empleando el algoritmo de divide y vencerás para no arriesgarme a "pasar por alto" casos como el anterior.

Con esto expuesto presentare la prueba total del resto de los parámetros faltantes, ésta contempla los siguientes puntos:

- Tipo de coeficientes cepstrales (Escala Mel, Bark o lineal)
- Transformada DCT o DFT
- Utilizar los coeficientes delta
- Utilizar los coeficientes delta-delta
- Utilizar los coeficientes de energía
- Utilizar el módulo VAD
- Utilizar reducción de vectores
- Centrar los vectores obtenidos

De entrada puede no parecer demasiado, pero esto representa 384 combinaciones que son de hecho 21 pruebas más que el total de las combinaciones probadas hasta el momento y que por sí mismas representan más de 130 días de procesamiento.

Para comprender mejor los resultados obtenidos empleare una técnica ligeramente distinta, por lo regular se modificaba uno, dos o máximo tres parámetros y analizábamos su comportamiento, pero en esta ocasión intervienen ocho parámetros, lo cual no hace tan fácil de separar que efecto puede causar el uso de una técnica o configuración determinada.

En este sentido, consideraremos de primera instancia la reducción y centrado de vectores como una unidad, esto dado que observe de los resultados que su impacto es mínimo sobre las calificaciones finales del sistema, para tal fin tomaré las combinaciones que se pueden formar con estos dos parámetros y que son:

Centrar	*Reducir*
Si	No
No	No
Si	Si
No	Si

Tabla 17: Combinaciones de centrado y reducción

El comportamiento de nueva instancia no es tan estable como quisiéramos, de acuerdo a nuestra propuesta los 384 experimentos se dividen en 96 grupos, de estos 96 grupos se detectaron los siguientes comportamientos:

- 23 no modifican su comportamiento a pesar de cualquiera de las 4 combinaciones
- 14 grupos sólo ven efecto adverso en 1 de sus combinaciones, pero no siempre es la misma combinación la que se ve afectada
- 9 grupos presentan dos configuraciones que son ligeramente superiores a las otras dos
- 50 grupos sólo tienen una configuración que maximiza su potencial

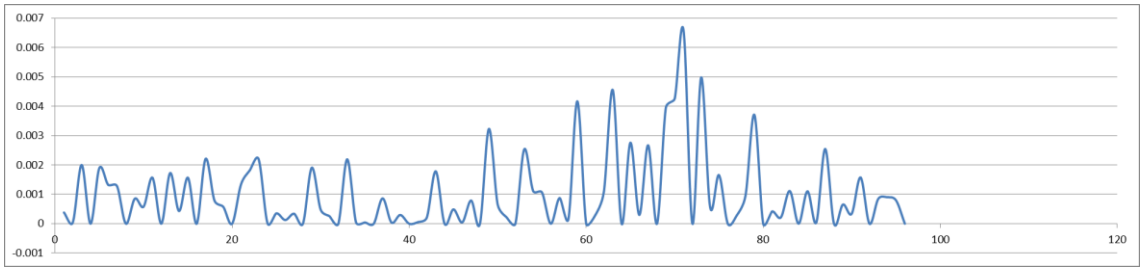


Figura 4.26: Ganancia en porcentajes

En la figura 4.26 vemos el comportamiento en cuanto a la ganancia obtenida por el uso del centrado o la reducción de los vectores, ésta podría parecer impactante alrededor del grupo 70, sin embargo, sólo representa una ganancia del 0.658% con respecto de la eficiencia lograda en dicho grupo, aunado a que el grupo que logra la mayor eficiencia presenta una ganancia del 0% por la aplicación de esta metodología, motivo por el cual, por lo que respecta a este sistema discontinuaremos el uso de estas opciones ya que consumen tiempo de proceso y no ofrecen ganancia o mejora digna de contemplarse.

Con esto avanzado podemos determinar que de las 384 configuraciones que se resumen en los ya mencionados 96 grupos, 42 de ellos logran su máxima eficiencia sin aplicar centrado o reducción de vectores, de los 54 restantes 45 alcanzan su máxima eficiencia aplicando solo la reducción de los vectores y solo 9 alcanzan su máxima eficiencia sólo con el centrado de los vectores, pero ninguno alcanza su máxima eficiencia aplicando ambos procesos.

De lo anterior y dado que la ganancia en el mejor de los casos es marginal, representaremos a cada grupo por aquella configuración que logra maximizar su eficiencia y que minimiza el costo computacional, por lo que en adelante sólo tenderemos 96 datos, ya que cada dato representa a su mejor opción dentro de las 4 configuraciones ya cubiertas.

En este mismo tren de ideas veamos ahora que pasa con el tipo de escala, en la figura 4.27 vemos que al presentar las configuraciones de acuerdo a su calificación en orden creciente, la escala Mel resulta la ganadora seguida muy de cerca por la escala Bark, siendo sólo la 9ª mejor configuración aquella que emplea la escala lineal, mismo que soporta que las escalas psicoacústicas siguen siendo la mejor opción, dentro de éstas, pero no de manera contundente, la escala Mel es la que arroja los mejores resultados.

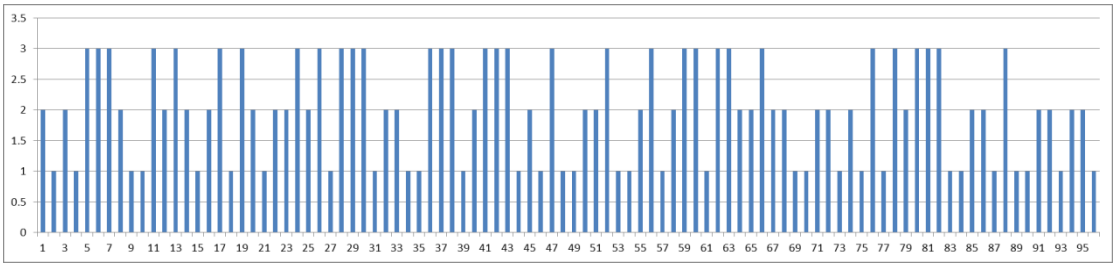


Figura 4.27: Efecto de la escala

De este resultado ahora descartaremos la escala Bark y la lineal, ahora sólo quedan 32 opciones de configuración posibles, para ver dentro de estas opciones el efecto del detector de actividad vocal o VAD por sus siglas en inglés, basta con observar la figura 4.28, de igual manera ordenado de menor a mayor calificación los mejores 4 resultados emplean el detector de voz, motivo por el cual lo consideraremos útil, esto en concordancia con las recomendaciones generales.

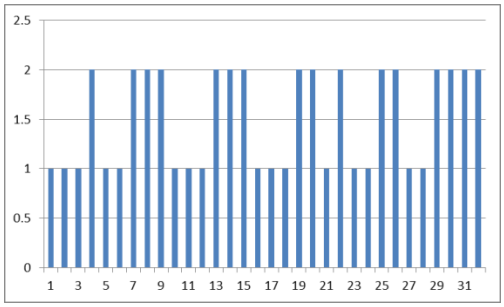


Figura 4.28: Efecto del VAD

Con este parámetro estudiado podemos pasar a la segunda transformación para obtener los coeficientes cepstrales, como se aprecia en la figura 4.29, las dos mejores configuraciones emplean la DCT por sus siglas en inglés o transformada coseno discreta, esto de nueva cuenta concuerda con las recomendaciones generales de que la transformación de Fourier completa presenta desventajas en cuanto a sus resultados.

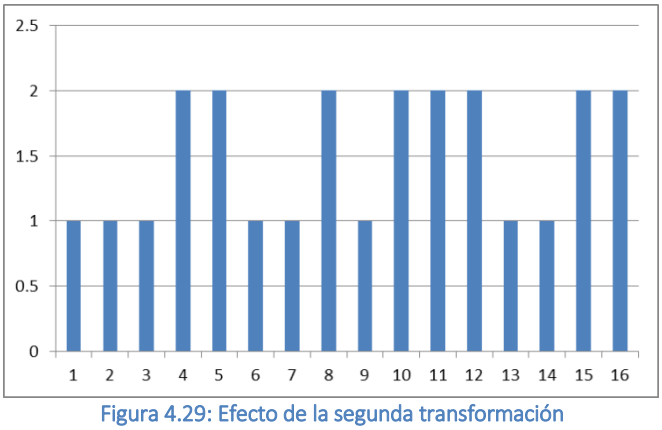


Figura 4.29: Efecto de la segunda transformación

Para la última parte dejamos la conformación del paquete que integra los coeficientes cepstrales, en la figura 4.30, apreciamos de nueva cuenta las calificaciones de menor a mayor, pero contrario a la recomendación general, la información dinámica no ayuda a obtener un mejor modelo, sin embargo, la energía si ayuda a mejorar un poco el desempeño de la solución como se puede ver entre el experimento 7 y el 8.

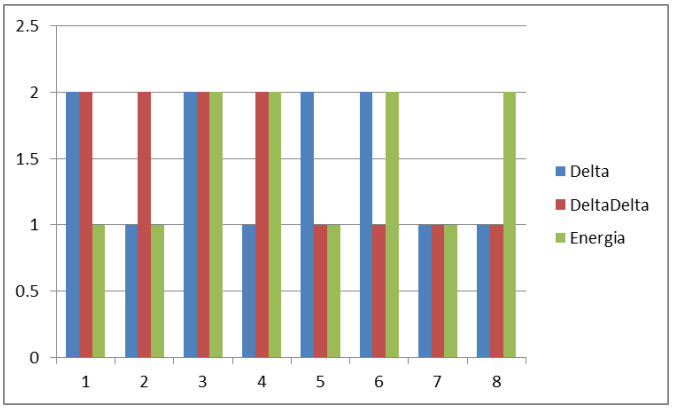


Figura 4.30: Conformación de los coeficientes

Con esto queda concluida la configuración del sistema, ahora sabemos de manera empírica cuál es la mejor configuración, lográndose maximizar la eficiencia de la solución, misma que será analizada un poco más a detalle a continuación.

5. Resultados

Este quinto presenta el análisis de resultados, la pregunta a responder es ¿Cuáles son los parámetros que maximizan la eficiencia del motor biométrico?

El presente trabajo se compone de dos partes principales, creación de un motor biométrico por voz y su consecuente "afinación" para lograr el mejor desempeño posible, en este sentido se dió un recorrido rápido por la historia de esta relativamente joven biometría, se estudió el estado del arte y se implementó un motor biométrico.

5.1 El motor biométrico

El motor fue desarrollado enteramente en MATLAB 2012a, está compuesto por una interface gráfica o GUI que facilita su uso y 23 módulos de código que lo implementan con más de 3,000 líneas de código.

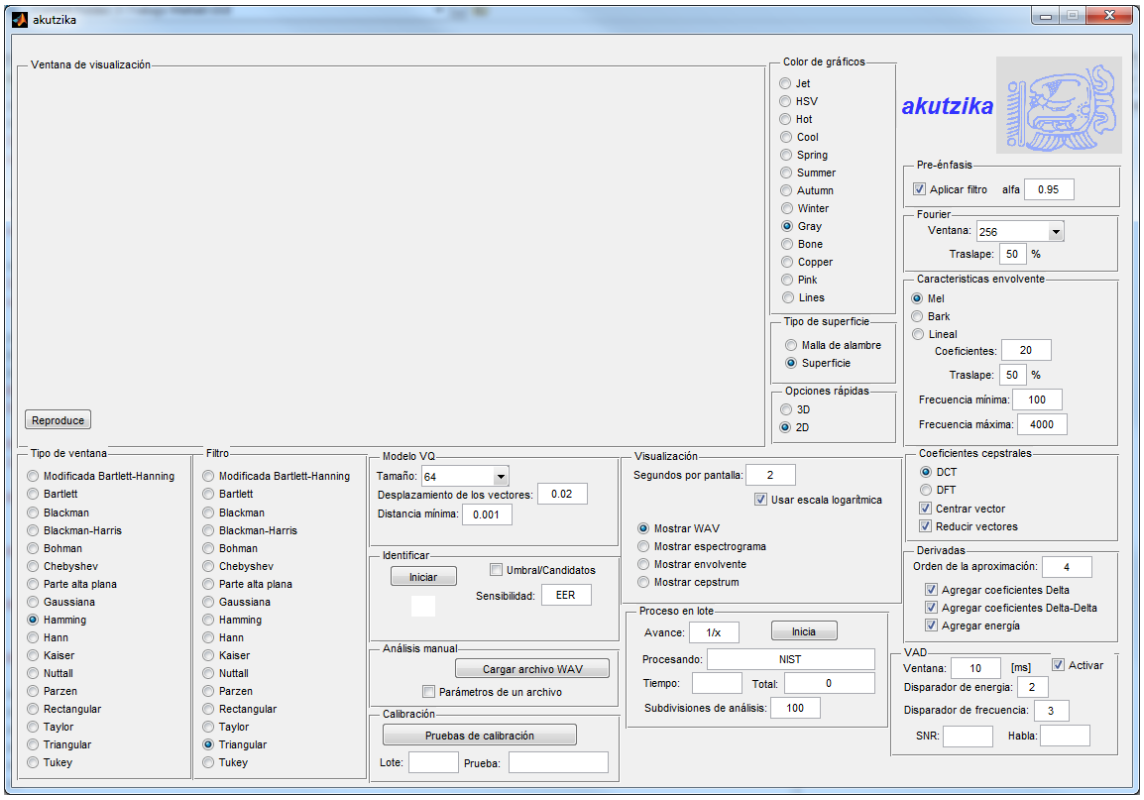


Figura 5.1: Interface gráfica de akutzika

En la figura 5.1 podemos ver la interface gráfica de la solución, ésta ofrece menús y controles que nos permiten "configurar" el motor biométrico y realizar las tareas de enrolamiento e identificación, aunado a un módulo de visualización de resultados.

De entrada puede parecer abrumador y desordenado, pero la idea básica fue tener toda la solución en una sola pantalla, en la figura 5.2 vemos la sección de pre-énfasis, aquí podemos decidir aplicar o no aplicar este procesamiento, si lo aplicamos podemos seleccionar un valor de alfa entre 0 y 1, es pertinente mencionar que este campo en particular y algunos otros si vigilan que los valores que se introducen sean válidos, sin embargo, en un punto determinado como el número de coeficientes cepstrales a calcular no existe una regla firme que diga mínimo X,

máximo Y coeficientes, motivo por el cual dejé de controlar la validez de los campos de texto, después de todo la finalidad no es lograr un sistema comercial, sino crear un motor biométrico sumamente flexible con el cual poder experimentar.

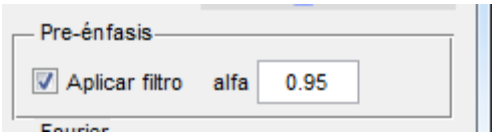


Figura 5.2: Sección de pre-énfasis

A continuación veremos en la figura 5.3 la ventana para la transformada de Fourier, ésta nos permite seleccionar el tamaño de la ventana en número de muestras que será lo que defina si se trata de un espectro de banda ancha o de banda estrecha, así como el traslape de las ventanas en un porcentaje abierto de cero a 99 %, por supuesto que un traslape del 100% no es viable ya que nunca avanzaríamos, pero este parámetro va acompañado de otro más que por cuestiones de espacio se colocó por separado.

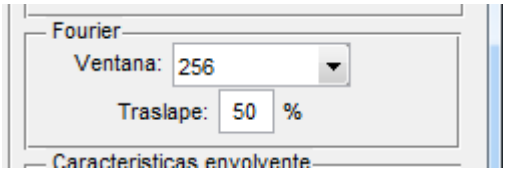


Figura 5.3: Sección de ventana para el cálculo de la FFT

Aquí tenemos la parte complementaria, el tipo de ventana como ya fue comentado antes, podemos elegir cualquiera de los 17 tipos de ventana que se ven en la figura 5.4.

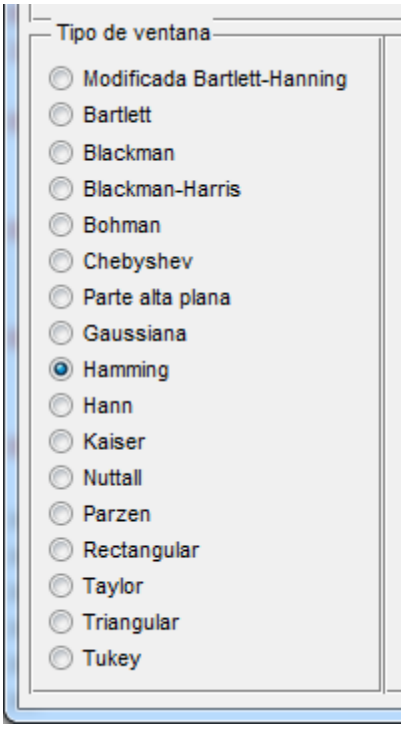


Figura 5.4: Tipo de ventana matemática

A continuación veremos las características de la envolvente que nos dará los coeficientes cepstrales, en la figura 5.5 vemos que tenemos la opción de escoger la escala que emplearemos, por lo que estrictamente no trabajamos sólo con MFCC, además podemos

seleccionar el número de coeficientes deseados, el traslape de los filtros que conformarán el banco y una frecuencia mínima y máxima para el filtro paso banda que podemos aplicar antes de este proceso.

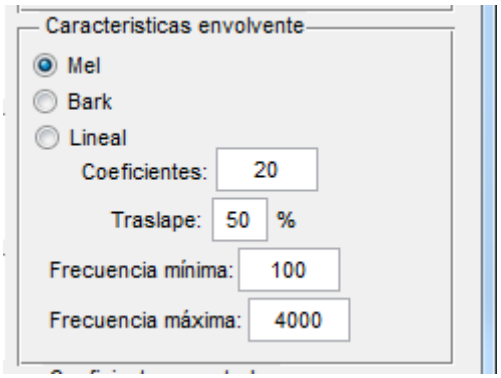


Figura 5.5: Parámetros de la envolvente

En esta sección de nueva cuenta podemos elegir de los 17 tipos de filtro disponibles, cuál será el que emplearemos como lo muestra la figura 5.6.

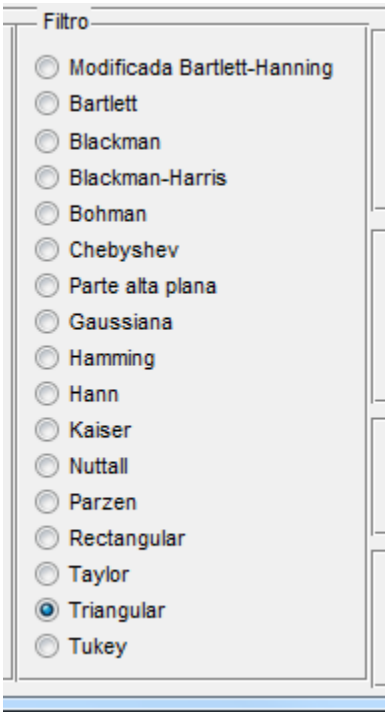


Figura 5.6: Tipo de filtros

Como siguiente parámetro podemos elegir si deseamos calcular los coeficientes cepstrales con una transformada coseno discreta (DCT) o con una transformada discreta de Fourier (DFT), los otros dos puntos no están estrictamente relacionados con este cálculo, sin embargo, coloque aquí dos procesamientos opcionales que son el centrado y reducción de los vectores obtenidos después de la segunda transformación.

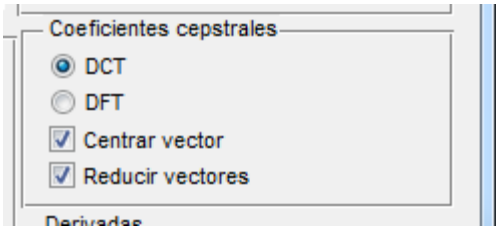


Figura 5.7: Coeficientes cepstrales

En seguida podremos elegir también el orden de la aproximación polinomial mediante la cual se calculará la información dinámica de cómo cambia el habla, mejor conocido como coeficientes delta y coeficientes delta-delta o coeficientes Δ y Δ Δ por su simbología, también podemos seleccionar si deseamos agregar al vector final esta información dinámica y la energía de la señal por cada ventana.

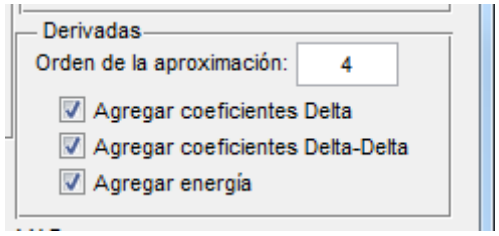


Figura 5.8: Informacion dinámica

Como un módulo opcional y agregado prácticamente de última hora podemos elegir los parámetros de un detector de actividad vocal simple o VAD por sus siglas en inglés, en la figura 5.9 observamos que podemos seleccionar el tamaño de la ventana, el umbral con el que trabajaran los dos métodos complementarios (energía y frecuencia máxima) y su activación o no para aplicar este procesamiento.

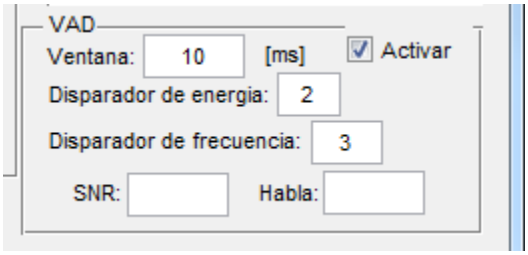


Figura 5.9: Detector de actividad vocal

De manera complementaria y sólo cuando se trabaja un archivo de manera manual, este módulo nos entrega la relación señal a ruido del archivo procesado y el número de segundos que el detector considero como “habla”.

Por último, dentro de los parámetros del motor tenemos el modelado del vector de parámetros, en la figura 5.10 vemos que podemos seleccionar el tamaño o número de palabras código, el desplazamiento que será utilizado para dividir las palabras código y la distancia mínima que deseamos para aceptar la convergencia del método reiterativo, este último parámetro sólo es empleado si utilizamos la primera versión del modelado por cuantización vectorial, ya que para su segunda versión que tiene velocidades de convergencia más rápidas este parámetro fue eliminado, dejando que el programa lo calcule de manera automática.

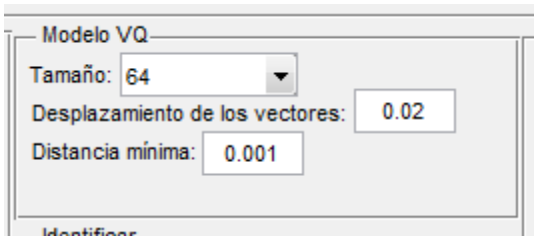


Figura 5.10: Modelado por cuantización vectorial

Ahora veremos qué podemos hacer con el sistema; como su primera funcionalidad y la que presenta más opciones, vemos el análisis manual de un archivo con voz, en la figura 5.11 se muestra esta opción, de entrada es muy simple, un sólo botón para procesar el archivo y una opción para cargar la configuración del motor desde un archivo.

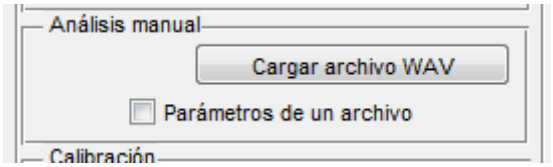


Figura 5.11: Análisis manual

Los archivos que la solución "entiende" son generados mediante la función de MATLAB combina.m que se encuentra en el anexo 1, esta función nos permite crear de manera manual archivos de configuración con una o tantas configuraciones como quisiéramos probar, proporcionando uno o más archivos con las configuraciones deseadas.

El número de combinaciones y el tiempo aproximado de cómputo teniendo como base mil archivos de voz y 10 segundos de procesamiento por archivo (promedio obtenido de forma empírica con una computadora con procesador i7 2677M de Intel y 4 GB de RAM en un sistema operativo Windows 8 Pro), más tarde veremos para que sirve tener más de una configuración por archivo.

Ahora, ya sea que se obtenga la configuración de la forma o de un archivo, esto disparará que la aplicación procese el archivo, proporcionando toda la información que hemos mencionado, así como algunos gráficos que auxilian a corroborar el funcionamiento y resultados obtenidos.

En la figura 5.12 podemos observar que en la ventana de visualización se muestra la gráfica de la forma de onda (visualización por default) y debajo de ella un botón para reproducir el audio ya procesado por el VAD (si se utilizó la opción), también observamos que en el recuadro del VAD se puede apreciar el SNR del archivo y los segundos de habla "pura" que se trabajaron por el motor biométrico.

Así mismo, se podrán utilizar nuevas opciones que nos permiten manejar la representación gráfica de la señal, en la figura 5.13 se muestran los controles de visualización, podemos seleccionar si se utiliza una escala logarítmica para mostrar todos los modos de visualización excepto por la forma de onda, y podremos seleccionar el número de segundos a mostrar por cada pantalla, el default es de 2 segundos, por último las visualizaciones disponibles son la forma de onda, el espectro, la envolvente y el cepstrum, en la figura 5.15 se muestran algunos ejemplos de visualizaciones alternas, cabe aclarar que el botón de "Refresca" que aplica las nuevas opciones de dibujo sólo estará visible cuando hay gráficos disponibles.

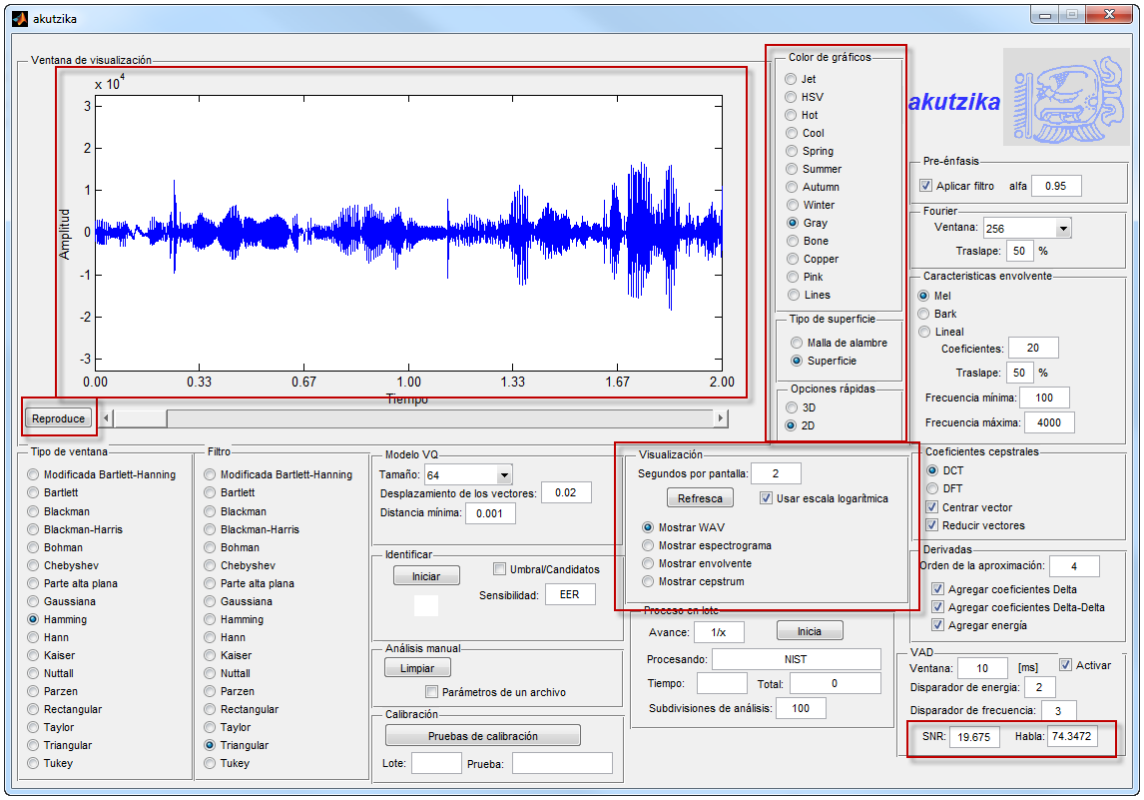


Figura 5.12: Resultado en análisis manual

En la figura 5.14 se muestran de hecho tres menús, pero todos están relacionados, el primero nos deja seleccionar el color de los gráficos a mostrar, el segundo si la visualización será en una superficie tipo malla o sólida, y la última opción nos deja seleccionar una visualización en forma plana o tridimensional (estas tres opciones no aplican para la gráfica de forma de onda).

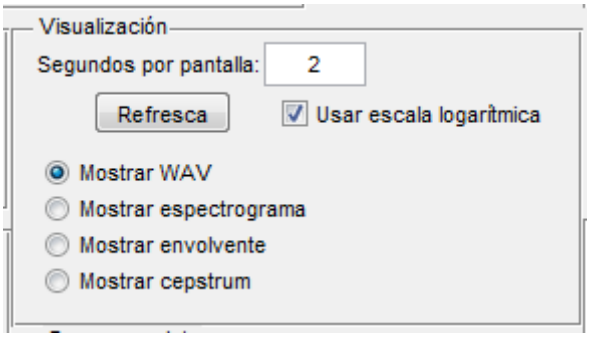


Figura 5.13: Opciones de visualización

Es pertinente mencionar que durante el análisis manual el botón de “Cargar archivo WAV” desaparece y aparece un botón “Limpiar”, mismo que removerá las gráficas y dejará la forma lista para procesar otro archivo, también es conveniente mencionar que se generará un archivo con el mismo nombre del WAV pero con extensión “mat”, éste es un archivo de MATLAB que contiene el modelo del locutor.

Ahora pasaremos a los análisis por lotes, para esto existen dos modalidades que van de la mano pero son ligeramente diferentes y al mismo tiempo complementarias, en la figura 5.16 se aprecian los menús de “Proceso en lote” y “Calibración”, de hecho el segundo menú utiliza las funciones del primero, así que explicaremos ambos y como se combinan.

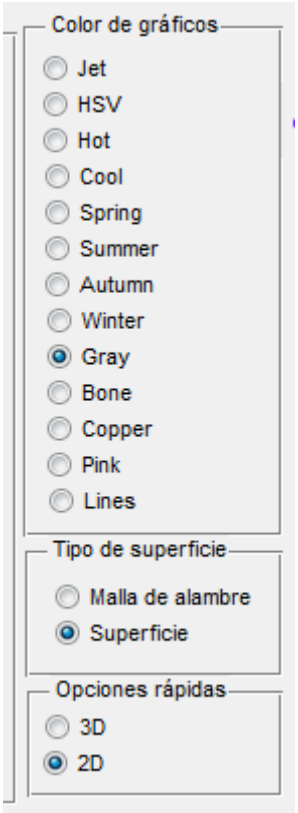


Figura 5.14: Otras opciones gráficas

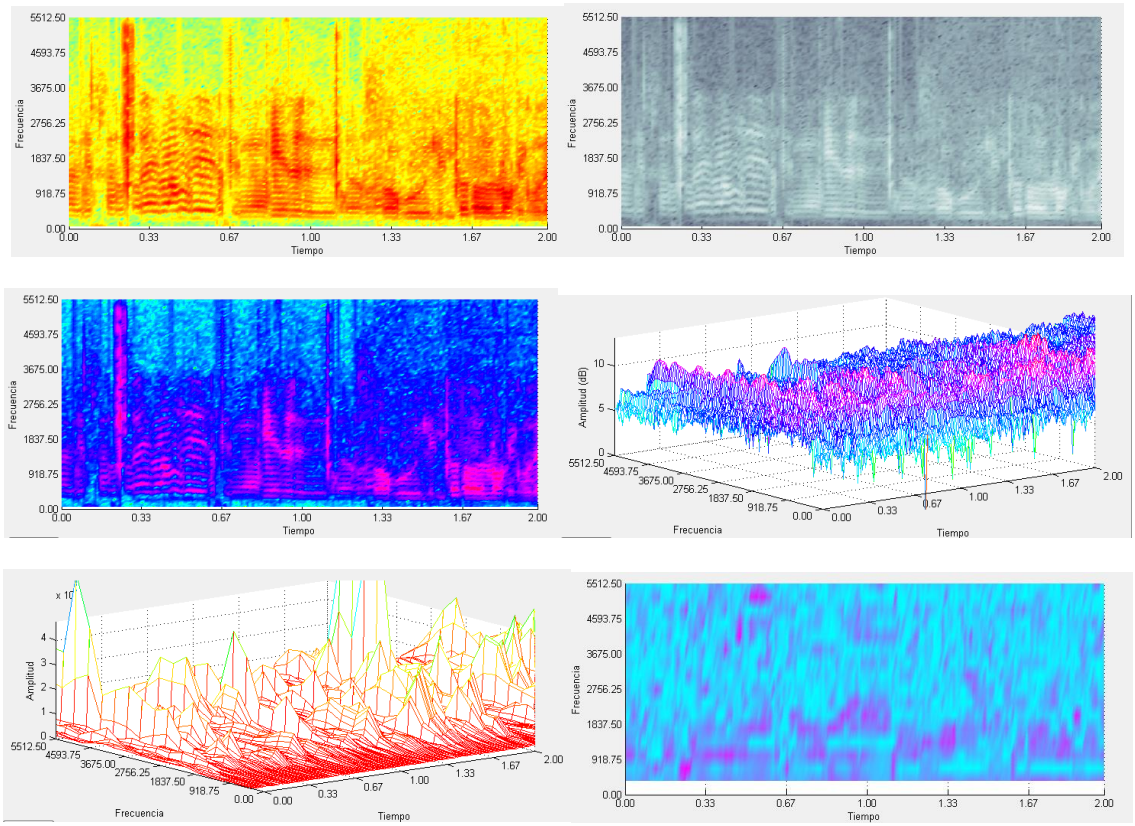


Figura 5.15: Ejemplos de visualización

En el menú de procesamiento por lotes tenemos un botón de iniciar, un campo para texto que dice “Procesando” y como valor por default dice “NIST”, así como un último campo para las

subdivisiones de análisis, el valor de NIST es simplemente el nombre de la carpeta donde espera encontrar los archivos a procesar, como yo utilice audios del NIST para mis pruebas puse este nombre por default, pero puede cambiarse a lo que uno guste, y el campo de las subdivisiones se refiere a cuantas divisiones se crearán para la escala de resultados.

Explicando un poco más este último término, digamos que la distancia mínima es de 1.1 y la máxima de 3.6, en este tenor si dejamos el valor de 100 esto indicara que creará 100 regiones entre estos valores a una distancia de 0.025, esto se usa para crear "regiones" ya que las distancias nunca caerán en valores cerrados como sería una escala 1, 2, 3… n.

Cada subdivisión será un punto de nuestra escala o regleta, sólo cabe hacer mención que el número de archivos procesados debe ser mayor al número de divisiones o se generaran errores de interpretación. Como ya se explicó antes, la forma en que un sistema biométrico califica son tablas y estas tablas serán generadas a partir de estos procesamientos en lotes, esta es la única manera de generar las tablas que permiten "calificar" una identificación.

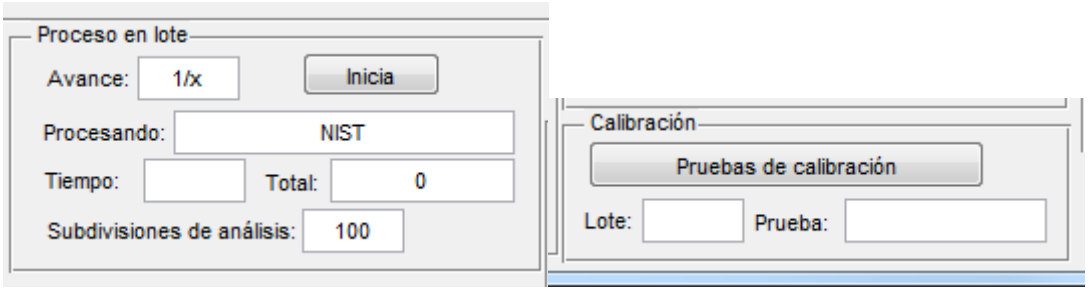


Figura 5.16: Procesamiento por lotes

Con esto dicho basta con aclarar que se tomaran los parámetros de la forma y se procesaran los X archivos de audio que se proporcionan en un archivo TXT (contiene los nombres de los archivos WAV en texto plano), los resultados se grabaran en una carpeta fija llamada "Modelos" que contendrá los X modelos matemáticos de los locutores y dos archivos adicionales, uno llamado "Resultados_Y.mat" donde la Y es el nombre del archivo TXT proporcionado, y el otro "Tabla_Y.txt" donde de nuevo Y es el nombre del TXT, éste último es una tabla en texto plano que contiene los resultados de comparar todos los X audios contra sí mismos, por lo tanto es una tabla de dimensión X por X, por ejemplo, si tomamos 1,000 audios esto generará una tabla con 1,000,000 de comparativos.

El archivo de MATLAB contiene la calificación obtenida por el sistema, ésta se conforma del inverso del EER sumado con los porcentajes de aciertos y descartes certeros (una calificación de 3 sería un sistema que jamás se equivoca y uno con cero uno que siempre se equivoca), se proporcionan los datos de las distancias intra-locutor y entre-locutores, un juego simple de medidas estadísticas para cada una de las mencionadas tablas, una tabla con los parámetros de trabajo usados en la prueba, la tabla de resultados para las "100" regiones, misma con la que se generan los resultados de un comparativo y las gráficas de eficiencia, el ERR del sistema, el porcentaje de aciertos y de rechazos correctos y el disparador o umbral ideal para dicha configuración.

Cabe hacer mención que para distinguir cuando se trata de un mismo locutor y cuando de uno distinto se usó la misma nomenclatura del NIST, el nombre inicia con una "F" para mujeres y

con una “M” para hombres, después cuatro dígitos que lo identifican seguidos de un número entre paréntesis que representa el número de grabación para los casos en donde hay más de una grabación por cada locutor, por ejemplo “M1024(1).wav”, “M1024(2).wav” y “F0012(1).wav”, aquí dos grabaciones son del mismo locutor varón y la ultima de uno distinto femenino, este juego de grabaciones donde a priori se sabe si se trata o no del mismo locutor son conocidos como corpus de calibración, si no se respeta la nomenclatura el sistema no podría distinguir cuando se trata de un mismo locutor o de uno distinto.

Los campos de “Tiempo” y “Total” muestran el tiempo de proceso de un archivo y el total acumulado en el lote, estos son datos meramente para analizar el comportamiento del sistema.

Ahora para el menú de calibración, éste hace uso cabal del procesamiento en lote, sólo que en lugar de tomar los parámetros de la forma los tomara de un archivo que puede tener uno o más juegos de parámetros generados con la función “combina.m”, el resto del comportamiento es idéntico a lo ya descrito excepto porque en los campos de lote y de prueba se mostrará el número de prueba que se procesa, por ejemplo “2/15” o prueba 2 de 15 pruebas y el nombre del archivo que contiene dichas configuraciones.

Se podría decir que éste es un proceso de lotes por lotes, es utilizado para obtener los resultados ya mostrados de cómo se comporta el sistema, para evitar que se sobrescriban modelos y resultados se creará una carpeta dentro de “Modelos” con el número de la prueba, aunado a los resultados ya descritos se generara un archivo llamado “ResultadoFinal.mat” que no es sino una tabla con las X calificaciones de las X pruebas para su rápido análisis.

Por último, tenemos la sección que nos permite identificar a una persona contra una X cantidad de personas, esto requiere tener un resultado previo con algún corpus de calibración, esto dado que requeriremos un archivo de resultados que contenga una tabla de distancias y que será empleada para calificar la identificación del locutor, esto se puede obtener con cualquiera de las dos opciones de procesamiento por lotes ya explicadas.

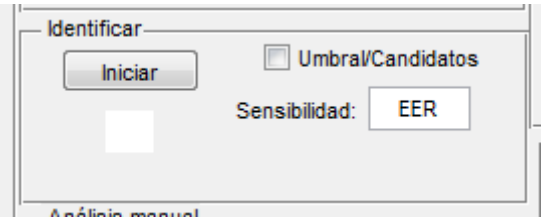


Figura 5.17: Menú de identificación

Lanzar la identificación es simple, en la figura 5.17 se ve que se tiene el botón de Iniciar, éste nos permite seleccionar el archivo con los resultados, es importante mencionar que se puede trabajar sobre el umbral de la calificación (las regiones) o sobre el número de candidatos que deseamos ver, de entrada el sistema nos colocara en el EER o lo que es lo mismo en el punto óptimo, pero lo podemos hacer más flexible o menos permisible, esto ya depende de nosotros.

Una vez que tenemos el archivo seleccionado se modificará ligeramente nuestro menú, en la figura 5.18 vemos que se cambia el botón a Continua y se pone una barra deslizable que nos permite modificar el número de candidatos o el umbral de sensibilidad, dependiendo de nuestra selección.

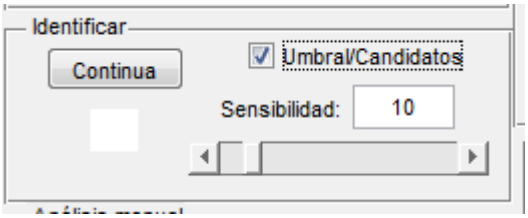


Figura 5.18: Continua el menú de identificación

Una vez que presionamos el botón de continuar nos pedirá localizar el modelo de la persona a identificar y el directorio que contiene toda nuestra "base de datos" de locutores, con esto bastara para realizar la identificación, en la figura 5.19 se pueden ver los resultados obtenidos.

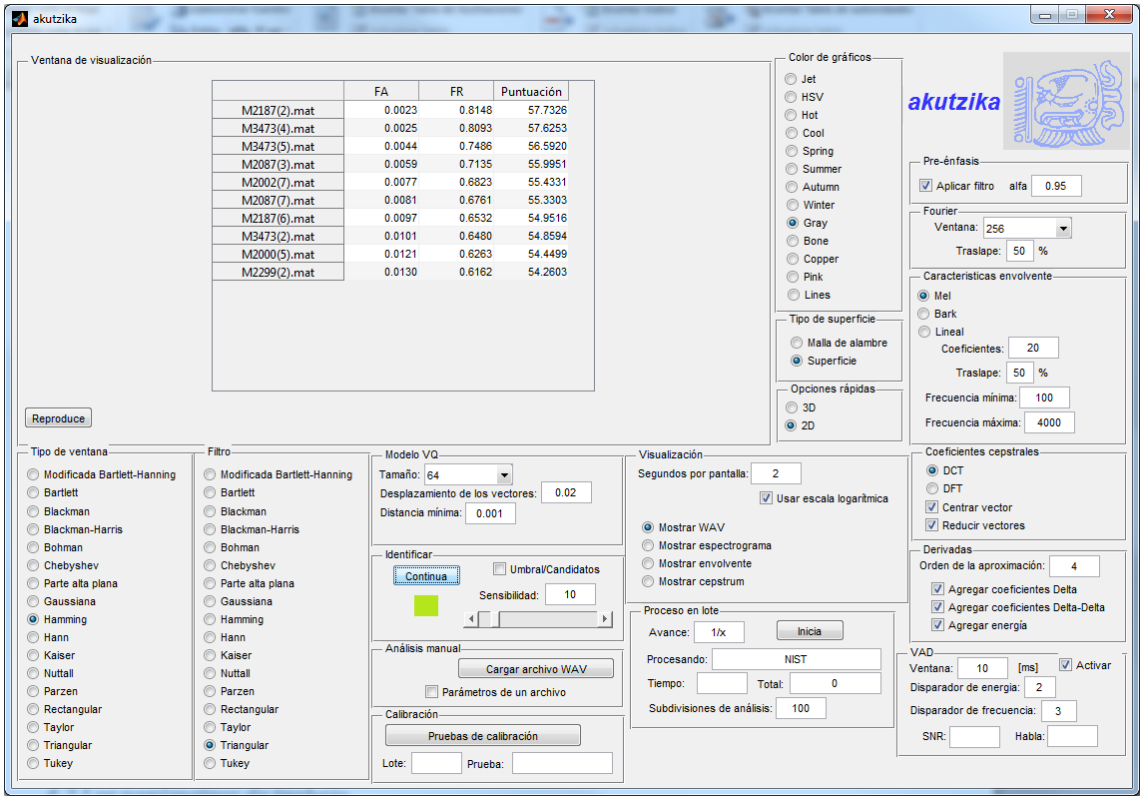


Figura 5.19: Resultados de la identificación

Como vemos aparece un pequeño recuadro verde indicando que si hay candidatos, si no tuviéramos candidatos sería rojo, en la sección de visualización veremos una tabla con candidatos de acuerdo a la opción seleccionada, en este caso se probó con una identificación con los mejores 10 candidatos, se buscó al locutor M2187(3) y cómo podemos ver el primer candidato es el M2187(2) y en el séptimo lugar tenemos el candidato M2187(6), de la explicación de la nomenclatura de NIST sabemos que éstas son distintas muestras de un mismo locutor.

5.2 Los parámetros de trabajo

Se realizaron un total de 747 pruebas, cada una de ellas compuesta de un grupo de 1,000 grabaciones distintas con texto independiente, idioma independiente y 174 locutores distintos, esto representa lo siguiente:

- Más de 180 días de procesamiento

- 747 mil modelos biométricos
- 747 millones de comparaciones biométricas

De esto podemos ver que la presente tesis es uno de los trabajos más largos y completos en cuanto a biometría de voz, su programación llevó a una interface gráfica simple y versátil, permitiendo dedicar el debido tiempo a las pruebas de calibración y parametrización para afinar el motor biométrico, logrando así su máxima eficiencia.

Para ejemplificar esto mencionaremos tres puntos de operación, la mejor configuración de acuerdo a las pruebas ya descritas, una prueba con los valores más comúnmente sugeridos y una última prueba con los parámetros que hacen al sistema lo menos eficiente posible, estos resultados se muestran a continuación:

- La máxima eficiencia lograda nos lleva a un EER de 14.898%
- Mediante los parámetros “recomendados” se obtiene un EER de 23.720%
- La eficiencia más baja del mismo motor mal configurado logra un EER de 39.373%

De estos puntos podemos ver que la mejora en eficiencia del peor caso al mejor es de un 24.475%, mismo que demuestra la importancia de parametrizar correctamente un motor biométrico, pero más importante aún, si utilizamos los parámetros que se sugieren a nivel general, en los diversos artículos y libros que se consultaron todavía nos quedamos un 8.822% por debajo del nivel de eficiencia logrado en la presente tesis.

De los resultados mencionados se puede apreciar que la idea de un sistema modular y configurable fue la aproximación correcta, este sistema permitió a través del presente trabajo probar y experimentar con distintos componentes, buscando no sólo el corroborar su utilidad para la tarea de identificar a un locutor, sino afinar de manera adecuada cada módulo para que dé lo mejor de sí mismo.

Así mismo, se puede vislumbrar que el probar distintas técnicas y métodos tiene sus frutos, la identificación de locutores no es simple, es tanta la información que lleva el habla, que está muy enmascarada la información del locutor, motivo por el cual queda claro que a nivel de investigación la mejor técnica es el desarrollo rápido mediante ambientes que permitan la rápida generación de prototipos y la prueba de los mismos, ya que en caso contrario en la aproximación usual se puede pasar un muy largo tiempo desarrollando una herramienta que a fin de cuentas podría no dar los mejores resultados.

Otro punto que queda claro respecto a los resultados obtenidos es lo siguiente, lograr buenos resultados cuando la población de referencia es pequeña es relativamente simple, pero al crecer la población o encontrar circunstancias poco favorables como el cruce de canal o el ruido, la eficiencia de los motores disminuye significativamente.

El último punto que podemos destacar de esta experiencia es que el refinamiento final entre un sistema que comete dos errores por cada cien intentos y uno que comete quince errores por cada cien intentos es enorme, es relativamente fácil lograr un motor con una tasa de error igual del 20% (EER), pero disminuirlo al 10% ya representa mucho mayor dificultad, sin mencionar que lograr una tasa del 2% es ya una meta que solo muy pocas universidades o empresas a nivel mundial podrían aspirar a lograr.

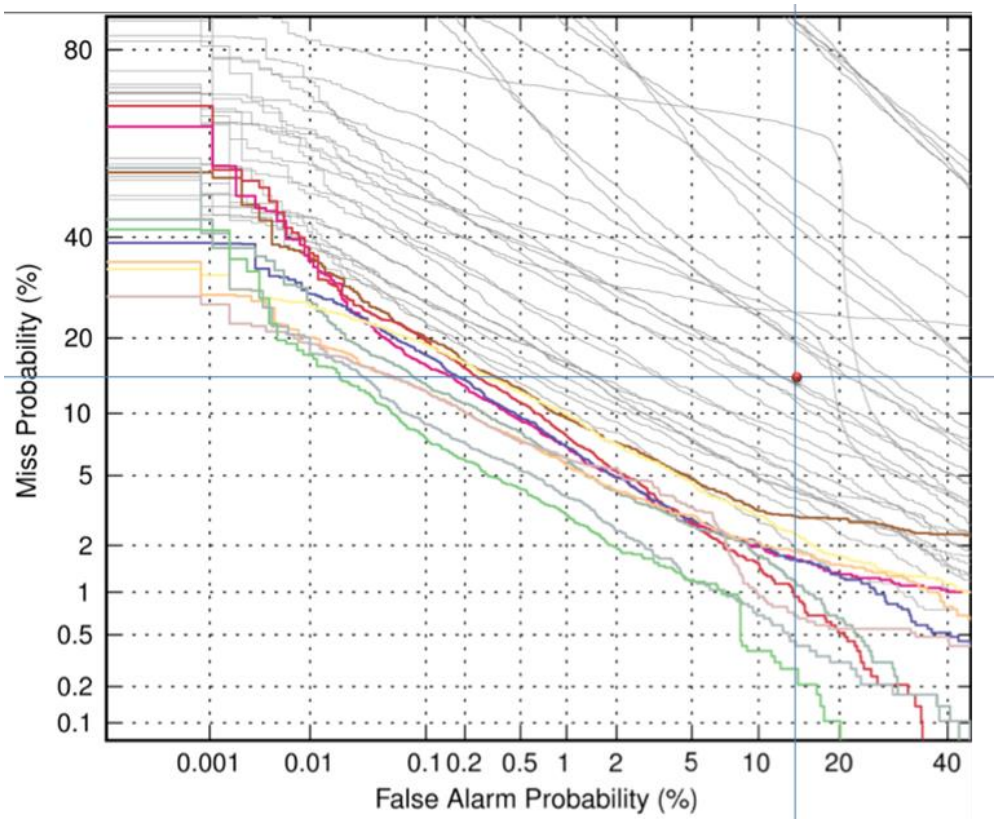


Figura 5.20: Curva DET del NIST 2012

En la figura 5.20 podemos observar una modalidad ligeramente diferente de la gráfica DET (Detection Error Tradeoff), en ésta podemos ver los resultados de la evaluación NIST 2012 sobre identificación automatizada de locutores, el mejor sistema presentado en esta competencia internacional logró una tasa de error igual de aproximadamente un 2% y el último de los mejores 10 sistemas a nivel mundial logro un EER de aproximadamente el 5%.

De esta gráfica podemos ver que aunque un EER del 14.898% no nos coloca en ningún lugar cercano a las 10 mejores soluciones, nos pone por encima de muchos participantes en condiciones de una entrevista microfónica o telefónica con posible ruido moderado añadido, esto resulta definitivamente inspirador, ya que en esta prueba en específico participaron 75 universidades y empresas de todo el mundo, y de ningún modo quedaríamos fuera de la gráfica de resultados, siendo el punto rojo el EER de la solución desarrollada.

6. Conclusiones

En general podemos decir que los resultados obtenidos superan lo esperado, ***akutzika 2.0*** logró un imaginario lugar 26 en las pruebas NIST 2012, lo cual representa un resultado inspirador, mi programa "Acústica Forense 1.0" que formó parte de mi tesis de licenciatura fue una aproximación aceptable para su tiempo, pero pertenecía a la vieja escuela de los sistemas dependientes del texto, esta nueva aproximación es independiente del texto y del idioma, motivo que lo hace un sistema que trabaja bajo condiciones más reales.

En general se comprobó que el uso de un ambiente de generación de prototipos como MATLAB es lo más adecuado para este tipo de desarrollos, este ambiente presenta retos y limitantes propias, pero en general ofrece bondades que lo hacen ideal; desarrollo a una velocidad acelerada, enfoque a los aspectos importantes sin desviar la atención a funciones triviales, amplia documentación y ejemplos; la solución se puede probar y mejorar a medida de las necesidades en un ciclo de desarrollo más rápido que en cualquier otro lenguaje de programación.

Con esta experiencia ganada y un motor biométrico totalmente funcional nos queda la duda ¿Cómo mejorar?

6.1 Sugerencias de mejora

La respuesta a la anterior pregunta es mediante la implementación de la calidad total. ¿Cómo llegar a la meta de un sistema que compita con los mejores? Con mejoras, mejoras y aún más mejoras…

El término calidad total es el estado más evolucionado dentro de las sucesivas transformaciones que ha sufrido el término calidad a lo largo del tiempo, primero se habló del control de la calidad, después nace el aseguramiento de la calidad y finalmente se llega al concepto calidad total que implementa básicamente la mejora continua y que por supuesto engloba también la garantía en la continuidad del nivel alcanzado y la inspección del producto o sistema.

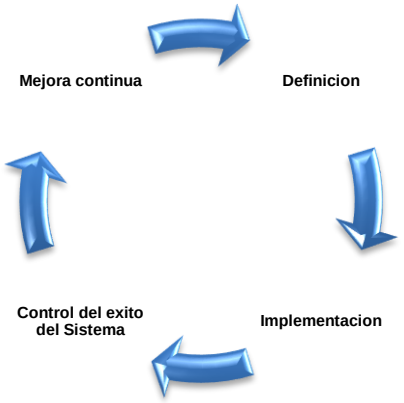


Figura 6.1: Ciclo de mejora continua

La implementación es simple, se tiene que definir una meta y un cómo lograrla, que fue lo que se cubrió en los capítulos dos y tres, se continúa con la implementación de un primer sistema base, en nuestro caso el sistema ***akutzika 2.0*** que ya es capaz de reconocer al parlante, control del éxito mediante evaluaciones objetivas del desempeño, que fue la aportación principal de

esta tesis, se corrieron cientos de miles de experimentos para probar su eficiencia, finalmente restando el análisis de áreas de oportunidad para su mejora; la mejora continua en base a la experiencia y la repetición del ciclo hasta lograr la metas deseadas es el camino a la mejora.

No se puede mejorar lo que no existe, como Ricardo Hirata lo plantea, "lo que no se mide, no se controla y no se puede mejorar", no es posible generar un sistema de la nada que logre mejores resultados que otras universidades o empresas a nivel mundial, se debe iniciar por crear una base sólida de conocimiento y experiencia, con esto se deben plantear metas ambiciosas pero reales, se debe investigar, se debe participar en la comunidad científica internacional y se debe innovar para poderse colocar en un mejor posicionamiento en evaluaciones tecnológicas como las que realiza el NIST.

6.2 Trabajo pendiente

Contando ya con un sistema funcional, el trabajo que se deja pendiente es el robustecimiento del mismo, ahora sabemos que un sistema de este tipo tiene tres capas básicas, lo que podemos hacer antes de trabajar la señal (pre-proceso), la extracción de la información relevante (extracción de características) y la creación de un objeto comparable a partir de dicha información (modelado), en este tenor se pueden seguir creando nuevos módulos o mejorando los ya existentes para evaluar cómo se modifica el desempeño.

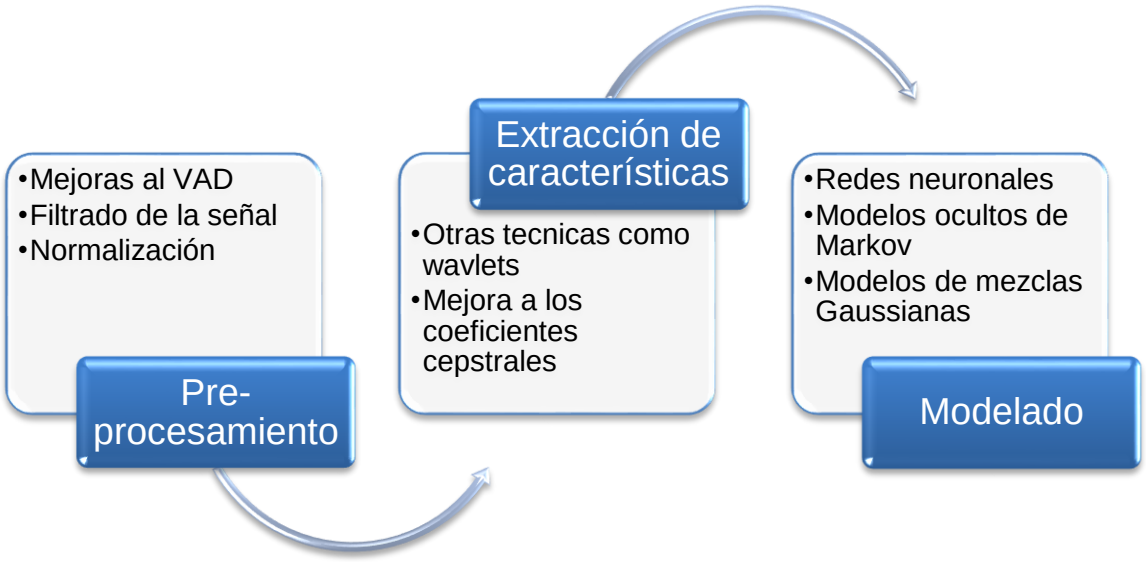


Figura 6.2: Módulos principales

El cómo lograrlo no es simple, no hay un camino claro o seguro, alcanzar a los punteros puede significar cambios radicales, algunas de las técnicas más modernas como i-vectors, los clasificadores de análisis lineal probabilístico discriminativo, el modelado por covarianzas entre otras técnicas podrían ayudar, pero están poco documentadas por su relativa novedad en la identificación del parlante.

6.3 Trabajo a futuro

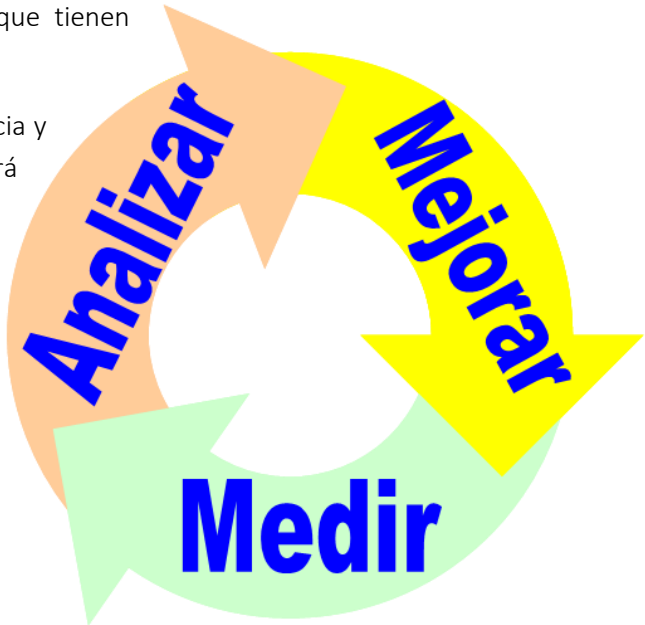
El trabajo que se debe realizar a futuro es implementar tantos métodos y técnicas como nos sea posible, esto dado que la innovación no siempre estará en el mismo lugar, la próxima idea original que impulse a la identificación de locutor puede radicar en una idea simple que auxilie a resolver algún problema vigente.

Por citar un ejemplo de este tipo de innovaciones se pueden mencionar a los coeficientes PLP (Perceptual Linear Prediction), ésta es una metodología que nace a partir de mejorar los coeficientes MFCC y LPC con una escala psicoacústica similar a Mel o a Bark, mismos que han logrado buenos resultados, lo mismo puede decirse de técnicas ahora tan famosas como el modelado GMM (Gaussian Mixture Model), que no es sino una refinación de las cadenas ocultas de Markov de un sólo estado.

La implementación de tantas técnicas como nos sea posible nos permitirá experimentar con estas metodologías, para mejorar algo se le debe conocer, no podemos innovar lo que no conocemos, la innovación puede estar en la conjunción de dos o más metodologías para crear una nueva técnica, puede estar en la mejora o robustecimiento de algún método basado en la experimentación o puede ser una idea radical que supere o corrija lo que se usa actualmente.

Como final de estas conclusiones podemos decir que la tarea es compleja pero digna de continuarse, el interés en esta materia se denota por el incremento de participantes en las pruebas del NIST que se realizan cada 2 años, un punto interesante es que la tasa de error en estos sistemas tiende a bajar en un 50% cada 3 evaluaciones (seis años), pero en general los grupos que presentan saltos significativos son aquellos que tienen experiencia en estas tecnologías.

Como comentario de cierre se puede decir que la experiencia y la mejora continua son factores clave, difícilmente se podrá crear un sistema competitivo sin tener una base de conocimientos y un sistema funcional que permita la experimentación y el robustecimiento en un ciclo de mejora continua.



BIBLIOGRAFÍA

1) PHILIP ROSE, (2002) "*Forensic Speaker Identification"*. Nueva York, Taylor & Francis Inc. Capítulo II.
2) MARK PITTS, (2008) "*Writing in Maya Glyphs: Names, Places, & Simple Sentences*". El Proyecto de Ayuda y Educación
3) DANIEL T. LAROSE, (2006) "*Data mining methods and models*". Nueva Jersey, John Wiley & Sons Inc.
4) M. Deza, E. Deza, (2006) "*Dictionary of Distances*". Elsevier.
5) S. PRUZANSKY, "Pattern-matching procedure for automatic talker recognition," J.A.S.A., 35, pp. 354-358, 1963.
6) S. PRUZANSKY y M. V. MATHEWS, "Talker recognition procedure based on analysis of variance," J.A.S.A., 36, pp. 2041-2047, 1964.
7) W. Endress, et. al., "Voice spectrograms as a function of age," Voice Disguise and Voice Imitation, J.A.S.A., 49, 6(2), pp. 1842-1848, 1971.
8) S. Furui, "An analysis of long-term variation of feature parameters of speech and its application to talker recognition," Electronics and Communications in Japan, 57-A, pp. 34-41, 1974.
9) P. D. Bricker y otros, "Statistical techniques for talker identification," B.S.T.J., 50, pp. 1427-1454, 1971.
10) K. P. Li and G. W. Hughes, "Talker differences as they appear in correlation matrices of continuous speech spectra," J.A.S.A., 55, pp. 833-837, 1974.
11) B. Beek, et al., "An assessment of the technology of automatic speech recognition for military applications," IEEE Trans. Acoustics, Speech, Signal Processing, ASSP-25, pp. 310-322, 1977.
12) M. R. Sambur, "Speaker recognition and verification using linear prediction analysis," Tesis doctoral, M.I.T., 1972.
13) A. E. Rosenberg y F. K. Soong, "Evaluation of a vector quantization talker recognition system in text independent and text dependent models," Computer Speech and Language 22, pp. 143-157. 1987.
14) F. K. Soong y otros, "A vector quantization approach to speaker recognition," AT&T Technical Journal, 66, pp. 14-26. 1987.
15) M. Sugiyama, "Segment based text independent speaker recognition," Proc. Acoust., Spring Meeting of Soc. Japan, pp. 75-76, 1988.
16) B.-H. Juang and F. K. Soong, "Speaker recognition based on source coding approaches," Proc. ICASSP, 1, pp. 613-616, 1990.
17) A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," Proc. ICASSP, 2, pp. 1291-1294, 1982.
18) N. Tishby, "On the application of mixture AR hidden Markov models to text independent speaker recognition," IEEE Trans. Acoust., Speech, Signal Processing, ASSP-30, 3, pp. 563-570, 1991.
19) R. Rose and R. A. Reynolds, "Text independent speaker identification using automatic acoustic segmentation," Proc. ICASSP, pp. 293-296, 1990.

20) T. Matsui and S. Furui, “Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMMs,” Proc. ICSLP, pp. II-157-160, 1992.
21) T. Matsui and S. Furui, “Concatenated phoneme models for text-variable speaker recognition,” Proc. ICASSP, pp. II-391-394, 1993.
22) A. Higgins, et al., “Speaker verification using randomized phrase prompting,” Digital Signal Processing, 1, pp. 89-106, 1991.
23) T. Matsui and S. Furui, “Similarity normalization method for speaker verification based on a posteriori probability,” Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, pp. 59-62, 1994.
24) D. Reynolds, “Speaker identification and verification using Gaussian mixture speaker models,” Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, pp.27-30, 1994.
25) H. Gish, M. Siu y R. Rohlicek, “Segregation of speakers for speech recognition and speaker identification,” Proc. ICASSP, S13.11, pp. 873-876, 1991.
26) M. Siu y otros, “An unsupervised, sequential learning algorithm for the segmentation of speech waveforms with multiple speakers,” Proc. ICASSP, pp. I-189-192, 1992.
27) L. Wilcox y otros, “Segmentation of speech using speaker identification,” Proc. ICASSP, pp. I-161-164, 1994.
28) F. J. Bimbot y otros, “A tutorial on text-independent speaker verification,” EURASIP Journ. on Applied Signal Processing, pp. 430-451, 2004.
29) G. R. Doddington, “Speaker recognition based on idiolectal differences between speakers,” Proc. Eurospeech, pp. 2521-2524, 2001.
30) D. Reynold, T. Quatieri, R. Dunn (2000). “Sepaker Verification Using Adapted Gaussian Mixture Models”. Digital Signal Processing, (10), N. 1-3, 19-41.
31) F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau,S: Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, D. Reynold, (2004). “A Tutorial on Text Independient Speaker Verification”. EURASIP Journal on Applied Signal Processing, (4), 430-451.
32) K. Farrel, R. Mammone, A. Richard; K. Assalch (1994). “Speaker Recognition Using Neural Network and Conventional Classifiers”. IEEE Tran. on Speech and Audio Processing, (2), No 1, Parte II.
33) J. Campbell (1997). “Speaker Recognition: A Tutorial”. Proc. of the IEEE, (85), No 9, 1437-1462.
34) C. Burges (1998). “A Tutorial on Support Vector Machinas for Pattern Recognition”. DM(2) 121-167.
35) B. Kenny (2006). “Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms”.
36) J. Gonzalez-Rodriguez, P. Rose, D. Ramos, D. Toledano, J. Ortega-García (2007). “Emulating DNA: Rigorous Quantification of Evidential Weight in Transparent and Testable Forensic Speaker Recognition”. IEEE Tran. on Audio, Speech and Language processing, (15), No 7, 2104-2114.
37) J. Gonzalez-Rodriguez, S. Drygajlo, D. Ramos-Castro, Daniel, M. Garcia-Gomar, J. Ortega-García, (2005). “Robust estimation, interpretation and assessment of likelihood ratios in forensic speaker recognition”. Computer Speech and Language, (20), 331-355.

38) L. Rabiner, B. H. Juang (1997). "Fundamentals of Speech Recognition". Prentice Hall, NY, USA.
39) Bimbot, F. J., Bonastre, F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-Garcia, J., Petrovska-Delacretaz D. and Reynolds, D. A. (2004) "A Tutorial on Text-Independent Speaker Verification," EURASIP Journ. on Applied Signal Processing, pp. 430-451.
40) Fauve, B. G. B., Matrouf, D., Scheffer, N., and Bonastre, J.-F (2007) "State-of-the-Art Performance in Text-Independent Speaker Verification through Open-Source Software," IEEE Trans. On Audio, Speech, and Language Process., 15, 7, pp. 1960-1968.
41) Furui, S. (1991) "Speaker-Independent and Speaker-Adaptive Recognition Techniques," in Furui, S. and Sondhi, M. M. (Eds.) Advances in Speech Signal Processing, New York: Marcel Dekker, pp. 597-622.
42) Furui, S. (1997) "Recent Advances in Speaker Recognition", Proc. First Int. Conf. Audio- and Video-based Biometric Person Authentication, Crans-Montana, Switzerland, pp. 237-252.
43) Furui, S. (2000) Digital Speech Processing, Synthesis, and Recognition, 2nd Edition, New York: Marcel Dekker.
44) Yin, S.-C., Rose, R. and Kenny, P. (2007) "A Joint factor Analysis Approach to Progressive Model Adaptation in Text-Independent Speaker Verification," IEEE Trans. On Audio, Speech, and Language Process., 15, 7, pp. 1999-2010.
45) Douglas A. Reynolds (2001) "An Overview of Automatic Speaker Recognition Technology", MIT Lincoln Laboratory, Lexington, MA USA
46) Amarin Deemagarn, Asanee Kawtrakul (2004) "Thai Connected Digit Speech Recognition Using Hidden Markov Models", Natural Language Processing and Intelligent Information System Technology Research Laboratory, Department of Computer Engineering, Kasetsart University, Bangkok, 10900, THAILAND
47) B. Plannerer (2005) "An Introduction to Speech Recognition", Bernd Plannerer, Munich, Germany
48) Vladan Velisavljevic, Christian Cornaz, Urs Hunkeler (2003) "An automatic speaker recognition system", Digital Signal Processing, Ecole Polytechnique Federale de Lausanne
49) Jon A. Gomez, Leandro Graci (2009) "BioVoiceDemo: Prototipo para Verificación de Locutor en la plataforma iPhone", Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Spain
50) Reynolds, D. A., Campbell, W. M.: (2007) "Text-Independent Speaker Recognition", Springer Handbook of Speech Processing and Communication, Springer-Verlag GMBH, Heidelberg, Germany, 763{781 (2007).
51) PHILIP ROSE, (2002) "*Forensic Speaker Identification"*. Nueva York, Taylor & Francis Inc. Capítulo II.
52) MARK PITTS, (2008) "*Writing in Maya Glyphs: Names, Places, & Simple Sentences*". El Proyecto de Ayuda y Educación
53) DANIEL T. LAROSE, (2006) "*Data mining methods and models*". Nueva Jersey, John Wiley & Sons Inc.
54) M. Deza, E. Deza, (2006) "*Dictionary of Distances*". Elsevier.

55) JIEHUA DAI, ZHENGZHE WEI, (2007) "*Study and Implementation of Feature Extraction and Comparison In Voice Recognition*". Suiza, Universidad Linköpings.
56) DOUGLAS A. REYNOLDS, LARRY P. HECK, (2000) "*Automatic Speaker Recognition Recent Progress, Current Applications and Future Trends*", USA, M.I.T. Lincoln Laboratory & Nuance Communications
57) EVGENY KARPOV, (2003) "*Real-Time Speaker Identification*", Finlandia, Universidad de Joensuu
58) RONALD A. COLE, JOSEPH MARIANI, HANS USZKOREIT, ANNIE ZAENEN, VICTOR ZUE, (1996) "*Survey of the State of the Art in Human Language Technology*", Italia, Universidad de Pisa.
59) SHOU-CHUN YIN, RICHARD ROSE, PATRICK KENNY, (2008) "*Adaptive Score Normalization For Progressive Model Adaptation In Text Independent Speaker Verification*", Canada, Universidad McGill
60) D. E. Sturim and D. A. Reynolds, (2005) "*Speaker Adaptive Cohort Selection For Tnorm In Text-Independent Speaker Verification*", USA, Universidad MIT
61) Ville Hautamäki, Marko Tuononen, Tuija Niemi-Laitinen y Pasi Fränti, (2007) "Improving Speaker Verification", Finland, Speech and Image Processing Unit, Department of Computer Science and Statistics, University of Joensuu
62) Davis, S., and Mermelstein, P. (1980). "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences" In IEEE Transactions on Acoustics, Speech and Signal Processing 28, p. 357-366.
63) Oppenheim, A., and Schafer, R. (1968). "Homomorphic Analysis of Speech" In IEEE Transactions on Audio and Electroacoustics 16, p. 221-226.
64) Bogert, B., Healy, M., and Tukey, J. (1963). "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking. In Proceedings of the Symposium on Time Series Analysis", p. 209-243.
65) JONATHAN J. LAREAU (2006) "Application of Shifted Delta Cepstral Features for GMM Language Identification", Department of Computer Science, Rochester Institute of Technology.
66) MARK D. SKOWRONSKI, JOHN G. HARRIS (2002) "Increased MFCC Filter Bandwidth for Noise-Robust Phoneme Recognition", Computational Neuro-Engineering Laboratory, University of Florida
67) BEN J. SHANNON, KULDIP K. PALIWAL (2003) "A Comparative Study of Filter Bank Spacing for Speech Recognition". Microelectronic Engineering Research Conference
68) FANG ZHENG, GUOLIANG ZHANG, ZHANJIANG SONG, (2001) "Comparison of Different Implementations of MFCC". Beijing, China, Department of Computer Science and Technology, Tsinghua University

ANEXO 1: CÓDIGOS EN MATLAB

Código para la creación de las gráficas del capítulo 2, éste se compone de varias funciones o módulos, el llamado 'grafica', dispara la carga de la información, su procesado y la generación de las gráficas en una, dos o tres dimensiones, dependiendo de los parámetros que se alimentan al ejecutarla.

La función 'est' obtiene los valores estadísticos y consume a su vez a las funciones 'carga' que carga la información de un archivo 'pit', que genera el programa SIS (formato usado para guardar la información de tono fundamental), 'cuenta' que separa las cinco repeticiones de cada locutor que vienen en cada archivo, 'param' que obtiene los valores estadísticos de cada iteración realizada por un locutor, y 'param2' que obtiene la estadística de los valores ya obtenidos, por lo que serían parámetros estadísticos de los parámetros estadísticos del tono fundamental.

	Código
grafica.m	

```
function []=grafica(par1,par2,par3)
% par2 = 0 para una dimensión en una línea
% par2 = -1 para una dimensión en
% varias líneas
% par3 = 0 para dos dimensiones
% par1 = parámetro a dibujar de 1 a 6
% Carga los tonos fundamentales de SIS y los procesa

r0=est('Data\0.pit');
r1=est('Data\1.pit');
r2=est('Data\2.pit');
r3=est('Data\3.pit');
r4=est('Data\4.pit');
r5=est('Data\5.pit');
r6=est('Data\6.pit');
r7=est('Data\7.pit');
r8=est('Data\8.pit');
r9=est('Data\9.pit');

% Graficas en una, dos o tres dimensiones

if par2<1
    if par2==0
        % Dibuja una sola dimensión en un sola línea
        plot(r0(par1,1:5),0,'bo')
        hold on;
        plot(r1(par1,1:5),0,'gx')
        plot(r2(par1,1:5),0,'r+')
        plot(r3(par1,1:5),0,'c*')
        plot(r4(par1,1:5),0,'ms')
        plot(r5(par1,1:5),0,'cd')
        plot(r6(par1,1:5),0,'kv')
        plot(r7(par1,1:5),0,'bp')
        plot(r8(par1,1:5),0,'g<')
        plot(r9(par1,1:5),0,'r>')
        hold off;
    elseif par2==-1
        % Dibuja una dimensión con desfase de 1 en 'y'
        plot(r0(par1,1:5),0,'bo')
        hold on;
        plot(r1(par1,1:5),1,'gx')
```

```
            plot(r2(par1,1:5),2,'r+')
            plot(r3(par1,1:5),3,'c*')
            plot(r4(par1,1:5),4,'ms')
            plot(r5(par1,1:5),5,'cd')
            plot(r6(par1,1:5),6,'kv')
            plot(r7(par1,1:5),7,'bp')
            plot(r8(par1,1:5),8,'g<')
            plot(r9(par1,1:5),9,'r>')
        end
    elseif par3==0
        % Dibuja en dos dimensiones
        plot(r0(par1,1:5),r0(par2,1:5),'bo')
        hold on;
        plot(r1(par1,1:5),r1(par2,1:5),'gx')
        plot(r2(par1,1:5),r2(par2,1:5),'r+')
        plot(r3(par1,1:5),r3(par2,1:5),'c*')
        plot(r4(par1,1:5),r4(par2,1:5),'ms')
        plot(r5(par1,1:5),r5(par2,1:5),'cd')
        plot(r6(par1,1:5),r6(par2,1:5),'kv')
        plot(r7(par1,1:5),r7(par2,1:5),'bp')
        plot(r8(par1,1:5),r8(par2,1:5),'g<')
        plot(r9(par1,1:5),r9(par2,1:5),'r>')
    else
        % Dibuja en tres dimensiones
        plot3(r0(par1,1:5),r0(par2,1:5),r0(par3,1:5),'bo')
        hold on;
        plot3(r1(par1,1:5),r1(par2,1:5),r1(par3,1:5),'gx')
        plot3(r2(par1,1:5),r2(par2,1:5),r2(par3,1:5),'r+')
        plot3(r3(par1,1:5),r3(par2,1:5),r3(par3,1:5),'c*')
        plot3(r4(par1,1:5),r4(par2,1:5),r4(par3,1:5),'ms')
        plot3(r5(par1,1:5),r5(par2,1:5),r5(par3,1:5),'cd')
        plot3(r6(par1,1:5),r6(par2,1:5),r6(par3,1:5),'kv')
        plot3(r7(par1,1:5),r7(par2,1:5),r7(par3,1:5),'bp')
        plot3(r8(par1,1:5),r8(par2,1:5),r8(par3,1:5),'g<')
        plot3(r9(par1,1:5),r9(par2,1:5),r9(par3,1:5),'r>')
        hold off;
    end
    end
```

carga.m

Código

```
function [data]=carga(archivo)

delimitador = ' ';  % Definición del delimitador
encabezado = 3;     % Número de líneas del encabezado

% Da lectura al archivo del tono fundamental
newData1 = importdata(archivo, delimitador,
encabezado);


data=newData1.('data');
```

cuenta.m

Código

```
function [data1, data2, data3, data4,
data5]=cuenta(data)
% Función que separa las 5 repeticiones de una palabra
en un mismo archivo y los coloca en la variable dataX
    s=size(data);
    s2=1;
```

```
    for y =1:5
    for x=s2:s(1)
        if data(x,2)==0
            % Sigue buscando
        else
            s1=x;
            break;
        end;
    end
    for x=s1:s(1)
        if data(x,2)==0
            s2=x;
            break;
        else
            % Sigue contando
        end;
    end
    switch y
        case 1
            data1=data(s1:s2-1,2);
        case 2
            data2=data(s1:s2-1,2);
        case 3
            data3=data(s1:s2-1,2);
        case 4
            data4=data(s1:s2-1,2);
        case 5
            data5=data(s1:s2-1,2);
    end
    end
end
```

	Código
est.m	

```
function [todo]=est(nom)

    % Variable intermedia para almacenar información
    todo=zeros(6,11);
    [d1 d2 d3 d4 d5]=cuenta(carga(nom));
    % Llama a la separación y carga de los archivos

    % Procesado de las 5 repeticiones
    todo(:,1)=param(d1);
    todo(:,2)=param(d2);
    todo(:,3)=param(d3);
    todo(:,4)=param(d4);
    todo(:,5)=param(d5);
    % Estadística de los datos estadísticos de las 5
repeticiones
    todo(:,6)=param2(todo(1,1:5));
    todo(:,7)=param2(todo(2,1:5));
    todo(:,8)=param2(todo(3,1:5));
    todo(:,9)=param2(todo(4,1:5));
    todo(:,10)=param2(todo(5,1:5));
    todo(:,11)=param2(todo(6,1:5));
end
```

	Código
param.m	

```
function [param]=param(data1)
    % Obtiene datos estadísticos de una serie de datos
```

```
de tono fundamental
    m1=mean(data1(:,1));
    m2=median(data1(:,1));
    m3=std(data1(:,1));
    m4=var(data1(:,1));
    m5=skewness(data1(:,1));
    m6=kurtosis(data1(:,1));
    param=[m1 m2 m3 m4 m5 m6];
end
```

	Código

param2.m

```
function [param]=param2(data1)
    m1=mean(data1);
    m2=median(data1);
    m3=std(data1);
    m4=var(data1);
    m5=skewness(data1);
    m6=kurtosis(data1);
    param=[m1 m2 m3 m4 m5 m6];
end
```

Filtro de pre-énfasis:

	Código

PreEnfasis.m

```
function [dataOut]=PreEnfasis(dataIn, a)

% dataIn datos a filtrar
% a valor entre 0 y 1 para el filtro

% Realiza el proceso de filtrado
dataOut = filter([1 -a],1,dataIn);
```

Código para crear ventanas

	Código

Ventana.m

```
function [dataOut]=Ventana(dataIn, tamano, tipo)

% dataIn datos a aplicar la ventana
% tamano es el tamaño de la ventana a emplear
% tipo indica el tipo de ventana de acuerdo a la
siguiente tabla
% 1 - Ventana modificada Bartlett-Hanning
% 2 - Ventana Bartlett
% 3 - Ventana Blackman
% 4 - Ventana Blackman-Harris
% 5 - Ventana de Bohman
% 6 - Ventana de Chebyshev
% 7 - Ventana de parte alta plana
% 8 - Ventana Gaussiana
% 9 - Ventana de Hamming
% 10 - Ventana de Hann
% 11 - Ventana de Kaiser
% 12 - Ventana de Nuttall (Blackman-Harris de N puntos)
% 13 - Ventana de Parzen
% 14 - Ventana rectangular
% 15 - Ventana de Taylor
```

```
% 16 - Ventana triangular
% 17 - Ventana de Tukey

% Aplica la ventana
switch tipo
    case 1
        dataOut=dataIn.*barthannwin(tamano);
    case 2
        dataOut=dataIn.*bartlett(tamano);
    case 3
        dataOut=dataIn.*blackman(tamano);
    case 4
        dataOut=dataIn.*blackmanharris(tamano);
    case 5
        dataOut=dataIn.*bohmanwin(tamano);
    case 6
        dataOut=dataIn.*chebwin(tamano);
    case 7
        dataOut=dataIn.*flattopwin(tamano);
    case 8
        dataOut=dataIn.*gausswin(tamano);
    case 9
        dataOut=dataIn.*hamming(tamano);
    case 10
        dataOut=dataIn.*hann(tamano);
    case 11
        dataOut=dataIn.*kaiser(tamano);
    case 12
        dataOut=dataIn.*nuttallwin(tamano);
    case 13
        dataOut=dataIn.*parzenwin(tamano);
    case 14
        dataOut=dataIn.*rectwin(tamano);
    case 15
        dataOut=dataIn.*taylorwin(tamano);
    case 16
        dataOut=dataIn.*triang(tamano);
    case 17
        dataOut=dataIn.*tukeywin(tamano);
end
```

Función para crear un banco de filtros

bancoFiltros.m

Código

```
function [ dataOut ] = bancoFiltros( k, tipo, forma,
tra, fre, minF, maxF, datos,dibuja )
% Funcion que crea el filtro y lo aplica
%   k - número de filtros (coeficientes)
%   tipo - tipo de filtro (lineal, Mel o Bark)
%           1 - Mel
%           2 - Bark
%           3 - Lineal
%   forma - forma del filtro (17 tipos)
%           1 - Ventana modificada Bartlett-Hanning
%           2 - Ventana Bartlett
%           3 - Ventana Blackman
%           4 - Ventana Blackman-Harris
%           5 - Ventana de Bohman
%           6 - Ventana de Chebyshev
```

```
%            7 - Ventana de parte alta plana
%            8 - Ventana Gaussiana
%            9 - Ventana de Hamming
%            10 - Ventana de Hann
%            11 - Ventana de Kaiser
%            12 - Ventana de Nuttall (Blackman-Harris de
N puntos)
%            13 - Ventana de Parzen
%            14 - Ventana rectangular
%            15 - Ventana de Taylor
%            16 - Ventana triangular
%            17 - Ventana de Tukey
%    tra - porcentaje de traslape (valor de 0 a 1)
%    fre - frecuencia de muestreo
%    minF - frecuencia minima
%    maxF - frecuencia maxima
%    datos - datos a filtrar
%    dibuja - 1 para dibujar el filtro

% Aplica la ventana
dataIn=cast(datos,'double');
tam=size(dataIn);
dataOut=zeros(k,tam(2));
filtros=zeros(tam(1)/2,k);
dF=1/(tam(1)*(1/fre));
% Calcula espaciamiento de acuerdo a la escala
pares=zeros(k,2);
switch tipo
    case 1
        % Mel
        minFec=1127*log(1+(minF/700));
        maxFec=1127*log(1+(maxF/700));
        dFec=(maxFec-minFec)/k;
        finFec=minFec;
        for X = 1:k
            if (X>1)
                inicioConTras=finFec-(dFec*tra);
            else
                inicioConTras=finFec;
            end
            if (X<k)
                finConTras=(finFec+dFec)+(dFec*tra);
            else
                finConTras=(finFec+dFec);
            end

pares(X,1)=round((700*(exp((inicioConTras)/1127)-
1))/dF);

pares(X,2)=round((700*(exp(finConTras/1127)-1))/dF);
            finFec=finFec+dFec;
        end
    case 2
        % Bark
        minFec=(26.81/(1+(1960/minF)))-0.53;
        maxFec=(26.81/(1+(1960/maxF)))-0.53;
        dFec=(maxFec-minFec)/k;
        finFec=minFec;
        for X = 1:k
            if (X>1)
```

```
                inicioConTras=finFec-(dFec*tra);
            else
                inicioConTras=finFec;
            end
            if (X<k)
                finConTras=(finFec+dFec)+(dFec*tra);
            else
                finConTras=(finFec+dFec);
            end

pares(X,1)=round((1960/((26.81/(inicioConTras+0.53))-
1))/dF);

pares(X,2)=round((1960/((26.81/(finConTras+0.53))-
1))/dF);
            finFec=finFec+dFec;
        end
    case 3
        % Lineal
        minFec=minF;
        maxFec=maxF;
        dFec=(maxFec-minFec)/k;
        finFec=minFec;
        for X = 1:k
            if (X>1)
                inicioConTras=finFec-(dFec*tra);
            else
                inicioConTras=finFec;
            end
            if (X<k)
                finConTras=(finFec+dFec)+(dFec*tra);
            else
                finConTras=(finFec+dFec);
            end
            pares(X,1)=round(inicioConTras/dF);
            pares(X,2)=round(finConTras/dF);
            finFec=finFec+dFec;
        end
end
% Construye banco de filtros
for X = 1:k
    switch forma
        case 1
    f=vertcat(zeros(pares(X,1)-1,1),
barthannwin(pares(X,2)-pares(X,1)));
        case 2
    f=vertcat(zeros(pares(X,1)-1,1),
bartlett(pares(X,2)-pares(X,1)));
        case 3
    f=vertcat(zeros(pares(X,1)-1,1),
blackman(pares(X,2)-pares(X,1)));
        case 4
    f=vertcat(zeros(pares(X,1)-1,1),
blackmanharris(pares(X,2)-pares(X,1)));
        case 5
    f=vertcat(zeros(pares(X,1)-1,1),
bohmanwin(pares(X,2)-pares(X,1)));
        case 6
    f=vertcat(zeros(pares(X,1)-1,1),
chebwin(pares(X,2)-pares(X,1)));
        case 7
```

```
    f=vertcat(zeros(pares(X,1)-1,1),
flattopwin(pares(X,2)-pares(X,1)));
        case 8
    f=vertcat(zeros(pares(X,1)-1,1),
gausswin(pares(X,2)-pares(X,1)));
        case 9
    f=vertcat(zeros(pares(X,1)-1,1),
hamming(pares(X,2)-pares(X,1)));
        case 10
    f=vertcat(zeros(pares(X,1)-1,1), hann(pares(X,2)-
pares(X,1)));
        case 11
    f=vertcat(zeros(pares(X,1)-1,1), kaiser(pares(X,2)-
pares(X,1)));
        case 12
    f=vertcat(zeros(pares(X,1)-1,1),
nuttallwin(pares(X,2)-pares(X,1)));
        case 13
    f=vertcat(zeros(pares(X,1)-1,1),
parzenwin(pares(X,2)-pares(X,1)));
        case 14
    f=vertcat(zeros(pares(X,1)-1,1),
rectwin(pares(X,2)-pares(X,1)));
        case 15
    f=vertcat(zeros(pares(X,1)-1,1),
taylorwin(pares(X,2)-pares(X,1)));
        case 16
    f=vertcat(zeros(pares(X,1)-1,1), triang(pares(X,2)-
pares(X,1)));
        case 17
    f=vertcat(zeros(pares(X,1)-1,1),
tukeywin(pares(X,2)-pares(X,1)));
    end
    f=vertcat(f,zeros(((tam(1)/2)-pares(X,2))+1,1));
    filtros(:,X)=f;
end
% Opcional, dibuja el filtro obtenido
if (dibuja==1)
    cla;
    hold on;
    for X=1:k

plot(linspace(minF,maxF,tam(1)/2),filtros(:,X)');
    end
end
% Aplica el filtro a los datos
%dataOut=zeros(k,tam(2));
%filtros=zeros(tam(1)/2,k);
limite=tam(1)/2;
for X=1:tam(2)
    for Y=1:k

dataOut(Y,X)=sum(filtros(:,Y).*abs(datos(1:limite,X)));
    end
end
end
```

Detector de actividad vocal

	Código
vad.m	

```
function [a, SNR,
habla]=vad(senal,fr,tamVentana,disparadorEnergia,disparadorFrecue
ncia,puntosG,queG)
% senal - señal a trabajar
% fr - frecuencia de muestreo
% tamVentana - tamaño de la ventana en ms
% disparadorEnergia - disparador (triger) en energía
% disparadorFrecuencia - disparador (triger) en frecuencia
% puntosG - puntos a graficar
%               0 = no graficar
% queG - que graficar 1 WAV 2 Energía 3 Frecuencia [1 1 1]


puntosVentana=floor(fr*(tamVentana/1000));
tam=size(senal);
ventanas=floor(tam(1)/puntosVentana);
inicio=1;
E=zeros(ventanas,1);
maxFr=zeros(ventanas,1);
for X=1:ventanas
    fin=inicio+puntosVentana;
    if (fin>tam(1))
        fin=tam(1);
    end
    dato=cast(senal(inicio:fin,1),'double');
    E(X,1)=sum(dato.^2);
    trans=abs(fft(dato(:,1)));
    maxFr(X,1)=max(trans);
    inicio=inicio+puntosVentana;
end
% Calculo de los disparadores
dE=disparadorEnergia*((max(E)-min(E))/100);
dMaxFr=disparadorFrecuencia*((max(maxFr)-min(maxFr))/100);
% Toma de decicion
a=zeros(tam(1),1);
sil=zeros(tam(1),1);
b=zeros(tam(1),1);
inicio=1;
for X=1:ventanas
    con=0;
    if (E(X)>=dE)
        con=con+1;
    end
    if (maxFr(X)>=dMaxFr)
        con=con+1;
    end
    fin=inicio+puntosVentana;
    if (fin>tam(1))
        fin=tam(1);
    end
    if (con>=1)
        % La señal con voz
        a(inicio:fin,1)=senal(inicio:fin,1);
        b(inicio:fin,1)=28000;
    else
        % El ruido
        sil(inicio:fin,1)=senal(inicio:fin,1);
```

```
    end
    inicio=inicio+puntosVentana;
end
% Calcula el SNR
SNR=20*log10(mean(abs(a))/mean(abs(sil)));
% Elimina los ceros
a = a(a~=0);
ta=size(a);
% Calcula el habla pura
habla=ta(1)/fr;
% Grafica si se pide
if (puntosG>0)
    grafVAD(senal,E,maxFr,puntosG,puntosVentana,queG,b);
end
```

	Código

grafVAD.m

```
function [  ] = grafVAD( se,e,f,tam,ven,que,a )
%UNTITLED2 Summary of this function goes here
%   se - señal original
%   e - energía
%   f - frecuencia máxima
%   tam - tamaño de la grafica
%   ven - tamaño de la ventana
%   que - que se desea graficar
%   a - grafica simple para mostrar que se eliminara

% Selecciona que graficar
nG=sum(que); %número de graficas
cual=1;
tam2=tam/ven;

% Grafica del WAV
if (que(1)==1)
    subplot(nG,1,cual,'align')
    plot(se(1:tam,1));
    hold
    plot(a(1:tam,1),'g');
    axis([1,tam,-32768,32768]);
    cual=cual+1;
end

% Grafica de la energía
if (que(2)==1)
    subplot(nG,1,cual,'align')
    plot(e(1:tam2,1));
    axis([1,tam2,min(e),max(e)]);
    cual=cual+1;
end
% Grafica de la frecuencia máxima
if (que(3)==1)
    subplot(nG,1,cual,'align')
    plot(f(1:tam2,1));
    axis([1,tam2,min(f),max(f)]);
end
end
```

Cuantización vectorial

	Código
VQdata.m	

```
function [ dataOut ] = VQdata( dataIn, tamano,
incremento , disAceptable )
% Función que realiza la cuantización vectorial por
algoritmo LBG
% dataIn - datos de entrada
% tamano - es el tamaño del libro, es siempre par por
el método
% incremento - incremento para la división de vectores
% disAceptable - distancia promedio mínima aceptable

tam=size(dataIn);
dataOut=zeros(tam(1),tamano);

% Normaliza los datos
dataIn=reduceF(dataIn);

% Creación del vector 1
for X=1:tam(1)
    dataOut(X,1)=mean(dataIn(X,:));
end

N=1;
NN=zeros(tam(2),1); % Nearest-Neighbor
disProm=zeros(tam(2),1);
while N<tamano
   % Divide vectores
   for X=1:N
        dataOut(:,X+N)=dataOut(:,X)+incremento;
        dataOut(:,X)=dataOut(:,X)-incremento;
   end
   N=N*2;
   disProm(:,1)=0;
   while mean(disProm(:,1))<disAceptable
        % Busqueda del vecino mas cercano
        for X=1:tam(2)
            % Busca contra todos los codeword
            for Y=1:N

dis=distancia(dataIn(:,X),dataOut(:,Y));
                if (Y==1)
                    NN(X,1)=1;
                    d=dis;
                    disProm(X,1)=dis;
                else
                    if (d>dis)
                        NN(X,1)=Y;
                        d=dis;
                        disProm(X,1)=dis;
                    end
                end
            end
        end
        % Actualiza centroides
        dataOut(:,:)=0;
        for X=1:tam(1)
            for Y=1:tam(2);
```

```
dataOut(X,NN(Y,1))=dataOut(X,NN(Y,1))+dataIn(X,Y);
            end
        end
        for X=1:N
            dataOut(:,X)=dataOut(:,X)/sum(NN==X);
        end
   end
end

end
```

Calculo de distancia entre vectores

distancia.m

```
function [ distanciaOut ] = distancia( arg1 , arg2 )
% Calcula la distancia Euclidiana entre los argumentos
% arg1 - Argumento 1 (Vectores del mismo tamaño)
% arg2 - Argumento 2 (Vectores del mismo tamaño)

distanciaOut=arg1-arg2;
distanciaOut=distanciaOut.^2;
distanciaOut=sum(distanciaOut);
distanciaOut=sqrt(distanciaOut);
end
```

Análisis de resultados

analiza.m

```
function [
tabla,EER,porIdentificados,porRechazados,disparador,calificac
ion ] = analiza( datosEntre, datosIntra, estadisticaEntre,
estadisticaIntra, divisiones,gra )
% Función que evalúa los resultados
%   datosEntre - Las distancias de comparativos entre
locutores
%   datosIntra - Las distancias de comparativos intra locutor
%   estadisticaIntra - Estadísticos de comparativos intra
locutor
%   estadisticaEntre - Estadísticos de comparativos entre
locutores
%   divisiones - número de divisiones de la tabla
%   gra - Grafica el FRR contra el FAR
%           1 - FAR contra FRR
%           2 - ROC

tabla=zeros(divisiones,9);
tamI=size(datosIntra);
tamE=size(datosEntre);
% Busca los minimos y los maximos
if (datosIntra(1)<datosEntre(1))
    minD=estadisticaIntra(1);
else
    minD=estadisticaEntre(1);
end
if (datosIntra(2)<datosEntre(2))
    maxD=estadisticaEntre(2);
```

```
else
    maxD=estadisticaIntra(2);
end
inicio=minD;
divD=(maxD-minD)/divisiones;
for X=1:divisiones
    tabla(X,1)=inicio;
    tabla(X,2)=inicio+divD;
    tabla(X,3)=inicio+(divD/2);

cuenta=size(find(datosIntra>tabla(X,1)&datosIntra<tabla(X,2))
);
    tabla(X,4)=cuenta(1);
    tabla(X,5)=cuenta(1)/tamI(1);

cuenta=size(find(datosEntre>tabla(X,1)&datosEntre<tabla(X,2))
);
    tabla(X,6)=cuenta(1);
    tabla(X,7)=cuenta(1)/tamE(1);
    inicio=inicio+divD;
end
% Obtiene los valores de FRR y FAR para cada distancia
for X=1:(divisiones+1)
    % Calculo de FRR
    por=0;
    for Y=(X-1):-1:1
        por=por+tabla(Y,5);
    end
    tabla(X,8)=1-por; %Se invierten pues a mayor distancia
menos probable
    % Calculo de FAR
    por=0;
    for Y=X:divisiones
        por=por+tabla(Y,7);
    end
    tabla(X,9)=1-por;
end
b=[1:1:101];
[pB,EER]=intersections(b,tabla(:,9),b,tabla(:,8));
% Saca media si hay más de un punto
pB=mean(pB);
EER=mean(EER);
if (gra==1)
    plot(b,tabla(:,9),'g');
    hold;
    plot(b,tabla(:,8),'r');
    axis([0,100,0,1.1]);
    title('FAR contra FRR')
    xlabel('Disparador (sensibilidad)');
    ylabel('Error');
    cada=divisiones/10;
    divG=(maxD-minD)/10;
    set(gca, 'XTick',[0 cada cada*2 cada*3 cada*4 cada*5
cada*6 cada*7 cada*8 cada*9 divisiones])
    set(gca,
'XTickLabel',{num2str(minD,'%3.2f'),num2str(minD+(divG*1),'%3
.2f'),num2str(minD+(divG*2),'%3.2f'),num2str(minD+(divG*3),'%
3.2f'),num2str(minD+(divG*4),'%3.2f'),num2str(minD+(divG*5),'
%3.2f'),num2str(minD+(divG*6),'%3.2f'),num2str(minD+(divG*7),
'%3.2f'),num2str(minD+(divG*8),'%3.2f'),num2str(minD+(divG*9)
,'%3.2f'),num2str(maxD,'%3.2f')})
```

```
    text(5,1.05,'FAR');
    text(90,1.05,'FRR');
    text(pB+5,EER,'EER');
else
    if (gra==2)
        plot(tabla(:,9),1-tabla(:,8));
        axis([0,1,0,1]);
        title('ROC')
        ylabel('Tasa de verificación');
        xlabel('Tasa de falsa aceptación');
    end
end
% Calcula los porcentajes de aciertos en la identificación y
en el descarte
% de acuerdo al EER
disparador=tabla(round(pB),3);
cuenta=find(datosIntra<disparador);
tam=size(cuenta);
porIdentificados=tam(1)/tamI(1);
cuenta=find(datosEntre>disparador);
tam=size(cuenta);
porRechazados=tam(1)/tamE(1);
calificacion=(1-EER)+porIdentificados+porRechazados;
end
% 1-minimo 2-maximo 3-media 4-Conteo Intra 5-Fraccion Intra
6-Conteo entre
% 7-Fraccion entre 8-FRR 9-FAR
% Equal Error Rate

% calificación - "3" el sistema perfecto "0" un muy mal
sistema
```

Generador de bloque de pruebas

	Código

TODO.m

```
% Tamaño de la ventana para Fourier
v1=[64 128 256 512 1024 2048];
% Traslape de la ventana de Fourier
v2=[0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1];
% Valor de alfa para pre-énfasis
v3=[-1 0 .1 .2 .3 .4 .5 .6 .7 .8 .9 .95 1];
% Numero de coeficientes MFCC
v4=[10 20 25 30 35 40 45 50];
% Traslape de los filtros
v5=[0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1];
% Frecuencia mínima de corte
v6=[50 100 150 200 250 300];
% Frecuencia máxima de corte
v7=[5000 4500 4000 3500 3000];
% Reduce vectores
v8=[1 0];
% Centra vectores
v9=[1 0];
% Orden de aproximación para las deltas
v10=[1 2 3 4 5 6 7 8];
% Utilizar coeficientes delta
v11=[1 0];
% Utilizar coeficientes delta-delta
v12=[1 0];
```

```
% Utilizar coeficientes de energía
v13=[1 0];
% Tamaño del modelo por cuantización vectorial
v14=[8 16 32 64 128 256];
% Desplazamiento de los vectores
v15=[0.01 .015 .02 .025 .03];
% Distancia promedio mínima de estabilidad
v16=[.01 .001 .0001];
% Aplicar detector de actividad vocal
v17=[1 0];
% Tamaño de la ventana en [ms]
v18=[5 10 15 20];
% Escalón de corte en energía
v19=[1 2 3 4];
% Escalón de corte en frecuencia
v20=[1 2 3 4];
% Tipo de filtro para Fourier
v21=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17];
% Tipo de filtro para el MFCC
v22=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17];
% Tipo de transformación Mel, Bark o lineal
v23=[1 2 3];
% DCT o DFT
v24=[1 2];
% Crea las combinaciones y las guarda
tam=zeros(24,2);
tam(1,:)=size(v1);
tam(2,:)=size(v2);
tam(3,:)=size(v3);
tam(4,:)=size(v4);
tam(5,:)=size(v5);
tam(6,:)=size(v6);
tam(7,:)=size(v7);
tam(8,:)=size(v8);
tam(9,:)=size(v9);
tam(10,:)=size(v10);
tam(11,:)=size(v11);
tam(12,:)=size(v12);
tam(13,:)=size(v13);
tam(14,:)=size(v14);
tam(15,:)=size(v15);
tam(16,:)=size(v16);
tam(17,:)=size(v17);
tam(18,:)=size(v18);
tam(19,:)=size(v19);
tam(20,:)=size(v20);
tam(21,:)=size(v21);
tam(22,:)=size(v22);
tam(23,:)=size(v23);
tam(24,:)=size(v24);
num=1;
for X=1:24
    num=num*tam(X,2);
end
num
c=combvec(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,
v16,v17,v18,v19,v20,v21,v22,v23,v24);
parametros=c';
save('TODO.mat','parametros');
```

Combinación de parámetros

	Código
combina.m	

```
function [  ] = combina( nombre,expArchivo )
% Nombre de la salida
% Numero de combinaciones máximo por archivo

% Tamaño de la ventana para Fourier
v1=[64 128 256 512 1024 2048];
% Traslape de la ventana de Fourier
v2=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];
% Valor de alfa para pre-énfasis
v3=[-1 0 .1 .2 .3 .4 .5 .6 .7 .8 .9 .95 1];
% Numero de coeficientes MFCC
v4=[10 20 25 30 35 40 45 50];
% Traslape de los filtros
v5=[0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1];
% Frecuencia mínima de corte
v6=[50 100 150 200 250 300];
% Frecuencia máxima de corte
v7=[5000 4500 4000 3500 3000];
% Reduce vectores
v8=[1 0];
% Centra vectores
v9=[1 0];
% Orden de aproximación para las deltas
v10=[1 2 3 4 5 6 7 8];
% Utilizar coeficientes delta
v11=[1 0];
% Utilizar coeficientes delta-delta
v12=[1 0];
% Utilizar coeficientes de energía
v13=[1 0];
% Tamaño del modelo por cuantización vectorial
v14=[8 16 32 64 128 256];
% Desplazamiento de los vectores
v15=[0.01 .015 .02 .025 .03];
% Distancia promedio mínima de estabilidad
v16=[.01 .001 .0001];
% Aplicar detector de actividad vocal
v17=[1 0];
% Tamaño de la ventana en [ms]
v18=[5 10 15 20];
% Escalón de corte en energía
v19=[1 2 3 4];
% Escalón de corte en frecuencia
v20=[1 2 3 4];
% Tipo de filtro para Fourier
v21=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17];
% Tipo de filtro para el MFCC
v22=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17];
% Tipo de transformación Mel, Bark o lineal
v23=[1 2 3];
% DCT o DFT
v24=[1 0];

% Crea las combinaciones y las guarda
tam=zeros(24,2);
tam(1,:)=size(v1);
tam(2,:)=size(v2);
```

```
tam(3,:)=size(v3);
tam(4,:)=size(v4);
tam(5,:)=size(v5);
tam(6,:)=size(v6);
tam(7,:)=size(v7);
tam(8,:)=size(v8);
tam(9,:)=size(v9);
tam(10,:)=size(v10);
tam(11,:)=size(v11);
tam(12,:)=size(v12);
tam(13,:)=size(v13);
tam(14,:)=size(v14);
tam(15,:)=size(v15);
tam(16,:)=size(v16);
tam(17,:)=size(v17);
tam(18,:)=size(v18);
tam(19,:)=size(v19);
tam(20,:)=size(v20);
tam(21,:)=size(v21);
tam(22,:)=size(v22);
tam(23,:)=size(v23);
tam(24,:)=size(v24);
num=1;
for X=1:24
    num=num*tam(X,2);
end
fprintf(1,'%s','Combinaciones: ');
fprintf(1,'%d',num);
fprintf(1,'\n');
t2=num*1000*10; % Tiempo en segundos
t3=(((t2/60)/60)/24)/4; % Días por maquina
fprintf(1,'%s','Días con 4 máquinas: ');
fprintf(1,'%d',t3);
fprintf(1,'\n');
c=combvec(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15
,v16,v17,v18,v19,v20,v21,v22,v23,v24);
if (num>expArchivo)
    pasos=ceil(num/expArchivo);
    para=c';
    inicio=1;
    for X=1:pasos
        name=num2str(X);
        archivo=strcat(nombre,name,'.mat');
        fin=inicio+expArchivo-1;
        if (fin>num)
            fin=num;
        end
        parametros=para(inicio:fin,:);
        save(archivo,'parametros');
        inicio=inicio+expArchivo;
    end
else
    parametros=c';
    save(strcat(nombre,'.mat'),'parametros');
end
end
```