



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRIA Y DOCTORADO EN
INGENIERIA

HERRAMIENTA INTERACTIVA PARA MEJORAR
EL DESEMPEÑO EN LA PERCEPCIÓN
DEL LENGUAJE EN DÉBILES AUDITIVOS

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA

INGENIERÍA ELÉCTRICA - INSTRUMENTACIÓN

P R E S E N T A :

ANDRÉ PIMENTEL HERNÁNDEZ

TUTOR:

SANTIAGO JESÚS PÉREZ RUIZ

2012



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: DR. ORDUÑA BUSTAMANTE FELIPE
Secretario: DR. RENDÓN GARRIDO PABLO LUIS
Vocal: DR. PÉREZ RUIZ SANTIAGO JESÚS
1^{er}. Suplente: DRA. CASTAÑEDA GUZMÁN ROSALBA
2^{do}. Suplente: M. en I. PÉREZ LÓPEZ ANTONIO

Lugar donde se realizó la

tesis: CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO

TUTOR DE TESIS:

SANTIAGO JESÚS PÉREZ RUIZ



FIRMA

Agradecimientos

Me gustaría agradecer a las siguientes personas por ayudarme a aclarar procedimientos y conceptos relevantes a la elaboración de mi proyecto de tesis. También me gustaría agradecer a todas las personas que me brindaron su apoyo moral. Gracias.

M. en I. Israel De Olmos Ramirez UAM-I
M. en I. Jorge Matadamas Hernández UAM-I
Dr. Alfonso Prieto Guerrero UAM-I
Dr. Santiago Jesús Pérez Ruiz UNAM-CU
Dr. Pablo Luis Rendón Garrido UNAM-CU
M. en I. Antonio Pérez López UNAM-CU
Dr. Felipe Orduña Bustamante UNAM-CU

Índice

Resumen	5
1. Introducción	
1.1 Descripción de la Tesis	5
1.2 Contenido de la Tesis	6
2. Antecedentes	
2.1 Algunos Conceptos de Neurofisiología	8
2.2 Déficit Auditivo en Personas	10
2.3 Introducción a la Fonología y la Fonética	13
2.4 Procesamiento en el Dominio Tiempo contra Frecuencia	20
3. Codificador de Voz en Fase	
3.1 Transformada de Fourier en Tiempo Reducido (STFT)	21
3.2 Descripción del Codificador de Voz en Fase	24
3.3 Implementación del Codificador de Voz en Fase	26
4. Escalamiento en Tiempo	
4.1 Introducción	30
4.2 Desenvolvimiento de la Fase y Respuesta de Fase de un Filtro	31
4.3 Descripción del Escalamiento en Tiempo	33
4.4 Implementación del Escalamiento en Tiempo	34

5. Ecuación	
5.1	Introducción 38
5.2	Diseño de Filtros Campana y Pico de Segundo Orden 39
5.3	Filtro Digital Biquad 40
5.4	Implementación del Ecuación 43
6. Análisis de Resultados	
6.1	Codificador de Voz en Fase 45
6.2	Escalamiento en Tiempo 51
6.3	Ecuación 58
6.4	Aplicación HIPELDA 60
7. Aplicación HIPELDA	
7.1	Esquema 63
7.2	Interfaz 68
7.3	Funcionamiento 69
	Conclusiones 77
	Recomendaciones Finales para Trabajar a Futuro 78
	Bibliografía 79
	Códigos Java y MATLAB 80

Resumen

El trabajo de tesis presente trata acerca de la implementación de una herramienta que modifica señales de audio, en particular, se modifican palabras bisilábicas; la modificación consiste en escalar en el tiempo señales de audio y también ecualizarlas. El objetivo de modificar estas palabras es para que personas que padecen de un ligero déficit de audición, aprendan a identificarlas adecuadamente y como resultado aprendan a pronunciarlas de una manera correcta. Para llevar a cabo dicha implementación, superficialmente se abordaron algunos temas de Neurofisiología; con un poco de detalle, temas de Lingüística; y formalmente, temas como: Transformada de Fourier en Tiempo Reducido, Codificador de Voz en Fase y Filtros Biquad. Finalmente, para manejar la herramienta se desarrolló una aplicación utilizando el lenguaje Java y la plataforma MATLAB. También se desarrolló una interfaz gráfica para el uso de dicha herramienta, la cual únicamente fue implementada con el lenguaje Java. La aplicación recibe el nombre HIPELDA y se puede ejecutar en las plataformas Linux, Mac y Windows. La aplicación sí cumple el objetivo establecido, sin embargo, se podría mejorar en el sentido de ejecutar el procesamiento más rápidamente.

1. Introducción

1.1 Descripción de la Tesis

Los experimentos realizados en modelos neurológicos de aprendizaje perceptual, tanto en humanos como en monos [1], han demostrado que la capacidad de segmentar eventos sucesivos en el flujo de entradas sensoriales, aparentemente con la práctica puede afinarse a lo largo de toda la vida. Los estudios electrofisiológicos de plasticidad inducida por aprendizaje, llevados a cabo en la neocorteza cerebral de los monos, han proporcionado evidencia que mejora los procesos neuronales llevados a cabo con la práctica, tanto en la segmentación temporal como en la discriminación espectral de las entradas sensoriales. Estos estudios han demostrado que la capacidad de un animal adulto para hacer distinciones finas entre las características temporales o espectrales de estímulos complejos, puede ser mejorada mediante un entrenamiento de comportamiento intensivo. Los hallazgos de plasticidad neuronal también se han comprobado en humanos [2].

Paralelamente a estos estudios del aprendizaje, también existen otros estudios que son pioneros en el conocimiento de déficit receptivo en niños con Dificultad en el Aprendizaje del Lenguaje (DAL). Dichos estudios han demostrado que estos niños tienen un “déficit de procesamiento temporal” manifestado por una capacidad limitada para la identificación de algunos elementos fonéticos breves, presentados en contextos específicos del habla, y por un bajo desempeño para identificar o secuenciar estímulos de acústica de corta duración presentados en una sucesión rápida [3]. En consistencia con la hipótesis de un déficit de procesamiento temporal, los niños DAL podrán distinguir las características breves del lenguaje y podrán reconstruir correctamente la secuencia de estímulos acústicos, si los estímulos se presentan de manera más lenta.

Considerando estos hallazgos experimentales, se ha llegado a establecer la hipótesis “El déficit de comprensión de fonemas en niños con DAL puede surgir en una edad temprana, como consecuencia de un aprendizaje perceptual anormal que contribuye a un aprendizaje anormal del lenguaje” [4]. Por ejemplo, un niño podría generar una representación de la información fonética en condiciones no idóneas durante su infancia (por ejemplo, por presencia de enfermedades del oído medio), en donde los estímulos acústicos son sistemáticamente más débiles. Los estudios de plasticidad cortical indican que el déficit de procesamiento temporal como los que surgen en los niños con DAL, podrían superarse si ellos son rehabilitados mediante un régimen de entrenamiento que tome en cuenta los hallazgos de Tallal y colaboradores [5].

Los niños con DAL requieren de cientos de milisegundos entre eventos acústicos para su discriminación, mientras que los niños con edad y nivel de inteligencia iguales únicamente necesitan decenas de milisegundos. El déficit de integración temporal básico interfiere con la habilidad de los niños con DAL, para discriminar características sutiles entre sílabas y palabras que distinguen un fonema de otro. Para ser un poco más específicos, éstos niños no pueden identificar elementos rápidos en el habla humana que tienen una duración de entre 10 a 50 milisegundos [6], el cual es un tiempo crítico ya que muchos elementos fonéticos se producen en ese rango. Por ejemplo, tienen dificultad al identificar entre las sílabas /ba/ y /da/, también entre las sílabas /da/ y /ta/.

El umbral de integración temporal está altamente correlacionado con el grado de deterioro de la percepción del lenguaje en niños preescolares con DAL y con dificultades del aprendizaje de la lectura en esos mismo niños. Con estas investigaciones se ha llegado a postular la posibilidad de rehabilitar a niños con DAL, mediante estímulos del habla modificados. La modificación consistiría fundamentalmente en “lentificar” (hacer más lento) el flujo de información del habla y enfatizar la modulación de las formas de onda del habla para robustecer la representación de elementos fonéticos dentro de la maquinaria de aprendizaje cortical [6].

1.2 Contenido de la Tesis

En este trabajo de tesis la modificación del habla se realiza en dos etapas; la primera es escalar la señal de audio en el dominio temporal y la segunda es ecualizarla. Escalar la señal en tiempo significa que al hacerla más larga ésta se escuchará lentamente y al hacerla más corta ésta se escuchará más rápidamente [6]. El escalamiento se realiza del 50 a 25 % más lento, es decir las señal se alarga con un factor de 1.5 a 1.25. Ecualizar la señal consiste en resaltar o amplificar ciertas bandas de frecuencia; en este trabajo la gama de frecuencias 1-4 kHz se amplifica 20 dB [6].

El Escalamiento en Tiempo (*Time Scaling*) se realiza con el algoritmo Codificador de Voz en Fase (*Phase Vocoder*, PV). La Ecualización (*Equalization*) se lleva a cabo con filtros Biquad de segundo orden. Primero se implementan estas etapas utilizando la aplicación MATLAB, posteriormente se implementan utilizando el lenguaje Java. Se utiliza primero MATLAB porque es muy amigable en su manejo y además se obtienen resultados inmediatos; cada función que se ocupa en MATLAB posteriormente se implementa en Java. Hoy en día existen aplicaciones de dominio libre que pueden modificar señales de audio, por ejemplo el reproductor VLC, el cual se puede instalar en una computadora que utilice cualquiera de las plataformas principales. Sin embargo, la idea de utilizar el lenguaje Java es evitar los derechos de autoría que MATLAB implica. Las páginas y sitios electrónicos de donde se consiguieron los códigos para programar cada función con el lenguaje Java se muestran en la última sección que lleva el nombre de Códigos Java y MATLAB.

El método empleado para escalar la señal e incluso el diseño de los filtros se tomaron del libro “DAFX. Digital Audio Effects” del autor Udo Zölzer, capítulos ocho y dos respectivamente. Además, se utilizan otras fuentes para complementar los conceptos teóricos involucrados. El capítulo ocho del libro describe la teoría y la manera de implementar el Codificador de Voz en Fase, de acuerdo a los trabajos hechos por los autores: M. R. Portnoff, R. E. Crochiere y L. R. Rabiner.

En el capítulo 2 (Antecedentes) de este trabajo, se incluye la explicación de los conceptos fundamentales para entender la motivación e implementación del mismo. En los capítulos siguientes, se desglosa el concepto en cuestión y se muestra su implementación en MATLAB; se utilizan señales de ondas senoidales para visualizar los resultados de una manera sencilla. El archivo del código del concepto Codificador de Voz en Fase se llama **PV.m**, el del concepto Escalamiento en Tiempo se llama **TS.m** y el del concepto Ecuilización se llama **Equa.m**; estos archivos se encuentran en la carpeta nombrada *separados_m* [a]. En el capítulo 6 (Análisis de Resultados), se incluyen una serie de gráficas que muestran los resultados obtenidos con la aplicación MATLAB y el lenguaje Java. De esta manera, se tiene la certeza de que los resultados obtenidos con Java son los mismos a los obtenidos con MATLAB.

Como producto final, se implementa la aplicación de nombre HIPELDA, la cual esta dividida en seis secciones. El archivo principal en MATLAB de nombre **HIPELDA.m** junto con los demás archivos complemento se encuentran en la carpeta nombrada *hipelda_m* [b], y el archivo principal en Java de nombre **HIPELDA.java** junto con los archivos complemento se encuentran en la carpeta nombrada *hipelda_j* [c].

Finalmente, se implementa una interfaz gráfica para el uso de la aplicación. La interfaz se implementa con el API Swing en el entorno de desarrollo integrado Netbeans, el cual es de dominio libre y está hecho para el lenguaje de programación Java. Los archivos de la interfaz gráfica en Java se encuentran en la carpeta nombrada *InterfazHIPELDA_j* [d] y el archivo ejecutable de nombre **InterfazHIPELDA.jar** se encuentra en las carpetas llamadas: *estuche_HIPELDA_w* [e] para la plataforma Windows, *estuche_HIPELDA_l* [f] para la plataforma Linux y *estuche_HIPELDA_m* [g] para la plataforma Mac.

En la aplicación HIPELDA se deben de considerar dos características de uso. En primer lugar, la aplicación solamente trabaja con señales de audio de formato WAV (PCM a 16 bits), grabadas a dos canales. La segunda característica es que entre mayor sea la duración de la señal de audio a modificar, más se tardará la aplicación en escalarla; por ejemplo, si se desea escalar con un factor de 1.5, una señal de audio que dure 12 segundos, la aplicación se tardará 8 segundos aproximadamente; el proceso de ecualización es inmediato. El tiempo de ejecución variará dependiendo de la potencia de la computadora y en qué plataforma se esté ejecutando la aplicación. Debido a estas características de desempeño, lo recomendable es trabajar con pistas de audio que duren menos de 25 segundos. Este tiempo es suficiente para incluir en una pista, por lo menos cinco palabras bisilábicas, las cuales se utilizan en el programa de adiestramiento [5].

2. Antecedentes

2.1 Algunos Conceptos de Neurofisiología

Aprendizaje Perceptual

El término aprendizaje se refiere al proceso por el cual las experiencias modifican el sistema nervioso y, por lo tanto la conducta. El aprendizaje presenta al menos cuatro formas básicas: aprendizaje perceptual, aprendizaje como estímulo respuesta, aprendizaje motor y aprendizaje de relaciones. El aprendizaje perceptual es la capacidad de aprender a reconocer los estímulos como: la percepción, el desempeño, el pensamiento y la planeación. La función principal de este tipo de aprendizaje es la capacidad de identificar y jerarquizar objetos y situaciones. La gente no podrá aprender cómo conducirse en cierta situación, a menos que haya aprendido a reconocerla.

Cada uno de los sistemas sensoriales es capaz de un aprendizaje perceptual. Es posible aprender a reconocer objetos dada su apariencia visual, los sonidos que producen, cómo se sienten o cómo huelen. El aprendizaje perceptual parece lograrse, en primera instancia, debido a cambios en la corteza de asociación sensorial. Es decir, aprender a reconocer los estímulos visuales complejos implica cambios en la corteza de asociación visual, aprender a reconocer estímulos auditivos complejos supone cambios en la corteza de asociación auditiva, etc [7].

Corteza cerebral

La corteza cerebral es el manto de tejido nervioso que cubre la superficie de los hemisferios cerebrales, derecho e izquierdo. Es aquí donde ocurre la percepción, la imaginación, el pensamiento, el juicio y la decisión. A través de estudios se han localizado distintas áreas que realizan procesamientos motores y sensitivos específicos [8]:

Lóbulo Temporal: el lóbulo temporal contiene neuronas que captan cualidades sonoras en la corteza auditiva primaria. También contiene neuronas relacionadas con la comprensión del lenguaje, memoria y aprendizaje.

Lóbulo Frontal: el lóbulo frontal contiene principalmente la corteza motora primaria, en la cual se encuentran las neuronas que controlan los músculos del cuerpo. Está organizada en función de las partes del cuerpo.

Lóbulo Parietal: el lóbulo parietal aloja a la corteza somatosensorial primaria, compuesta por neuronas relacionadas con el tacto, también se organiza en función de las partes del cuerpo.

Lóbulo Occipital: el lóbulo occipital contiene la corteza visual primaria, localizada en la parte posterior, procesa la información visual que llega de la retina.

Plasticidad Neuronal

La plasticidad neuronal es la propiedad que tienen las neuronas de reorganizar sus conexiones sinápticas y de modificar los mecanismos bioquímicos y fisiológicos en respuesta a un estímulo externo (conducta determinada) o un estímulo interno (medicamentos).

Desde la perspectiva neurofuncional esta característica hace que la neurona sea una célula versátil, concebida para una relación dinámica con otras neuronas, por tanto la corteza cerebral tiene un carácter flexible, en cuyo desarrollo no solo intervienen los fenómenos genéticos sino también las vivencias del propio individuo con la posibilidad de cambios en la estructura psíquica e incluso de la función nerviosa. En consecuencia, el cerebro es un órgano activo con capacidad de cambio interno a la voluntad de la persona que puede reforzar la transmisión en las conexiones sinápticas así como establecer y consolidar redes neuronales.

Se sabe que a partir de los mecanismos de plasticidad neuronal también pueden restaurarse funciones neurológicas alteradas mediante diferentes métodos: farmacológicos, quirúrgicos, físicos, comunicativos, ocupacionales y psicosociales. Por ejemplo, se ha comprobado mediante técnicas de neuroimagen que los pacientes obsesivos tienen una mayor actividad en el lóbulo frontal. Después de realizar psicoterapia, estos pacientes mejoran su situación mental. Se ha podido observar, como resultado, que existe una disminución de la actividad en estas áreas cerebrales [9].

Déficit en el Procesamiento Temporal

Uno de los déficit más documentados en la literatura, es el del procesamiento temporal, que se traduce en dificultades para percibir estímulos no verbales presentados con un ritmo muy rápido; es decir, cuando las duraciones de los estímulos son muy cortas, sus logros son significativamente inferiores en comparación a los niños con desarrollo típico. Estos problemas han sido observados en una variedad de tareas auditivas, no sólo con estímulos lingüísticos (sílabas y palabras), sino también con estímulos no lingüísticos como tonos. Por lo tanto, los niños DAL tienen dificultad en el procesamiento de estímulos temporales de algunas decenas de milisegundos y también para diferenciar entre estímulos auditivos cuando la información es breve o cuando se da en una sucesión rápida, pero esta dificultad desaparece cuando el estímulo se presenta más lentamente. Además, de manera general, los niños con DAL procesan y resuelven las actividades lingüísticas y no lingüísticas a una velocidad inferior que los niños con lenguaje típico [10].

2.2 Déficit Auditivo en Personas

El déficit auditivo o sordera, es la dificultad o imposibilidad de captar correctamente el sonido y, principalmente el lenguaje hablado. Esto puede ser un rasgo hereditario o puede ser consecuencia de una enfermedad, medicamento agresivo, accidente, exposición excesiva al ruido o alguna otra razón [11].

Vías Auditivas

Para entender la naturaleza del déficit auditivo, es necesario entender cómo el sistema auditivo funciona. La vía auditiva está compuesta principalmente por 5 entidades: el oído externo, el oído medio, el oído interno, el nervio auditivo y el cerebro. La figura 1 muestra ciertas partes principales.

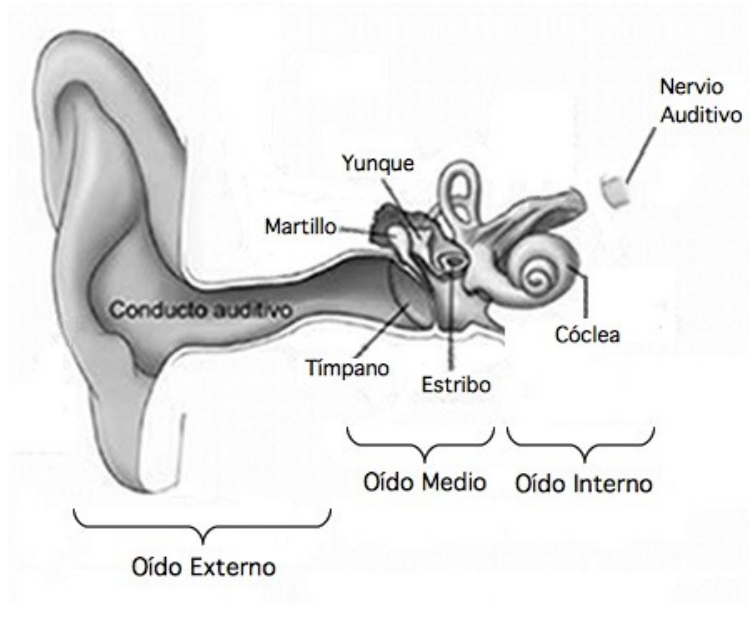


Figura 1. Componentes principales del sistema auditivo humano

Oído Externo: El oído externo consiste de la oreja y el conducto auditivo. Estas estructuras recogen las ondas sonoras y las transmiten al tímpano. Los problemas de audición que aquí se pueden presentar se deben a la acumulación de cera, objetos ajenos, infecciones, o a ciertos defectos de nacimiento que puede obstruir o perforar el tímpano.

Oído Medio: El oído medio incluye todas las estructuras entre el tímpano y el oído interno. Esta cavidad contiene los huesecillos (osículos): martillo, yunque y estribo. Las vibraciones del tímpano se transmiten a través del oído medio por estos tres huesecillos. El movimiento del estribo produce ondas en el medio líquido del oído interno. El sonido se amplifica por la acción mecánica del tímpano el cual es relativo al movimiento del estribo. Los problemas que en esta parte se pueden presentar son: infección, acumulación de líquido, alergias debido al medio ambiente, problemas en los huesecillos (inmovilidad parcial o total), una perforación en el tímpano, y tumores.

Oído Interno: La parte auditiva principal del oído interno se llama cóclea. Se compone de una cápsula densa de hueso que contiene cavidades llenas de fluido. Sobre la superficie que divide la cavidad central se encuentra la membrana basilar y sobre ella se encuentran las células ciliadas, células que tienen penachos de vellos muy finos llamados esterocilios. Las ondas sonoras que causan el movimiento del estribo se transmiten como ondas en el líquido del oído interno. Estas ondas causan vibraciones que desplazan a los esterocilios de las células ciliadas, provocando la generación de impulsos nerviosos (eléctricos). Los problemas que aquí se pueden presentar son: defectos por nacimiento, exposición a niveles altos de sonido, ciertos medicamentos que provocan hipoxia, infecciones, degeneración y golpes en la cabeza.

Nervio Auditivo: Los impulsos eléctricos creados por las células ciliadas se transmiten al cerebro por el nervio auditivo. Los senderos del nervio que llevan al cerebro están encapsulados en un pequeño canal junto con el nervio de equilibrio y el nervio que estimula el movimiento de los músculos faciales. El nervio auditivo se puede dañar por tumores, compresiones en los vasos sanguíneos, infecciones alrededor del cerebro (meningitis), algún golpe, envejecimiento, algún tipo de problema metabólico y defectos al nacer.

Cerebro: Los impulsos eléctricos son reconocidos por la parte superior del cerebro (corteza auditiva). El daño auditivo en el cerebro no está necesariamente limitado a los centros auditivos, éste puede estar asociado a otros problemas cerebrales. Sin embargo, cuando existe un problema cerebral, comúnmente el oído puede detectar el sonido apropiadamente, pero el cerebro puede que no procese las señales auditivas de una manera adecuada. Esto se conoce como desorden del proceso auditivo central [12].

Tipos de déficit auditivo

El déficit auditivo generalmente se divide en conductivo y en neurosensorial. Cuando aparece una combinación de ambos, se le llama mixto. Si se comprueba que la vía auditiva actúa normalmente, entonces el tipo de déficit se le cataloga como trastorno auditivo central.

Conductivo: Las vibraciones sonoras no estimulan debidamente la cóclea una vez atravesada la vía aérea de conducción normal. Cuando el bloqueo de la vía es total, la audiometría mostrará una pérdida máxima de 60 dB en la transmisión aérea. En este caso la persona puede percibir el sonido vía ósea, siempre y cuando el sonido sea controlado adecuadamente.

Neurosensorial: Las células ciliadas han sufrido algún daño o el nervio auditivo sufre de alguna disfunción. La sensación auditiva es menor en intensidad pero también se percibe deformada y alterada. La pérdida auditiva neurosensorial es casi siempre irreversible.

Mixta: Se presenta un obstáculo en la transmisión aérea y alguna disfunción en el oído interno a la vez.

Auditivo Central: No se manifiesta necesariamente en una disminución del grado de sensibilidad auditiva, más bien en una disminución en la comprensión de lo escuchado [13].

Grado del déficit auditivo

Un aspecto importante es saber la magnitud del grado de déficit. La gravedad del déficit auditivo se cataloga de acuerdo a la intensidad adicional que sobrepasa el umbral nominal (justo cuando el individuo detecta el sonido). Las clasificaciones más comunes utilizadas para identificar el grado de déficit auditivo han sido definidas por la sociedad BIAP (Bureau International D'Audiophonologie), en donde han clasificado el déficit en función de percibir tonos de frecuencias 500, 1000 y 2000 Hz sobre el mejor oído. La clasificación es la siguiente:

Normal: por debajo de 20 dB
Leve: 20 – 40 dB
Mediana: 40 – 70 dB
Grave: 70 – 90 dB
Profunda: mayor de 90 dB
Total: superior a 120 dB

Otro elemento que se debe tomar en cuenta para juzgar una disminución auditiva es al evaluar ambos oídos [14].

Repercusiones

Déficit Auditivo Leve: La mayoría de los adultos pueden compensar una pérdida auditiva leve al colocarse frente a frente con quien estén hablando y de esa manera evitan el ruido de los alrededores. Un niño con este nivel de pérdida por lo regular tiene dificultades en la escuela porque no puede escuchar adecuadamente a su profesor. La dificultad de entender bien se debe a que gran parte del sentido de las palabras en el lenguaje está contenido en las consonantes mudas las cuales son de una altura aguda y suave. Estas consonantes son: s, sh, t, p, k y ch. La gente con este déficit necesitará en ambos oídos algo de amplificación para escuchar claramente al estar en grupos o a cierta distancia.

Los niños que padecen este déficit no se les identifica hasta que llegan al primer grado de primaria. Con frecuencia se les considera “de lento aprendizaje”, porque no pueden entender cuando un profesor habla desde cierta distancia, y por lo tanto no responden adecuadamente a lo que se les pregunta. Cuando a estos niños se les coloca un dispositivo de ayuda auditiva para amplificar el sonido, se dan cuenta de que la escuela es más fácil de lo que creían.

Déficit Medio: La gente con esta pérdida auditiva puede escuchar a los demás solamente cuando están muy cerca de la persona, por lo menos a medio metro. Esta gente necesita dispositivos especiales de ayuda auditiva para escuchar incluso los sonidos más ligeros y así poder entender lo que se esté hablando. La mayoría de la veces los adultos con este déficit tendrán dificultades social y laboralmente, por lo que serán necesarios los dispositivos especiales para amplificar el sonido. Si los niños reciben estos dispositivos de ayuda auditiva antes de los cuatro años de edad, por lo regular progresan rápidamente para aprender a hablar. Estos niños pueden asistir a escuelas regulares pero con un apoyo adicional especial.

Déficit Grave: Las personas con una pérdida auditiva severa no pueden percibir el habla, sin importar que tan cerca se encuentren de una persona. Con dispositivos de ayuda auditiva, pueden detectar los sonidos de las vocales, algunas consonantes y ciertas partes de palabras. Con sus ojos, pueden detectar más o menos el 25% de los sonidos de las consonantes. Si leen los labios y escuchan al mismo tiempo, pueden recibir más o menos la mitad de lo que se está diciendo. Los niños con este déficit no aprenderán a hablar de una manera inteligible sin el uso de dispositivos auditivos y ayuda especial. Si los niños con este problema reciben esta ayuda en una edad temprana, tendrán muchas más posibilidades de adquirir el habla, que los que no hayan sido atendidos. Esto no quiere decir que un niño con este problema que haya recibido esta ayuda especial después de los tres años de edad, no aprenderá a hablar, pero sí que el hacerlo será más difícil. Muchos niños que no reciben esta ayuda hasta los seis años de edad nunca desarrollan un habla claro o la habilidad de entender fácilmente palabras habladas.

Déficit Severo: La gente con una pérdida auditiva severa recibe poca o nada de información auditiva proveniente de los dispositivos de ayuda. Afortunadamente, un avance científico llamado implante cóclea ahora permite que muchos individuos puedan oír. Con este dispositivo, una persona sorda puede oír sonidos a niveles de conversación normal.

Los usuarios de implantes cócleares necesitan mucho adiestramiento especial para aprender a identificar los sonidos. Todos los niños con este déficit, ya sea que utilicen un implante cóclea o un dispositivo auditivo, necesitan terapia auditiva y uso continuo de sus aparatos [15].

2.3 Introducción a la Fonología y la Fonética

La fonología y la fonética son las disciplinas encargadas de estudiar los sonidos del habla humana desde una perspectiva lingüística. La fonología describe cómo los sonidos de un lenguaje funcionan, desde un punto de vista abstracto, es decir, cómo el cerebro humano percibe los sonidos. La fonética describe los sonidos del habla humana a través de sus propiedades físicas. Sus principales ramas son la fonética experimental, la fonética articulada y la fonética acústica.

Las unidades que se emplean para estudiar la fonología es el Fonema y para estudiar la fonética es el Fono. Por ejemplo, el elemento que ortográficamente es transcrito como <n> aparece en las siguientes palabras: **entero**, **lengua**, **encima**, **sano**, **hinchar**. La percepción del ser humano es que en todos estos casos <n> siempre suena igual, sin embargo, la realidad es que no es así. El cerebro humano no distingue la diferencia. La percepción mental que se tiene de la <n> se le llama fonema y se expresa entre barras diagonales /n/. Los sonidos reales de cada caso se llaman fonos y se expresan en corchetes [n].

Los fonemas se clasifican como vocales y consonantes. Las vocales son emisiones de voz que no son obstruidas a lo largo del aparato fonador, el cual es el conjunto de órganos que intervienen en la articulación del lenguaje. Mientras que las consonantes son emisiones de voz que son un tanto obstruidas a través del mismo aparato. Además, en español todas las sílabas necesitan por lo menos de una vocal, ya que sin ellas las consonantes no se pueden pronunciar.

Elementos Articulatorios

En la pronunciación participan bastantes elementos fisiológicos que pertenecen al aparato fonador. Se deben tener en cuenta porque sirven para describir los fonemas. Estos son:

- Cuerdas Vocales: provocan que una consonante sea sonora (vibra) o sorda (no vibra).
- Lengua: roza partes de la cavidad bucal como el velo del paladar, el paladar, los alvéolos y/o los dientes.
- Labios
- Cavidad nasal

Punto y Modo de Articulación

Para clasificar los fonemas, se utilizan dos criterios principales:

- Punto de Articulación: el lugar en la boca donde se pronuncia un determinado fonema.
- Modo de Articulación: los procesos que se ponen en marcha para pronunciar un fonema.

Vocales

Las cinco vocales del español se clasifican de la siguiente manera:

- /i/ vocal inicial cerrada
- /u/ vocal final cerrada
- /e/ vocal inicial central media
- /o/ vocal central final media
- /a/ vocal central abierta

Además, las vocales /a/, /e/ y /o/ son llamadas vocales fuertes, mientras que /i/ y /u/ son las débiles.

Consonantes

Las consonantes se pueden clasificar respecto al punto y modo de articulación. Respecto al punto de articulación se tienen los siguientes criterios:

- Bilabial: participación de los dos labios, por ejemplo: /b/
- Labiodental: labio inferior con dientes superiores, /f/
- Linguodental: la lengua toca la parte trasera de los dientes superiores, /d/
- Linguoalveolar: la lengua toca los alvéolos superiores, /s/
- Linguopalatal: la lengua toca el paladar, /t/
- Linguovelar: la lengua toca el velo, /k/

Por lo que respecta al modo de articulación, las consonantes se catalogan por los siguientes rasgos:

- Oclusivas: se produce una explosión, /p/
- Fricativas: se produce un roce, /f/
- Africadas: es la combinación de las dos anteriores, /tʃ/
- Nasales: parte del aire sale por la nariz, /m/
- Laterales: la lengua sale por los laterales de la boca, /l/
- Vibrantes: se producen por la vibración de la lengua, /r/

Además de los rasgos anteriores, es necesario mencionar que existe uno más que es bastante importante. Se trata de la sonoridad o no sonoridad, que se debe a la vibración o no vibración de las cuerdas vocales. Todas las vocales son sonoras y entre las consonantes algunas son sonoras y otras son sordas (mudas) [16].

Fonética Experimental

La fonética experimental emplea métodos de investigación utilizados en física, fisiología y psicología. Este tipo de estudio analiza las características físicas de los sonidos del habla humana. Se utilizan espectrogramas para visualizar dichas características.

Gran parte del trabajo en fonética experimental se ha dirigido a obtener descripciones más precisas de los sonidos que caracterizan a lenguajes diferentes [17]. Los datos obtenidos de este tipo de estudio muestran que la mayoría de los sonidos, especialmente las vocales, están constituidos por combinaciones de pocas frecuencias, a los que se les llaman formantes, que permiten al oído reconocer dicho sonido. Los formantes vienen siendo las componentes de la señal, o de igual forma son los armónicos de la frecuencia fundamental.

La acústica de las vocales puede visualizarse haciendo uso de espectrogramas, que muestran la energía acústica en cada frecuencia y cómo ésta cambia con el tiempo. Las vocales corresponden a ondas complejas periódicas de amplitud relativamente alta en comparación con las consonantes. Los formantes significativos que describen las vocales se denominan F1 y F2. Siendo F1 la frecuencia fundamental y F2 el primer armónico. Las vocales se suelen ordenar decrecientemente, desde la más aguda a la más grave. Por tanto, se tiene la siguiente secuencia “i e a o u”. La representación de esta secuencia en tiempo y su correspondiente espectrograma donde se muestran los formantes se pueden observar en la figura 2. La secuencia fue producida por una voz de mujer adulta [18].

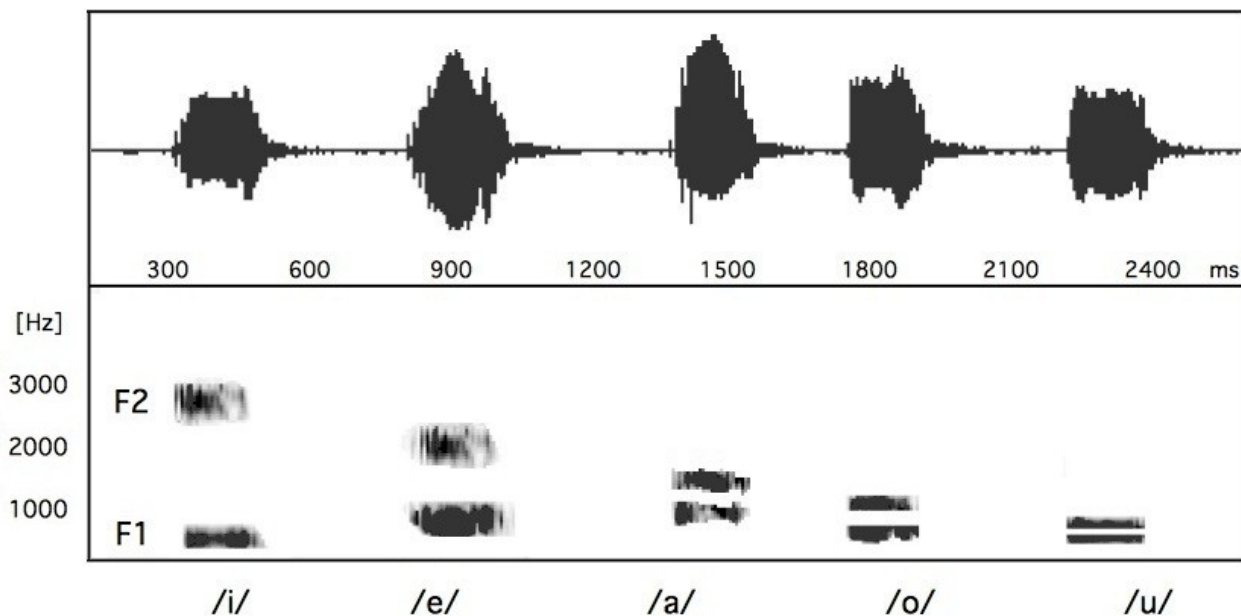


Figura 2. Vocales

La tabla 1 contiene las frecuencia de los formantes (valores redondeados) producidas por las vocales [19].

	F1 [Hz]	F2 [Hz]
/i/	240	2830
/e/	490	2250
/a/	660	1170
/o/	510	980
/u/	240	630

Tabla 1. Frecuencias de los formantes de las vocales

Las consonantes oclusivas se origina por un cierre (oclusión) en algún lugar del tracto vocal. Por tanto, la característica acústica durante esa fase es la ausencia de sonido. Aunque en realidad esto nada más se aplica para las oclusivas sordas, ya que para las oclusivas sonoras las cuerdas vocales vibran durante las fase de oclusión, generando una banda de frecuencias muy bajas, que se conoce como barra de sonoridad.

Ya sea una oclusión sorda o sonora, la presión de la columna de aire proveniente de los pulmones la rompe, originando un ruido característico conocido como “explosión”. Ésta, acústicamente, se manifiesta como una barra vertical que ocupa más o menos todo el espectro frecuencial, y recibe el nombre de Barra de Explosión (figura 3) [20].

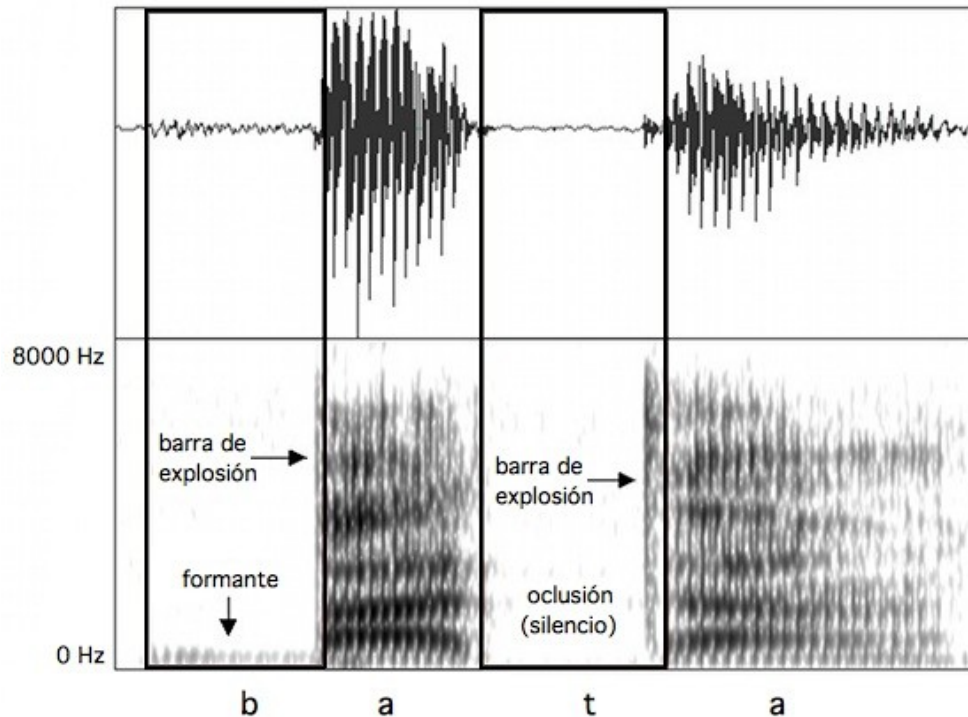


Figura 3. Señal de tiempo y espectrograma del efecto de las consonantes oclusivas

La figura 3 muestra la diferencia de comportamiento entre dos consonantes oclusivas, una sonora y otra sorda. La primera muestra la formante (barra de sonoridad) y su correspondiente barra de explosión. La segunda muestra el silencio que se produce debido a la oclusión y también su barra de explosión [21].

Las consonantes /p/, /t/ y /k/ se consideran oclusivas sordas. Estas consonantes tienen características acústicas muy diferentes a las vocales. La diferencia radica en que no tienen ruido de vibración de cuerdas vocales, entonces no tienen formantes o armónicos. Se caracterizan por una progresiva acumulación y repentina explosión de energía. La serie “pe ta ka” representa el comportamiento de estas consonantes (figura 4).

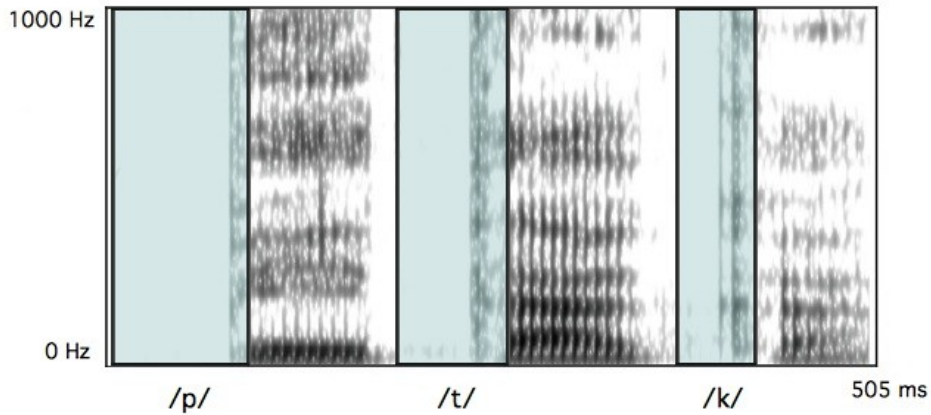


Figura 4. Consonantes Oclusivas Sordas

Un criterio con el que estas consonantes pueden ser descritas y diferenciadas acústicamente entre sí, es a través de la duración relativa de sus componentes y la diferencia de duración entre ellas. La consonante oclusiva sorda /p/ es la que más tiempo toma en completarse. Un ejemplar típico de esta consonante puede tomar 80 milisegundos aproximadamente (figura 5).

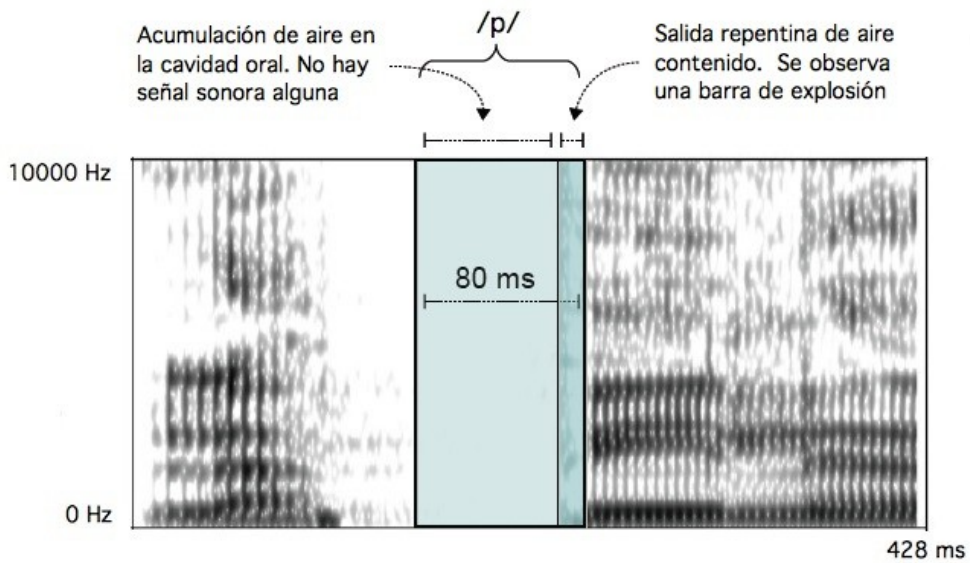


Figura 5. Consonante /p/

La consonante oclusiva sorda /t/ toma menos tiempo en completarse, 60 milisegundos aproximadamente (figura 6).

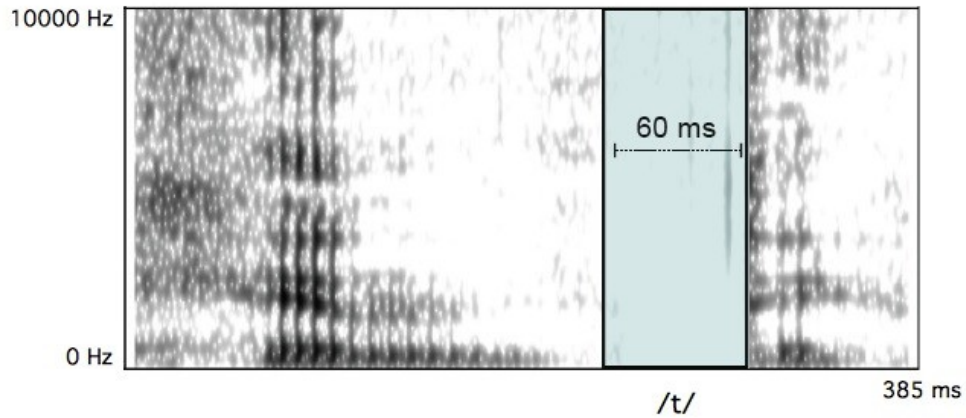


Figura 6. Consonante /t/

La consonante /k/ es la más breve de este tipo y toma en completarse 40 milisegundos aproximadamente (figura 7).

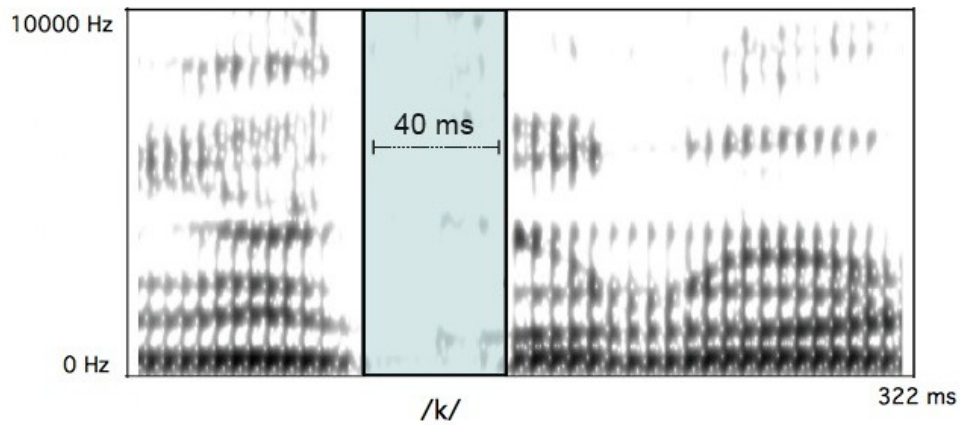


Figura 7. Consonante /k/

Las consonantes /b/, /d/ y /g/ se consideran oclusivas sonoras. Estas consonantes se diferencian de las sordas, básicamente por contar con la presencia de una barra horizontal de sonoridad y por su duración (son más breves). También presentan barras de explosión. La serie “ba da ga” representa las consonantes oclusivas sonoras (figura 8).

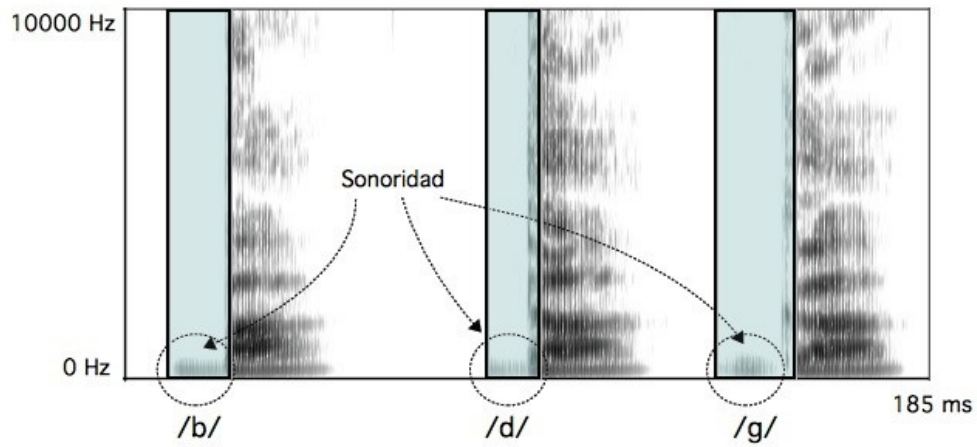


Figura 8. Consonantes Oclusivas Sonoras

Las figuras 9, 10 y 11 muestran las consonantes oclusivas sonoras con su respectivo tiempo que tarda en producirse cada una. Además de mostrar su correspondiente barra de sonoridad [22].

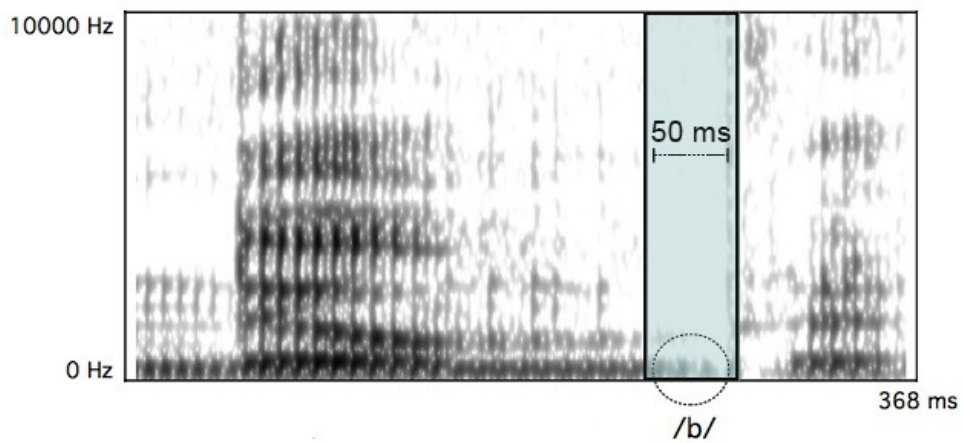


Figura 9. Consonante /b/

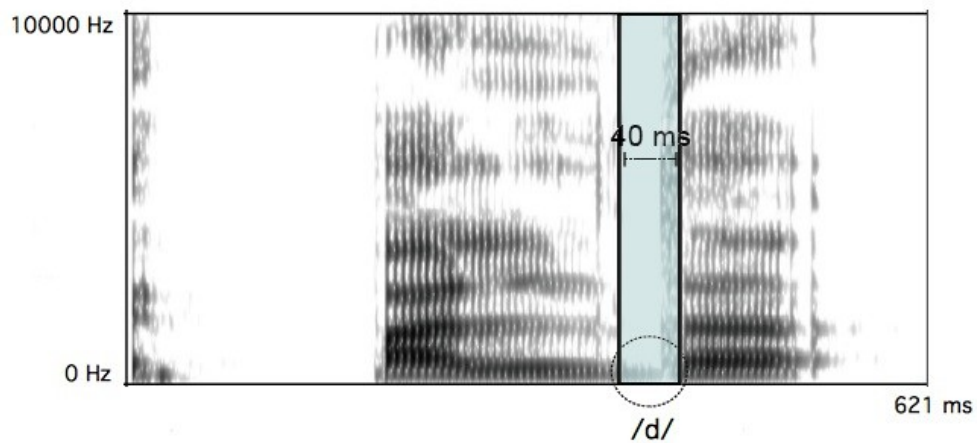


Figura 10. Consonante /d/

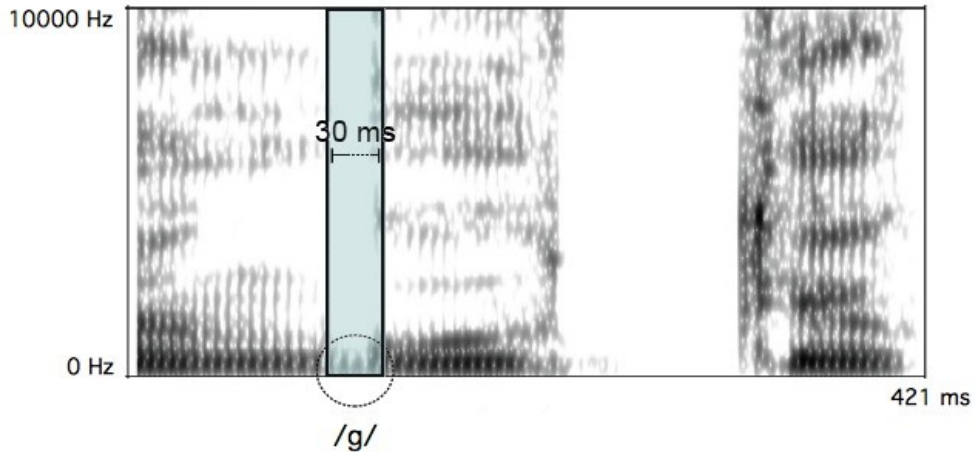


Figura 11. Consonante /g/

2.4 Procesamiento en el Dominio Tiempo contra Frecuencia

Una manera de modificar señales sonoras se logra al representarlas en dos dimensiones. Primero, la señal se representa en el dominio tiempo contra frecuencia, después se modifica esa representación y finalmente se reconstruye una señal resultante. Se puede decir entonces que para realizar un efecto de audio digital, basado en la representación tiempo contra frecuencia, se siguen tres pasos: un análisis (representación del sonido), una transformación (de la representación) y una síntesis (pasar la representación nueva a una señal de audio).

El esquema de análisis y síntesis de señales de audio recibe el termino Codificador de Voz en Fase (*Phase Vocoder*, PV). Este esquema comúnmente se implementa con la Transformada de Fourier de Tiempo Reducido (*Short Time Fourier Transform*, STFT). A diferencia de la Transformada de Fourier (FT), la cual toma toda la señal completa y la transforma toda a la vez, la STFT toma segmentos pequeños de la señal y los transforma uno por uno.

Las etapas del esquema análisis y síntesis son las siguientes (figura 12):

- a) Análisis: la señal de entrada es ponderada por una ventana deslizante de tamaño N , obteniendo así una cantidad de segmentos de la señal original.
- b) Transformación: los espectros de cada segmento son modificados para hacer algún tipo de efecto en el audio.
- c) Síntesis: se calcula la Transformada Rápida de Fourier Inversa (IFFT) de cada segmento y finalmente todos los segmentos son traslapados para así obtener una señal de salida [23].

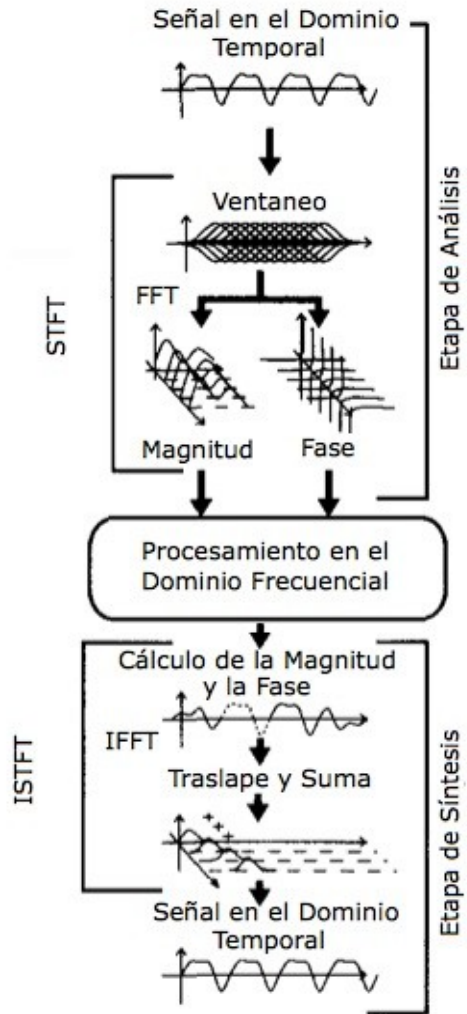


Figura 12. Esquema Codificador de Voz en Fase

3. Codificador de Voz en Fase

3.1 Transformada de Fourier de Tiempo Reducido (STFT)

La STFT comúnmente se utiliza como una herramienta para el análisis, la modificación y la síntesis de señales de voz y audio. Por lo que esta herramienta se puede utilizar para implementar un Codificador de Voz en Fase.

La Transformada Discreta de Fourier (DFT) se puede calcular con la siguiente ecuación:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad k = 0, 1, \dots, N-1 \quad (1)$$

Donde $W_N = e^{-j2\pi/N}$

La versión rápida de la formula anterior se llama la Transformada Rápida de Fourier (FFT). La FFT toma N (potencia de 2) muestras consecutivas de la señal $x(n)$ y desempeña una operación matemática para arrojar N muestras $X(k)$, las cuales representan el espectro de la señal $x(n)$. Cada muestra de frecuencia k a la vez está representada con una resolución de N muestras.

A diferencia de la DFT, la STFT en el dominio discreto está dada de la siguiente manera:

$$X(n, k) = \sum_{m=-\infty}^{\infty} x(m) h(n-m) W_N^{mk}, \quad k = 0, 1, \dots, N-1 \quad (2)$$

Donde $W_N = e^{-j2\pi/N}$

A diferencia de la ecuación (1), en la ecuación (2) la señal $x(n)$ es ponderada por una ventana h cierto número de veces, y a cada ponderación se le aplica la DFT (figura 13). Mientras que en la ecuación (1) se aplica la DFT a la señal $x(n)$ completa, toda a la vez.

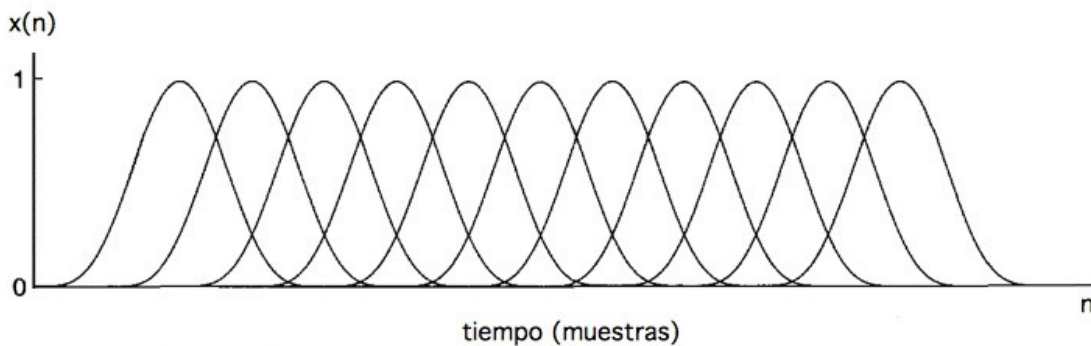


Figura 13. Ventaneo de la señal $x(n)$

Cada una de estas ponderaciones (tramos) es transformada utilizando la FFT. Dependiendo del tamaño total de la señal, el tamaño de la ventana y cada cuando avance la misma, se obtendrá cierto número de tramos, los cuales representarán un espectro cada uno.

El lado derecho de la ecuación (1) muestra que el índice n lleva el control de las muestras del tiempo de la señal $x(n)$, y estas son tomadas consecutivamente. Por otro lado, N indica el tamaño (potencia de 2) de la señal transformada y es un número inmediato que rebasa el tamaño de la señal.

Con el fin de programar la ecuación (2), se tiene que definir una variable que indique cada cuando la señal $x(n)$ es ponderada (ventaneada), es decir, cada cuando la ventana avanza al ponderar la señal. A esta variable se le llama Ra y no se especifica en la ecuación, únicamente se utiliza al programar dicha ecuación. El tamaño de esta variable es un número de potencia 2 y en la literatura este valor comúnmente se fija a 512. El lado izquierdo de la ecuación (2), ahora muestra que n es el índice que lleva el control del número de veces que la señal será ponderada. El lado derecho de la ecuación muestra que ahora N es el tamaño de la ventana y m es el índice que lleva el control de las muestras del tiempo del tramo. El tamaño de N también es un número de potencia de 2 (la FFT lo requiere) y este comúnmente se fija con una resolución de 2048 muestras.

Se puede concluir que la señal $x(n)$ al transformarla utilizando la STFT, será ponderada con una ventana de N muestras, n número de veces, cada Ra muestras. En donde ahora m lleva el control de las muestras del tiempo, pero ahora del tramo, y estas son tomadas consecutivamente de 0 a N (aunque en la ecuación muestre de menos infinito a infinito). Y ahora el lado izquierdo de la ecuación (2) depende del índice n .

Por otro lado, $X(n, k)$ es un número complejo y se puede representar con su magnitud $|X(n, k)|$ y su fase $\varphi(n, k)$, teniendo así:

$$X(n, k) = X_R(n, k) + jX_I(n, k) = |X(n, k)| e^{j\varphi(n, k)} \quad (3)$$

El índice de frecuencia k corresponde al intervalo $0 \leq k \leq N-1$ y el índice n corresponde al tiempo, el cual depende del tamaño total de la señal, el tamaño de la ventana y cada cuando avanza la ventana. Como un ejemplo sencillo, la figura 14 muestra tres ponderaciones arbitrarias de la señal $x(n)$; el eje de las x representa el tiempo en muestras y el eje de las y la amplitud.

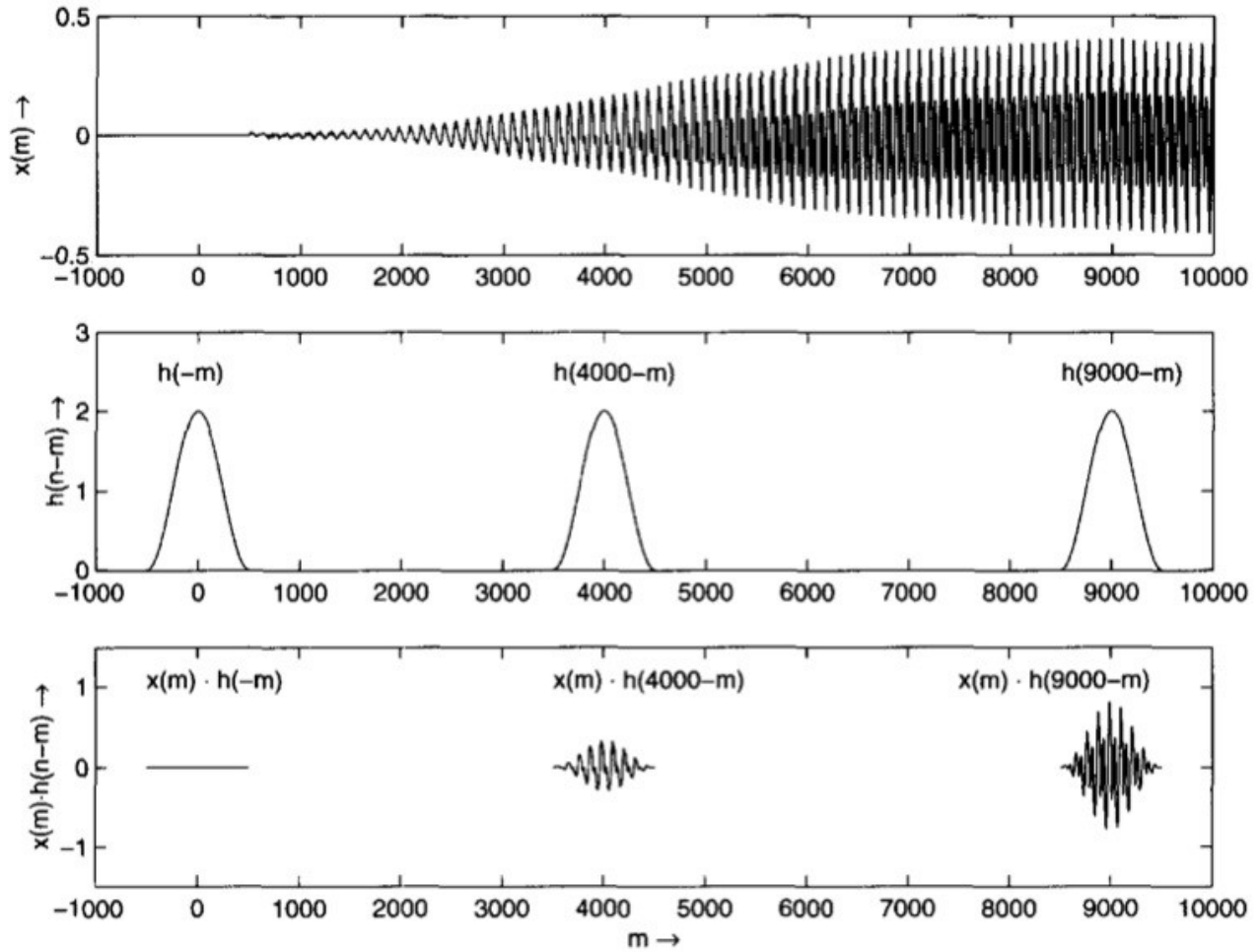


Figura 14. Tres ponderaciones arbitrarias de la señal $x(n)$

La gráfica superior de la figura 14 muestra la señal inicial total $x(n)$. La gráfica media muestra las posiciones de las ventanas con que es ponderada la señal; en este caso los índices son $n = 0, 4000$ y 9000 . La gráfica inferior muestra los tramos obtenidos al ponderar la señal. Posteriormente, cada tramo es transformado con la FFT arrojando un espectro por cada tramo, así como lo muestra la figura 15 en donde el eje de las x representa el tiempo en muestras y el eje de las y la amplitud. [24].

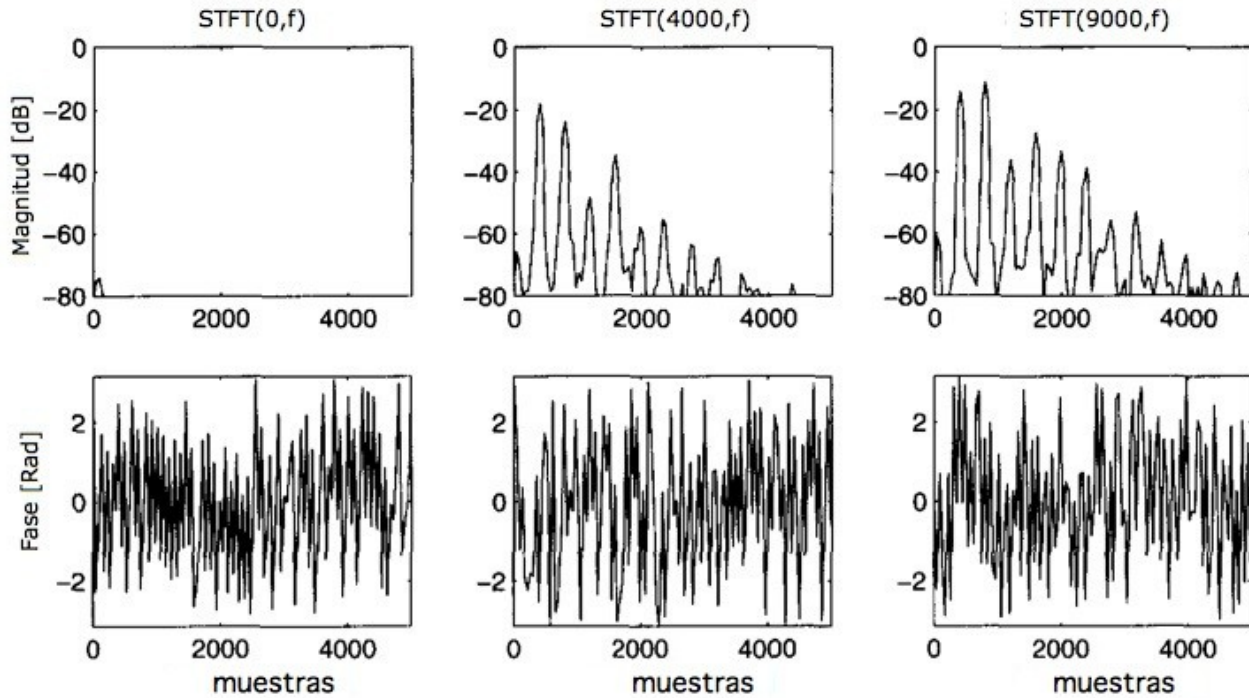


Figura 15. Magnitud y fase de cada tramo

La parte superior de la figura 15 muestra la magnitud obtenida después de transformar cada tramo. La parte inferior muestra su correspondiente respuesta en fase.

3.2 Descripción del Codificador de Voz en Fase

Un Codificador de Voz en Fase se utiliza para modificar una señal en el dominio frecuencial. El PV utiliza la información de la fase para escalar la señal en tiempo o en altura. Comúnmente se utilizan dos modelos teóricos para interpretar un PV, el primero se llama Suma de un Banco de Filtros y el segundo se llama Análisis y Síntesis por Bloques. En este trabajo se implementa el segundo. Una vez que se transforma el tramo de señal con la FFT, se mantienen fijos los valores de la amplitud y únicamente se manipulan los valores de la fase.

En la etapa de análisis, la señal inicial $x(n)$ es ponderada consecutivamente cada R_a muestras con una ventana de N muestras. Es decir, se avanza cada R_a muestras y por cada avance, se tienen N muestras. Al mismo tiempo que se pondera la señal, se calcula el espectro correspondiente de cada ponderación (figura 16). Los tamaños N y R_a son 2048 y 512, respectivamente.

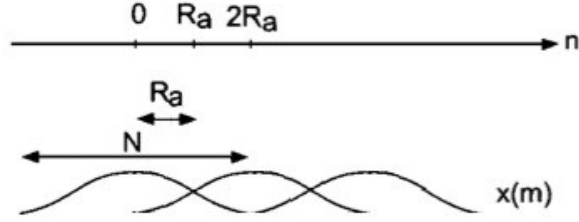


Figura 16. Etapa de análisis

El índice del tiempo n de la ecuación (2) ahora se cambia por sR_a , teniendo así:

$$X(sR_a, k) = \sum_{m=-\infty}^{\infty} x(m) h(sR_a - m) W_N^{mk} \quad (4)$$

La ecuación (4) se puede manipular para así obtener:

$$= W_N^{sR_a k} \sum_{m=-\infty}^{\infty} x(m) h(sR_a - m) W_N^{-(sR_a - m)k} \quad (5)$$

$$X(sR_a, k) = W_N^{sR_a k} \tilde{X}(sR_a, k) \quad (6)$$

La ecuación (6) se puede expresar de la siguiente manera:

$$\tilde{X}(sR_a, k) = W_N^{-sR_a k} X(sR_a, k) \quad (7)$$

La ecuación (7) dice que el espectro de cada tramo es modulado por un factor $W_N^{-sR_a k}$ y éste variará de acuerdo a los valores del tramo. Entonces, la señal será analizada s número de veces, cada vez que se avance R_a muestras.

Al llegar a este punto, a cada espectro obtenido $X(sR_a, k)$ se le puede hacer algún tipo de modificación, obteniendo así una serie de espectros modificados $Y(sR_s, k)$. Así como se utiliza una ventana de ponderación en la etapa de análisis también se utiliza una en la etapa de síntesis, la cual se representa como R_s y también es de tamaño N , así que tanto R_a como R_s son del mismo tamaño. La ventana de síntesis evitará que se produzca algún tipo de distorsión que se pueda originar por el efecto convolución circular. Finalmente, con los espectros $Y(sR_s, k)$ se sintetiza una señal resultante $y(n)$.

El algoritmo para sintetizar la señal resultante, el cual involucra el concepto de la IFFT, está dado por la siguiente ecuación:

$$y(n) = \sum_{s=-\infty}^{\infty} f(n - sR_s) y_s(n - sR_s) \quad (8)$$

$$\text{Donde } y_s(n - sR_s) = \sum_{k=0}^{N-1} Y(sR_s, k) W_N^{-nk} \quad (9)$$

Lo que la ecuación (8) nos dice es que se calcula la transformada inversa de cada espectro modificado $Y(sR_s, k)$, obteniendo así una serie de señales $y_s(n)$. Estos segmentos son ponderados por una ventana también de tamaño N muestras denotada como $f(n)$. Estas ponderaciones son trasladadas y sumadas cada s vez (aunque en la ecuación (8) muestre una suma infinita), cada R_s muestras para así conseguir una señal resultante $y(n)$ (figura 17) [25].

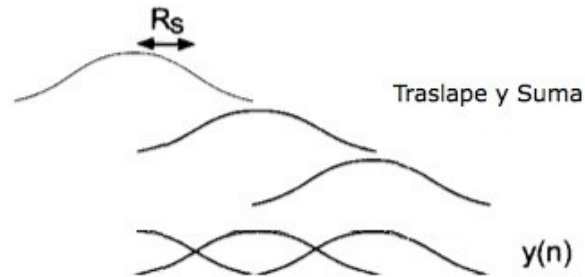


Figura 17. Etapa de síntesis

3.3 Implementación del Codificador de Voz en Fase

Como se menciona en la sección previa, se puede implementar un PV utilizando dos enfoques, el primero es un Banco de Filtros y el segundo es un Cálculo Directo de la FFT e IFFT. En este trabajo se implementa el segundo enfoque, en el cual se calculan valores de magnitud y fase en tramos cortos y únicamente se manipula la fase, lo que disminuye el tiempo de cálculo, razón por la cual este algoritmo es utilizado frecuentemente. Independientemente del enfoque que se utilice para reconstruir una señal, este debe garantizar la reconstrucción perfecta de la misma. Otra cualidad importante es implementar efectos en la señal de audio de tal manera que ésta no se distorsione de una manera considerable. Por ejemplo, si al escalar en tiempo una señal de audio ésta cambia su altura y/o timbre, siendo evidente al escucharla e incluso al visualizar su espectro, entonces el algoritmo implementado no será de utilidad.

Una representación de la señal en tiempo contra frecuencia se puede ver como una serie de FFTs trasladadas. Debido a que se puede obtener una señal del dominio frecuencial al dominio temporal utilizando la IFFT, entonces se puede reconstruir un sonido trasladando y sumando consecutivamente las IFFTs de cada tramo (figura 18).

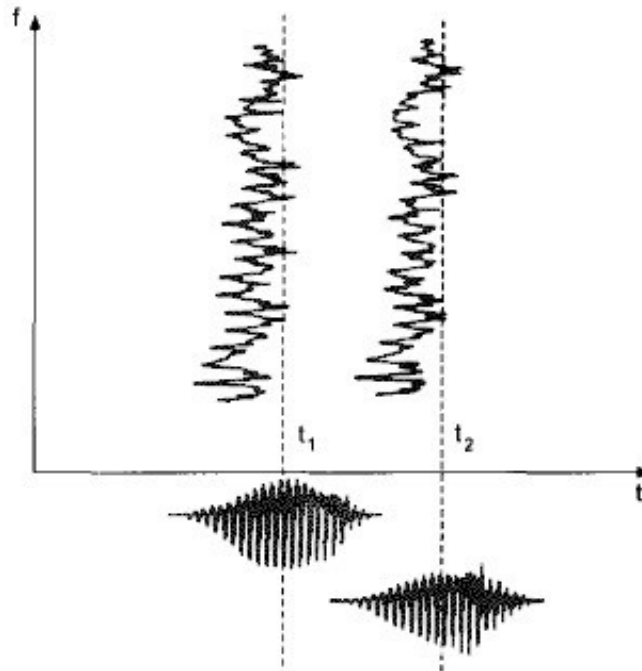


Figura 18. Dos tramos de la señal original en el dominio temporal (inferior) y frecuencial (superior)

El siguiente código en MATLAB muestra la implementación del PV utilizando el enfoque Cálculo Directo de la FFT e IFFT [27]. Este código se encuentra en el archivo `PV.m` en la carpeta `separados_m` [a].

```

1. close all
2. clear all
3. clc;
4.
5. fs = 44100;
6. f = 500;
7. T = 1/fs;
8. t = 0:T:1-T;
9.
10. x = sin(2*pi*f*t);
11. subplot(2,2,1);
12. plot(x(1:600),'k');grid
13. title('a')
14. xlabel('Tiempo (Muestras)');
15. ylabel('Amplitud');
16.
17. N = 2048;
18. Rs = N/4;
19. Ra = Rs;
20. w1 = hann(N);
21. w2 = w1;
22. L = length(x);
23. x_in = [zeros(N,1); x'; zeros(N,1)] / max(abs(x));
24.
25. x_out = zeros(length(x_in),1);
26. grain = zeros(N,1);
27.
28. s = 0;
29. s_times = (L+N) / Ra;

```

```

30.     sRa      = 0;
31.     sRs      = 0;
32.
33.     for s=1:s_times
34.
35.         grain = x_in(sRa+1:sRa+N).*w1;
36.         X      = fft(grain);
37.         X_abs  = abs(X);
38.         phi    = angle(X);
39.
40.         Y      = (X_abs.* exp(1i*phi));
41.         grain = real(ifft(Y)).*w2;
42.         x_out(sRs+1:sRs+N) = x_out(sRs+1:sRs+N) + grain;
43.
44.         sRa    = s * Ra;
45.         sRs    = s * Rs;
46.
47.     end
48.
49.     x_out = x_out(N+1:length(x_out)-N) / max(abs(x_out));
50.     subplot(2,2,3);
51.     plot(x_out(1:600),'k');grid
52.     title('c')
53.     xlabel('Tiempo (Muestras)')
54.     ylabel('Amplitud')
55.
56.     L = length(x);
57.     NFFT = 2^nextpow2(L);
58.     X = fft(x,NFFT)/L;
59.     f = fs/2*linspace(0,1,NFFT/2+1);
60.
61.     subplot(2,2,2);
62.     plot(f,2*abs(X(1:NFFT/2+1)),'k');grid
63.     title('b')
64.     xlabel('Frecuencia (Hz)')
65.     ylabel('Amplitud Normalizada')
66.     axis([470 530 0 1])
67.
68.     L = length(x_out);
69.     NFFT = 2^nextpow2(L);
70.     X_OUT = fft(x_out,NFFT)/L;
71.     f = fs/2*linspace(0,1,NFFT/2+1);
72.
73.     subplot(2,2,4);
74.     plot(f,2*abs(X_OUT(1:NFFT/2+1)),'k');grid
75.     title('d')
76.     xlabel('Frecuencia (Hz)')
77.     ylabel('Amplitud Normalizada')
78.     axis([470 530 0 1])

```

Como ejemplo, se utiliza una señal senoidal de $f = 500$ Hz, muestreada a $F_s = 44,100$ Hz por segundo y de duración $t = 1$ segundo. Se utilizan dos ventanas de ponderación tipo Hanning de tamaño $N = 2048$. La ventana de análisis es w_1 y la ventana de síntesis es w_2 . El número de muestras a avanzar en cada ponderación de análisis R_a y síntesis R_s es de $1/4$ del tamaño de la ventana. El número de veces que la señal será ponderada se indica con el índice s de acuerdo a la ecuación (4), en el código éste índice se indica como **s_times** y es calculado como lo muestra la línea 29.

El ciclo **for** abarca las etapas de análisis y síntesis. La parte de análisis se muestra de la línea 35 a la 38. La parte de síntesis se muestra de la línea 38 a la 40. Si hubiera algún tipo de modificación a la señal, esta se realizaría en la línea 40, justo antes de sintetiza la señal de salida.

4. Escalamiento en Tiempo

4.1 Introducción

El Escalamiento en Tiempo es el proceso de cambiar la rapidez o duración de una señal de audio sin afectar su altura ni timbre. Este proceso se utiliza por ejemplo, para ajustar el tiempo de duración de grabaciones o para hacer algún tipo de efecto instrumental.

El Escalamiento en Tiempo se puede lograr utilizando el Codificador de Voz en Fase. Cabe recordar que el PV se puede implementar de dos maneras, una es utilizando un banco de filtros en donde las amplitudes y frecuencias varían con el tiempo y la otra es utilizando una transformada de Fourier deslizable, como se describe en la sección 3.2 (Descripción del Codificador de Voz en Fase), en donde la magnitud se mantiene y la fase se modifica.

Si se quiere alargar la señal, hay que buscar una manera de propagar (alargar) los espectros en el dominio del tiempo contra frecuencia obtenidos con la ecuación (7). De esta manera, se pueden calcular valores nuevos de fase, es decir, más valores de fase para así calcular más frecuencias (instantáneas). Finalmente, con la ayuda de la Transformada Inversa de Fourier se pueden reconstruir nuevos sonidos en el tiempo. Para llevar a cabo todo esto, se necesita un algoritmo que desenvuelva la fase en la etapa de análisis y otro para que la envuelva (una vez manipulada) en la etapa de síntesis [28].

Al implementar el PV en la sección 3.3 (Implementación del Codificador de Voz en Fase), no hubo una etapa de procesamiento en donde los valores de fase se manipularan. Entonces, en la etapa de síntesis los valores de fase fueron utilizados tal cual para reconstruir la señal resultante, obteniendo así exactamente la misma señal inicial. Por otro lado, la ventana de ponderación en la etapa de análisis funciona como un filtro, el cual tiene su correspondiente respuesta en fase. Los valores de esta respuesta en fase deben ser cero para que los valores de fase de cada tramo, no sean alterados. De esta manera, la señal original no será distorsionada por la ventana de ponderación.

4.2 Desenvolvimiento de la Fase y Respuesta de Fase de un Filtro

Desenvolvimiento de la Fase

Los valores de fase de una señal comúnmente vienen confinados en el intervalo $(-\pi, \pi)$. A esta característica se le llama Fase Envuelta. Para poder manipular la fase es necesario desenvolverla. Si se trata de una señal senoidal, al desenvolver su fase, ésta resulta en una línea recta con cierta pendiente de inclinación (figura 19) [29].

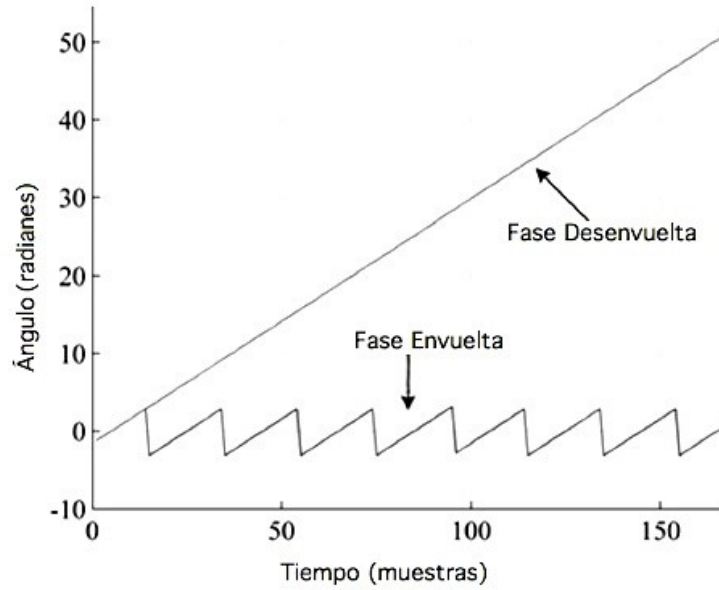


Figura 19. Desenvolvimiento de la fase

Respuesta de Fase de un Filtro

Un filtro de Fase Cero se caracteriza por tener una respuesta impulsiva y simétrica alrededor del origen. La forma de onda del lado negativo es un reflejo del lado positivo (figura 19, lado izquierdo). Al calcular la Transformada de Fourier de esta forma de onda simétrica, la fase es completamente cero (figura 20, lado derecho).

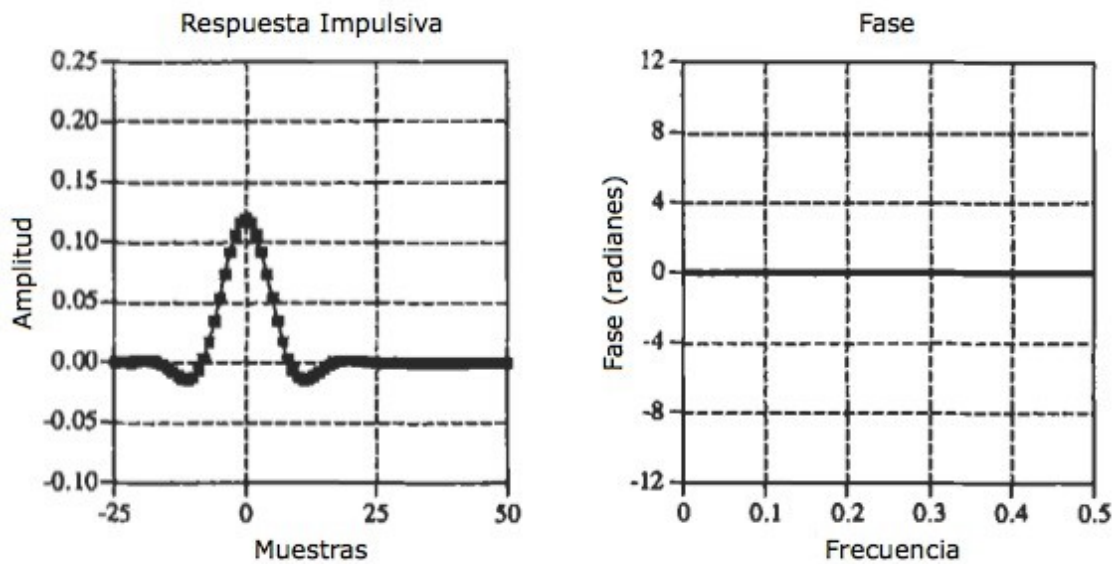


Figura 20. Filtro de Fase Cero

La desventaja del filtro de fase cero es que requiere utilizar valores de índice negativos, los cuales lo puede volver un poco inconveniente. El filtro de Fase Lineal es una manera de resolver en forma práctica la característica de operación no causal del filtro de fase cero. La forma de onda de este filtro es idéntica a la mostrada en la figura anterior. La diferencia radican en que ahora la forma de onda es recorrida hacia la derecha de tal manera que únicamente se utilizan muestras de números positivos (figura 21, lado izquierdo). Debido a este cambio de posición, al calcular la Transformada de Fourier crea como resultado una fase, y ésta se representa con una línea recta (figura 21, lado derecho). La pendiente de esta línea recta es directamente proporcional al cambio de posición de la respuesta impulsiva.

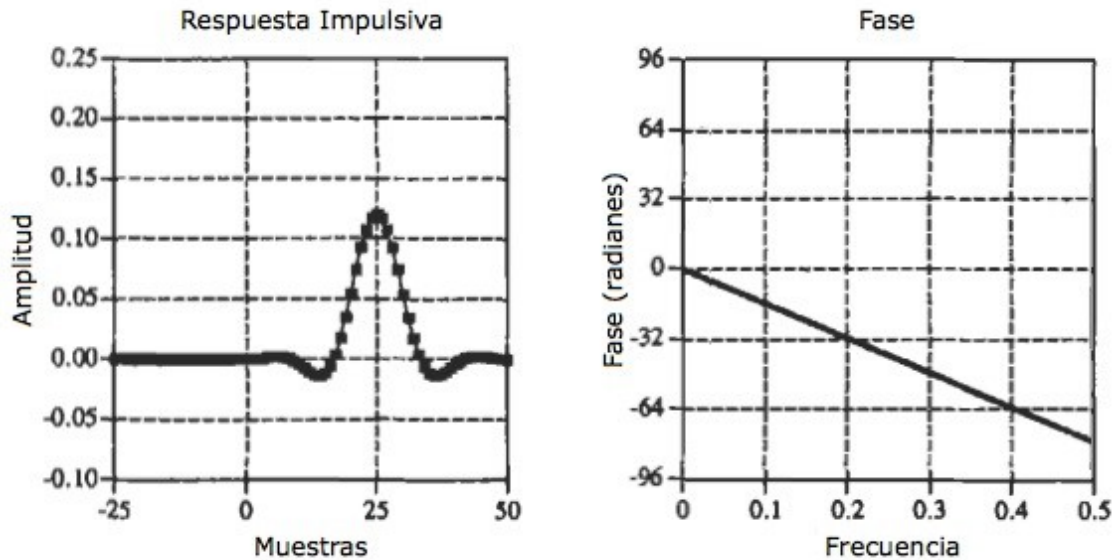


Figura 21. Filtro de Fase Lineal

Dado que el corrimiento de la respuesta impulsiva no hace más que producir el mismo corrimiento en la salida de la señal que se esté filtrando, el filtro de fase lineal es equivalente al filtro de fase cero y se puede utilizar indistintamente en algunas aplicaciones [30].

4.3 Descripción del Escalamiento en Tiempo

El objetivo de un algoritmo eficiente para escalar una señal de audio en tiempo es, encontrar la manera de preservar su altura y/o timbre, sin que haya un cambio considerable al escuchar o visualizar dicha señal. En el enfoque Cálculo Directo de la FFT e IFFT, lo que se hace es mantener los valores de la magnitud fijos y únicamente se modifican los valores de las fases. Una manera de describir este algoritmo es utilizando una rejilla como análisis y otra como síntesis (figura 22).

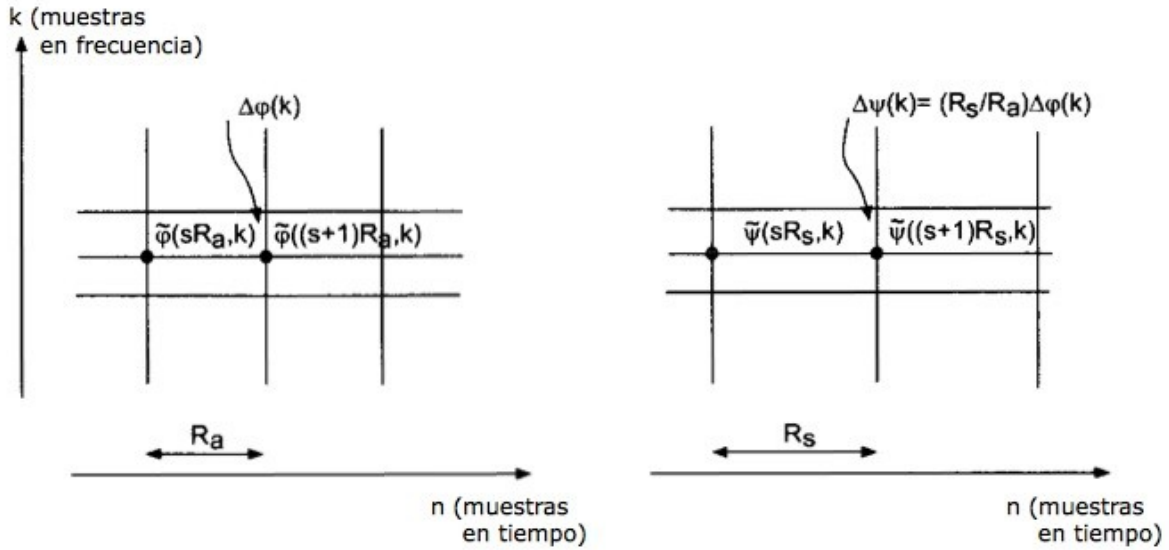


Figura 22. Principio del Escalamiento en Tiempo

De la figura 22, los espacios de la rejilla representan los tramos de la señal. El tamaño que avanza la ventana de análisis, mostrado del lado izquierdo, es de R_a muestras y el tamaño que avanza la ventana de síntesis, mostrado del lado derecho, es de R_s muestras y ésta se fija a 512. Como el PV recorre la señal tramo a tramo, entonces se pueden calcular nuevas diferencias de fase (frecuencias instantáneas) entre tramos consecutivos. Estos valores deben satisfacer la siguiente relación:

$$\Delta\psi(k) = \frac{R_s}{R_a} \Delta\varphi(k) \quad (10)$$

$\varphi(k)$ representa la fase del tramo a analizar y $\psi(k)$ representa la fase del tramo a sintetizar.

La diferencia de fases en la etapa de análisis (mostrada del lado izquierdo de la figura 22) se denota como: $\Delta\varphi(k) = \tilde{\varphi}((s+1)R_a, k) - \tilde{\varphi}(sR_a, k)$ y la diferencia de fases en la etapa de síntesis (lado derecho) como: $\Delta\psi(k) = \tilde{\psi}((s+1)R_s, k) - \tilde{\psi}(sR_s, k)$. El lado izquierdo de la figura 22 muestra que al recorrer la señal inicial cada R_a muestras, se calculan diferencias de fase entre las iteraciones $s+1$ y s . El lado derecho de la figura 22 muestra que por un lado, éstas diferencias de fase son escaladas por el factor R_s/R_a y por otro lado, al momento de sintetizar nuevas (más) fases, se avanza R_s muestras (número mayor a R_a , en este caso) para así obtener más muestras de fase. Hay que mencionar que estas diferencias de fases, son de fases ya desenvueltas. Entonces, al avanzar R_s muestra en la etapa de síntesis, se obtienen más muestras de fase entre tramos consecutivos.

De acuerdo a la relación de la ecuación (10), si se quiere escalar la señal con un factor de dos por ejemplo, entonces el número de muestras R_a que avanza la ventana de análisis tendrá que ser la mitad (256) del número de muestras R_s que avanza la ventana de síntesis [28].

4.4 Implementación del Escalamiento en Tiempo

Como ya se mencionó previamente, si se quiere alargar la señal entonces el tamaño del tramo a sintetizar R_s debe ser mayor que el tamaño del tramo a analizar R_a . Por otro lado, se deben calcular valores de fases nuevas tratando de preservar los cambios instantáneos de fase para cada muestra de frecuencia. Una vez más, esto se logra al calcular diferencias de fase desenvuelta para cada tramo, las cuales son proporcionales a la relación R_s/R_a . Además, se tienen que hacer que los valores de la respuesta de fase del filtro sean cero [31].

Desenvolvimiento de la fase

El algoritmo para desenvolver la fase debe incluir la correspondiente variación de fase Ω_k por cada tramo, para todas las muestras de frecuencias k [31].

La parte izquierda de la ecuación (4) se puede expresar en magnitud y fase [25]:

$$X(sR_a, k) = X_R(sR_a, k) + jX_I(sR_a, k) = |X(sR_a, k)| e^{j\varphi(sR_a, k)} \quad k = 0, 1, \dots, N-1 \quad (11)$$

De las ecuaciones (7) y (11) se tiene:

$$\tilde{X}(sR_a, k) = W_N^{-sR_a k} |X(sR_a, k)| e^{j\varphi(sR_a, k)} \quad (12)$$

La correspondiente fase es:

$$\tilde{\varphi}(sR_a, k) = \Omega_k sR_a + \varphi(sR_a, k) \quad (13)$$

Donde: $\Omega_k = \frac{2\pi}{N} k$ [26] (14)

También se necesita una función que confine cualquier valor arbitrario de fase, dentro del intervalo $(-\pi, \pi)$ (figura 23). A esta función se le llamará *argument* y su correspondiente código en MATLAB es:

```
function phase = argument(phasein)
phase = mod(phasein+pi,-2*pi)+pi;
```

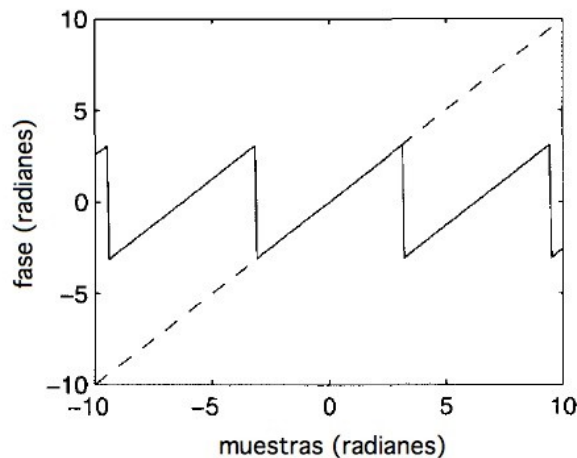


Figura 23. Confinamiento de la fase (diente se cierra)

La figura 23 muestra cómo la fase desenvuelta (líneas punteadas) es confinada dentro del intervalo $(-\pi, \pi)$, teniendo como resultado el trazo tipo diente de sierra.

La diferencia de fase desenvuelta se calcula con la siguiente ecuación:

$$\Delta \varphi((s+1)R_a) = \Omega_k R_a + \text{argument}[\tilde{\varphi}((s+1)R_a) - \tilde{\varphi}(sR_a) - \Omega_k R_a] \quad (15)$$

La ecuación (15) involucra la función *argument* y la variación de fase Ω_k . Los parámetros de la función *argument*, en este caso son $\tilde{\varphi}((s+1)R_a)$ y $\tilde{\varphi}(sR_a)$, los cuales representan las muestras de los tramos en las iteraciones $s+1$ y s , respectivamente; además de incluir la variación de la fase.

El correspondiente código de la ecuación (15) en MATLAB es el siguiente:

```
omega = 2*pi*Ra*(0:N-1)'/N;
phase_unwr = phi-phi0-omega;
phase_wr = argument(phase_unwr);
d_phi = omega + phase_wr;
```

Donde **phi** representa los valores de fase $\tilde{\varphi}((s+1)R_a)$ en la iteración $s+1$ y **phi0** representa los valores de fase $\tilde{\varphi}(sR_a)$ de la iteración previa. **delta_phi** representa la variación de fases desenvueltas $\Delta \varphi((s+1)R_a)$ entre dos tramos consecutivos. **omega** representa la variación de fase Ω_k [31].

Respuesta de Fase Cero del Filtro (Ventana de Ponderación)

El otro factor a considerar es hacer que la respuesta de fase del filtro (ventana), valga cero. De esta manera, no habrá valores de fase producidos por el corrimiento del filtro que interactúen con los valores de fase del tramo de la señal y así, únicamente se manipulan los valores de fase de los tramos. Teniendo esto en cuenta, al manipular la ecuación (2) para obtener la ecuación (4) se produce lo que se llama un corrimiento circular (figura 24). El corrimiento circular es equivalente a hacer un corrimiento a la derecha del tramo, de tal manera que únicamente se utilizan muestras de números positivos y así se obtiene una fase cero del filtro (ventana).

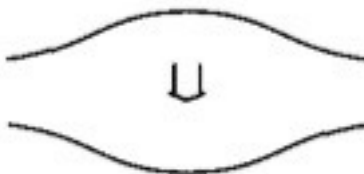


Figura 24. Corrimiento circular del tramo a analizar

La curva superior de la figura 24 es un tramo de la señal $x(n)$, en la etapa de análisis, el cual es de tamaño N . La curva inferior es el resultado de hacer un corrimiento circular del tramo.

La ecuación (7) se puede expresar de la siguiente manera:

$$\tilde{X}(e^{j\Omega}) = e^{-j\Omega \frac{N}{2}} X(e^{j\Omega}) \quad (16)$$

En este caso $\Omega = \Omega_k$, tendiendo así:

$$\tilde{X}(k) = e^{-j \frac{2\pi k N}{2}} X(k) = e^{-j\pi k} X(k) \quad (17)$$

Por la definición de la exponencial compleja, se tiene que la ecuación (17) es equivalente a [22]:

$$\tilde{X}(k) = (-1)^k X(k) \quad (18)$$

Implementación

Al implementar el PV en la sección 3.3 (Implementación del Codificador de Voz en Fase), no se tuvo la necesidad de considerar los dos factores mencionados previamente (Desenvolvimiento de la Fase y Respuesta de Fase Cero del Filtro), por un lado porque no se manipuló la fase de ninguna manera y por otro porque la respuesta de fase del filtro es lineal. Es decir, el mismo cambio de fase que sufre cada tramo de la señal de entrada, lo sufre cada tramo de la señal de salida y no repercute en el resultado de la señal de salida total. El corrimiento circular se tuvo que haber hecho después de la línea 33 del código **PV.m**, antes de calcular la FFT del tramo. Pero como se acaba de mencionar, no se tuvo la necesidad de hacer este corrimiento.

Una vez tomados en cuenta estos dos factores, ahora se procede a implementar el escalamiento en tiempo en MATLAB [32]. Reutilizando el código anterior **PV.m**, se tiene el siguiente código **TS.m** que se encuentra en la carpeta *separados_m* [a].

```

1. close all
2. clear all
3. clc;
4.
5. fs = 44100;
6. f = 500;
7. T = 1/fs;
8. t = 0:T:1-T;
9.
10. x = sin(2*pi*f*t);
11. subplot(2,2,1);
12. plot(x,'k');grid
13. title('a')
14. xlabel('Tiempo (Muestras)')
15. ylabel('Amplitud')
16. axis([0 70000 -1 1])
17.
18. N = 2048;
19. Rs = N/4;
20. Ra = 341; % factor = 1.5
21. w1 = hann(N);
22. w2 = w1;
23. L = length(x);
24.
25. x_in = [zeros(N,1); x'; zeros(N,1)] / max(abs(x));
26.
27. tsr = Rs / Ra;
28. x_out_aux = zeros(N+floor(length(x_in)*tsr),1);
29.
30. omega = 2*pi*Ra*(0:N-1)'/N;

```

```

31.     phi0 = zeros(N,1);
32.     psi  = zeros(N,1);
33.
34.     s_times = floor((L+N) / Ra);
35.     sRa     = 0;
36.     sRs     = 0;
37.
38.     for s=1:s_times-1
39.
40.         grain = x_in(sRa+1:sRa+N).*w1;
41.         grain_sh = fftshift(grain);
42.         X       = fft(grain_sh,N);
43.         X_abs   = abs(X);
44.         phi     = angle(X);
45.
46.         phase_unwr = phi-phi0-omega;
47.         d_phi     = omega + argument(phase_unwr);
48.         phi0     = phi;
49.         phase_unwr = psi+d_phi*tsr;
50.         psi     = argument(phase_unwr);
51.
52.         Y = complex(X_abs.*cos(psi),X_abs.*sin(psi));
53.         grain_sh_sc = real(ifft(Y,N));
54.         grain_aux = fftshift(grain_sh_sc);
55.         grain_sc = grain_aux.*w2;
56.         x_out_aux(sRs+1:sRs+N) = x_out_aux(sRs+1:sRs+N) + grain_sc;
57.
58.         sRa = floor(s * Ra);
59.         sRs = floor(s * Rs);
60.     end
61.
62.     m1 = floor(tsr*(5/6)*N+1);
63.     m2 = length(x_out_aux)-2*m1;
64.
65.     x_out = x_out_aux(m1:m2) / max(abs(x_out_aux));
66.     subplot(2,2,3);
67.     plot(x_out,'k');grid
68.     title('c')
69.     xlabel('Tiempo (Muestras)')
70.     ylabel('Amplitud')
71.     axis([0 70000 -1 1])
72.
73.     L = length(x);
74.     NFFT = 2^nextpow2(L);
75.     X = fft(x,NFFT)/L;
76.     f = fs/2*linspace(0,1,NFFT/2+1);
77.
78.     subplot(2,2,2);
79.     plot(f,2*abs(X(1:NFFT/2+1)),'k');grid
80.     title('b')
81.     xlabel('Frecuencia (Hz)')
82.     ylabel('Amplitud Normalizada')
83.     axis([470 530 0 1])
84.
85.     L = length(x_out);
86.     NFFT = 2^nextpow2(length(x_out));
87.     X_OUT = fft(x_out,NFFT)/L;
88.     f = fs/2*linspace(0,1,NFFT/2+1);
89.
90.     subplot(2,2,4);
91.     plot(f,2*abs(X_OUT(1:NFFT/2+1)),'k');grid
92.     title('d')
93.     xlabel('Frecuencia (Hz)')
94.     ylabel('Amplitud Normalizada')
95.     axis([470 530 0 1])

```

```

96.
97.     l = length(x);
98.     d = 1/fs;
99.     d = round(d*100);
100.    t = d/100
101.
102.    lout = length(x_out);
103.    d = lout/fs;
104.    d = round(d*100);
105.    tm = d/100
106.
107.    figure(2)
108.    subplot(1,2,1)
109.    plot(x,'k','LineWidth',2);grid;hold on
110.    title('a')
111.    xlabel('Tiempo (Muestras)')
112.    ylabel('Amplitud')
113.    axis([fs-200 fs+200 -1 1])
114.
115.    subplot(1,2,1);
116.    plot(x_out,'k')
117.
118.    L = length(x);
119.    NFFT = 2^nextpow2(L);
120.    X = fft(x,NFFT)/L;
121.    f = fs/2*linspace(0,1,NFFT/2+1);
122.
123.    subplot(1,2,2);
124.    plot(f,2*abs(X(1:NFFT/2+1)),'k','LineWidth',2);grid;hold on
125.    title('b')
126.    xlabel('Frecuencia (Hz)')
127.    ylabel('Amplitud Normalizada')
128.    axis([490 510 0 1])
129.
130.    L = length(x_out);
131.    NFFT = 2^nextpow2(length(x_out));
132.    X_OUT = fft(x_out,NFFT)/L;
133.    f = fs/2*linspace(0,1,NFFT/2+1);
134.
135.    subplot(1,2,2);
136.    plot(f,2*abs(X_OUT(1:NFFT/2+1)),'k');
137.
138.    figure(3)
139.    subplot(2,1,1)
140.    plot(x_out,'k');grid;
141.    title('a')
142.    xlabel('Tiempo (Muestras)')
143.    ylabel('Amplitud')
144.    axis([0 2500 -1 1])
145.
146.    subplot(2,1,2);
147.    plot(x_out,'k');grid;
148.    title('b')
149.    xlabel('Tiempo (Muestras)')
150.    ylabel('Amplitud')
151.    axis([length(x_out)-2500 length(x_out) -1 1])

```

La etapa de análisis se encuentra entre las líneas 40 y 44. La etapa de procesamiento se encuentra entre las líneas 46 y 50; en donde la diferencia de fases desenvuelta se calcula en la línea 47, la fase se manipula en la línea 49 y se envuelve en la 50. La etapa de síntesis se encuentra entre las líneas 52 y 56.

Para seleccionar el factor de escalamiento la variable R_a (línea 20) toma los valores mostrados en la tabla 4. Hay que recordar que la variable R_s se queda fija con un valor de 512 muestras.

Ra	Factor
682	0.75
512	1.0
410	1.25
341	1.5
293	1.75
256	2.0
228	2.25

Tabla 4. Factor de escalamiento

5. Ecuación

5.1 Introducción

La ecualización es el proceso de alterar la respuesta frecuencial de una señal de audio. El ecualizador, el cual utiliza filtros lineales, es el dispositivo analógico o digital que llevar a cabo el proceso de ecualización.

A diferencia de los filtros, los cuales atenúan el espectro frecuencial por encima o por debajo de la frecuencia de corte, los ecualizadores moldean el espectro frecuencial al resaltar ciertas bandas de frecuencia mientras que el resto de las bandas se mantienen intactas. Los ecualizadores se construyen con una serie de Filtros Campana (*shelving*) y Pico (*peak*) de primero y segundo orden, los cuales se controlan independientemente (figura 25). Los Filtros Campana amplifican o atenúan las bandas de frecuencias altas o bajas con los parámetros: frecuencia de corte f_c y ganancia G . Los Filtros Pico amplifican o atenúan las bandas de frecuencia medias con los parámetros: frecuencia de corte f_c , ancho de banda f_b y ganancia G .

Uno de los filtros que se utiliza típicamente es el Filtro Pico con constante Q . El factor Q se define por la relación ancho de banda a frecuencia de corte $Q = f_b/f_c$. Entonces, la frecuencia de corte de los Filtros Pico se ajusta manteniendo el factor Q constante. Esto quiere decir que el ancho de banda se incrementa cuando la frecuencia de corte se incrementa y viceversa.

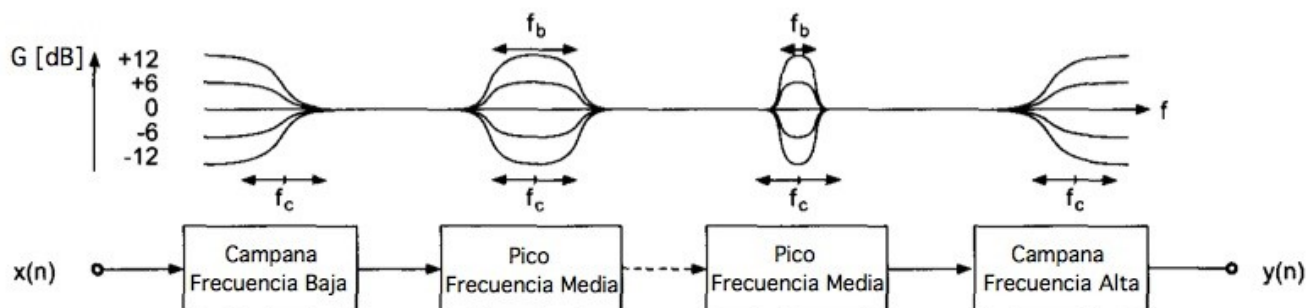


Figura 25. Serie de Filtros Campana y Pico

Las aplicaciones que pueden tener estos filtros paramétricos se encuentran en los ecualizadores paramétricos, ecualizadores de octavas y todo tipo de dispositivos de ecualización en consolas y mezcladoras [33].

5.2 Filtro Digital Biquad

En procesamiento de señales, el término Biquad es una abreviación de “biquadratic” (bicuadrático), y se refiere a un filtro digital que contiene dos polos y dos ceros. La función de transferencia de este filtro se define de la siguiente manera:

$$H(z) = g \frac{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (19)$$

g representa la ganancia del filtro. El nombre bicuadrático se refiere a que los polinomios tanto del numerador como el denominador de la función de transferencia, son de segundo orden.

Implementación

Para implementar el filtro Biquad, de la función de transferencia se llega a una ecuación, y la implementación de esa ecuación se conoce como Forma Directa 1. Sin embargo, la implementación se mejora al hacerla con la Forma Directa 2. El flujo del diagrama de la Forma Directa 2 para implementar el filtro Biquad se muestra en la figura 26.

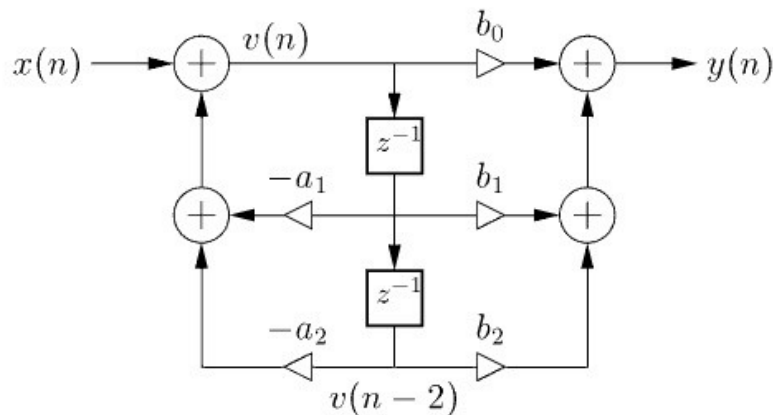


Figura 26. Forma Directa 2

A diferencia de la Forma Directa 1, la cual requiere de cuatro registros de retardo, la Forma Directa 2 únicamente requiere de dos. La ecuación para implementar la Forma Directa 2 es la siguiente:

$$y(n) = b_0 v(n) + b_1 v(n-1) + b_2 v(n-2) \quad (20)$$

en donde: $v(n) = x(n) - a_1 v(n-1) - a_2 v(n-2)$

Como en todas las estructura de filtros de segundo orden, los polos y ceros son sensibles a los errores de redondeo en los coeficientes a_i y b_i , especialmente para las funciones de transferencia de ordenes mayores a 2. Una sensibilidad menor y por lo tanto una mejor estabilidad del filtro, se obtiene al utilizar filtros de orden menor [34]. El que el filtro sea más estable quiere decir que la función impulsiva del filtro $h(n)$ tiende a cero conforme las muestras de tiempo n tienden a infinito [35].

5.3 Diseño de Filtros Campana y Pico de Segundo Orden

Filtros Campana

Para varias aplicaciones especialmente en diseños de ecualizadores avanzados, la pendiente de las curvas de frecuencia se incrementa por una función de transferencia de segundo orden. Las formulas para diseñar Filtros Campana de segundo orden se muestran en la tabla 2, en donde $K = \tan(\pi f_c / f_s)$. La respuesta en frecuencia de estos filtros de segundo orden para frecuencias bajas y altas a varios factores de ganancias se muestra en la figura 27. El eje de la coordenada x corresponde al dominio de la frecuencia en escala logarítmica. El eje de la coordenada y corresponde a la ganancia en decibeles. Cada línea representa la respuesta en frecuencia de una amplificación en particular; la figura muestra amplificaciones de aproximadamente 4, 8, 12 y 16 dB [36].

campana de frecuencia baja (amplificar $V_0 = 10^{G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1+\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(V_0K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$
campana de frecuencia baja (atenuar $V_0 = 10^{-G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1+\sqrt{2}K+K^2}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{2(V_0K^2-1)}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{1-\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2V_0}K+V_0K^2}$
campana de frecuencia alta (amplificar $V_0 = 10^{G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{V_0+\sqrt{2V_0}K+K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-V_0)}{1+\sqrt{2}K+K^2}$	$\frac{V_0-\sqrt{2V_0}K+K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$
campana de frecuencia alta (atenuar $V_0 = 10^{-G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1+\sqrt{2}K+K^2}{V_0+\sqrt{2V_0}K+K^2}$	$\frac{2(K^2-1)}{V_0+\sqrt{2V_0}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{V_0+\sqrt{2V_0}K+K^2}$	$\frac{2(K^2/V_0-1)}{1+\sqrt{2/V_0}K+K^2/V_0}$	$\frac{1-\sqrt{2/V_0}K+K^2/V_0}{1+\sqrt{2/V_0}K+K^2/V_0}$

Tabla 2. Diseño de Filtros Campana de segundo orden

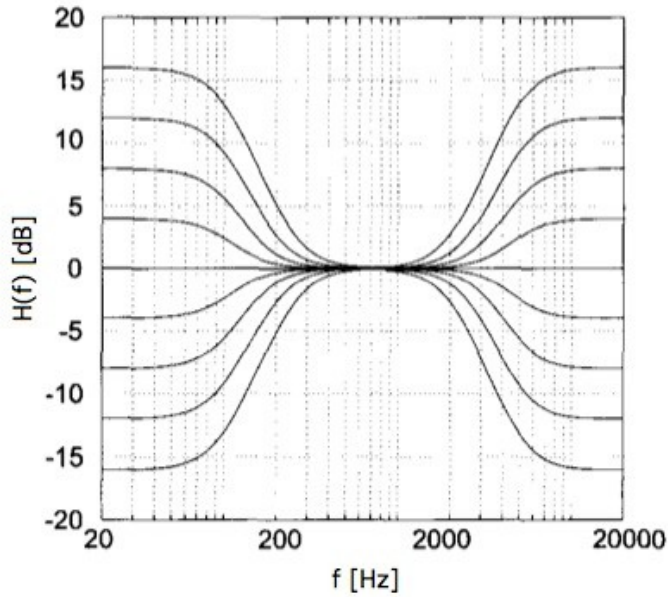


Figura 27. Respuesta frecuencial de Filtros Campana de segundo orden

Filtros Pico

Las formulas para diseñar Filtros Pico de segundo orden se muestran en la tabla 3. La respuesta frecuencial para diferentes configuraciones se muestra en la figura 28. El lado izquierdo de la figura muestra variaciones de la ganancia de aproximadamente 4, 8, 12 y 16 dB con un ancho de banda fijo. La parte derecha de la figura muestra una variación del ancho de banda o factor Q con una ganancia fija de aproximadamente 15 dB [37].

pico (amplificar $V_0 = 10^{G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1 + \frac{V_0}{Q_\infty} K + K^2}{1 + \frac{1}{Q_\infty} K + K^2}$	$\frac{2(K^2 - 1)}{1 + \frac{1}{Q_\infty} K + K^2}$	$\frac{1 - \frac{V_0}{Q_\infty} K + K^2}{1 + \frac{1}{Q_\infty} K + K^2}$	$\frac{2(K^2 - 1)}{1 + \frac{1}{Q_\infty} K + K^2}$	$\frac{1 - \frac{1}{Q_\infty} K + K^2}{1 + \frac{1}{Q_\infty} K + K^2}$
pico (atenuar $V_0 = 10^{-G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1 + \frac{1}{Q_\infty} K + K^2}{1 + \frac{V_0}{Q_\infty} K + K^2}$	$\frac{2(K^2 - 1)}{1 + \frac{V_0}{Q_\infty} K + K^2}$	$\frac{1 - \frac{1}{Q_\infty} K + K^2}{1 + \frac{V_0}{Q_\infty} K + K^2}$	$\frac{2(K^2 - 1)}{1 + \frac{V_0}{Q_\infty} K + K^2}$	$\frac{1 - \frac{V_0}{Q_\infty} K + K^2}{1 + \frac{V_0}{Q_\infty} K + K^2}$

Tabla 3. Diseño de Filtros Pico de segundo orden

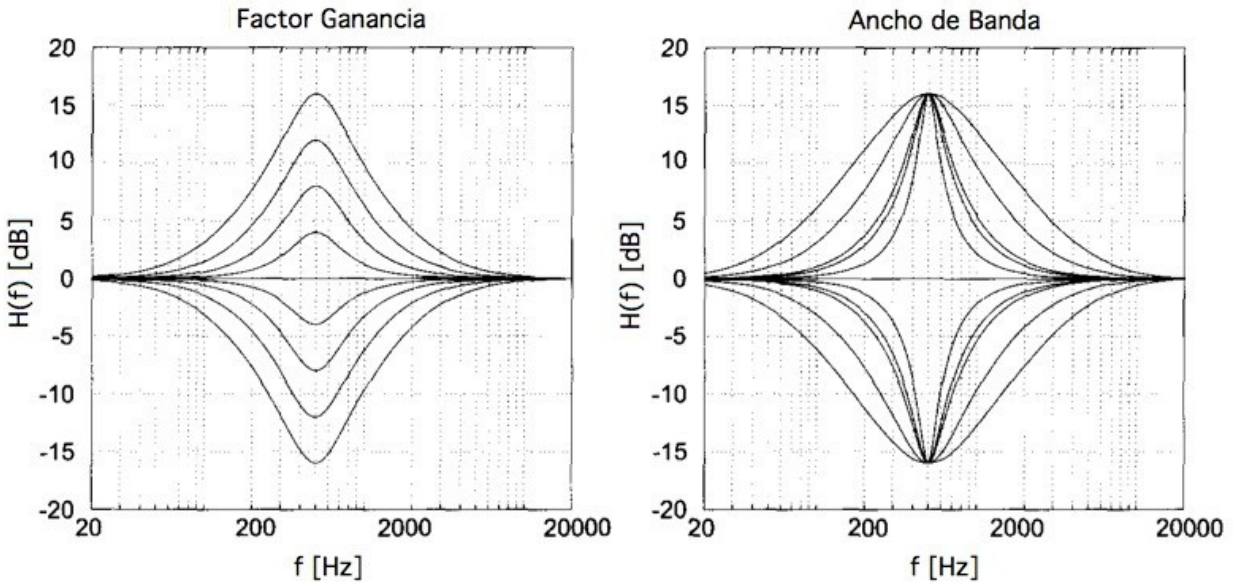


Figura 28. Respuesta frecuencial de Filtros Pico de segundo orden

5.4 Implementación del Ecuador

En el capítulo 1 (Introducción) se menciona la posibilidad de rehabilitar niños DAL mediante estímulos auditivos modificados. La modificación consiste en amplificar la señal de audio 20 dB en la gama de frecuencias 1-4 kHz (además de lentificarla del 50 al 25 %). Debido a esta gama de frecuencias, no es necesario utilizar Filtros Campana, los cuales abarcan frecuencias bajas y altas, solamente es necesario implementar el ecualizador con Filtros Pico. Después de hacer varias pruebas para obtener dicha amplificación, se utilizaron 5 Filtros Pico con el siguiente orden y rangos frecuenciales: 2 filtros de media de octava, 1 filtro de octava y 2 filtros de media de octava; y los valores de las ganancias son las siguientes: 10, 2, 16, 2 y 10, respectivamente.

El siguiente código en MATLAB muestra la implementación del ecualizador. Este código se encuentra en el archivo **Ecu.m** [a] que se encuentra en la carpeta **separados_m**, y ocupa las funciones: **filteringEph.m**, **peaking.m** y **biquad.m**. La función **filteringEph.m** define las bandas en frecuencia y los correspondientes valores de parámetros de acuerdo a las necesidades de cada filtro. La función **peaking.m** se encarga del diseño, es decir, de entregar los valores de los coeficientes por cada filtro. La función **biquad.m** lleva a cabo la ecualización (filtrado).

```

1. clear all
2. close all
3. clc;
4.
5. fs = 44100;
6. T = 1/fs;
7. t = 0:T:.02-T;
8.
9. %%% Señal
10. f = 500;
11. y1 = sin(2*pi*f*t); % 500 Hz
12. y2 = sin(2*pi*(2*f)*t); % 1000 Hz
13. y3 = sin(2*pi*(4*f)*t); % 2000 Hz

```

```

14.     y4 = sin(2*pi*(8*f)*t); % 4000 Hz
15.     y5 = sin(2*pi*(16*f)*t); % 8000 Hz
16.
17.     x = [y1 y2 y3 y4 y5];
18.
19.     L = length(y1);
20.     NFFT = 2^nextpow2(5*L);
21.     X = fft(x,NFFT)/L;
22.     f = fs/2*linspace(0,1,NFFT/2+1);
23.
24.     figure(1)
25.     plot(f,2*abs(X(1:NFFT/2+1)),'k')
26.     xlabel('Frecuencia (Hz)');
27.     ylabel('Amplitud Normalizada');
28.     axis([0 10000 0 1]); grid
29.
30.     %% Respuesta Impulsiva del Ecualizador
31.     l = 500;
32.     z = [1 zeros(1,l-1)];
33.     h = filteringEph(z,fs);
34.
35.     figure(2)
36.     subplot(2,1,1);
37.     plot(h,'k','LineWidth',2);
38.     title('a')
39.     xlabel('Muestras');
40.     ylabel('Amplitud');
41.     axis([-10 200 -2 3]); grid
42.
43.     %% Respuesta Frecuencial del Ecualizador
44.     m = 2^nextpow2(l);
45.     H = fft(h,m);
46.
47.     subplot(2,1,2);
48.     freq = fs/2*linspace(0,1,m/2);
49.     semilogx(freq,20*log10(abs(H(1:round(m/2))))),'k','LineWidth',2);
50.     title('b')
51.     xlabel('Frecuencia [Hz]');
52.     ylabel('Amplitud (dB)');
53.     axis([0 20000 -5 25]); grid;
54.
55.     %% Ecualización de la Señal
56.     x_eph = filteringEph(x,fs);
57.
58.     figure(3)
59.     subplot(2,2,1)
60.     plot(x,'k')
61.     title('a')
62.     xlabel('Tiempo (Muestras)')
63.     ylabel('Amplitud')
64.     axis([0 4400 -10.5 10.5]); grid;
65.
66.     subplot(2,2,2)
67.     plot(x_eph,'k')
68.     title('b')
69.     xlabel('Tiempo (Muestras)')
70.     ylabel('Amplitud')
71.     axis([0 4400 -10.5 10.5]); grid;
72.
73.     %% Comparación de Espectros
74.     L = length(y1);
75.     NFFT = 2^nextpow2(5*L);
76.     X = fft(x,NFFT)/L;
77.     f = fs/2*linspace(0,1,NFFT/2+1);
78.

```

```

79.     subplot(2,2,3);
80.     plot(f,2*abs(X(1:NFFT/2+1)),'k')
81.     title('c')
82.     xlabel('Frecuencia (Hz)');
83.     ylabel('Amplitud Normalizada');
84.     axis([0 10000 0 2]); grid
85.
86.     L = length(x_eph);
87.     NFFT = 2^nextpow2(L);
88.     X_EPH = fft(x_eph,NFFT)/(L);
89.     f = fs/2*linspace(0,1,NFFT/2+1);
90.
91.     subplot(2,2,4);
92.     plot(f,2*abs(X_EPH(1:NFFT/2+1)),'k')
93.     title('d')
94.     xlabel('Frecuencia (Hz)');
95.     ylabel('Amplitud Normalizada');
96.     axis([0 10000 0 2]); grid
97.
98.     figure(4)
99.     subplot(2,1,1)
100.    plot(x,'k','LineWidth',2);hold on;grid
101.    title('a')
102.    xlabel('Tiempo (Muestras)')
103.    ylabel('Amplitud')
104.    axis([2200 2500 -10.5 10.5])
105.
106.    subplot(2,1,1)
107.    plot(x_eph,'k')
108.
109.    L = length(y1);
110.    NFFT = 2^nextpow2(5*L);
111.    X = fft(x,NFFT)/L;
112.    f = fs/2*linspace(0,1,NFFT/2+1);
113.
114.    subplot(2,1,2);
115.    plot(f,2*abs(X(1:NFFT/2+1)),'k','LineWidth',2);hold on;grid
116.    title('b')
117.    xlabel('Frecuencia (Hz)');
118.    ylabel('Amplitud Normalizada');
119.    axis([1700 2300 0 2])
120.
121.    L = length(x_eph);
122.    NFFT = 2^nextpow2(L);
123.    X_EPH = fft(x_eph,NFFT)/(L);
124.    f = fs/2*linspace(0,1,NFFT/2+1);
125.
126.    subplot(2,1,2);
127.    plot(f,2*abs(X_EPH(1:NFFT/2+1)),'k')

```

Para ilustrar la ecualización de una manera clara, se utiliza una señal compuesta por ondas senoidales de frecuencias: 500, 1000, 2000, 4000 y 8000 Hz; muestreada a $F_s = 44,100$ Hz por segundo y de duración $t = 0.1$ segundos cada una. El cálculo de la respuesta frecuencial del ecualizador se encuentra en la línea 45. La ecualización (filtrado) de la señal se encuentra en la línea 56.

6. Análisis de Resultados

La implementación del Codificador, el Ecuador y el Escalamiento únicamente se implementaron con la aplicación MATLAB pero no con el lenguaje Java. La finalidad de utilizar MATLAB es para obtener resultados rápidos y concisos y así tener la certeza de contar con los resultados esperados. La implementación de la aplicación HIPELDA sí se hizo con ambas herramientas. Los resultados se muestran a continuación.

6.1 Codificador de Voz en Fase

De acuerdo a la sección 3.3; el código **PV.m** utiliza como ejemplo una señal senoidal a $f = 500$ Hz, muestreada a $F_s = 44,100$ Hz cada segundo y de duración $t = 1$ segundo. Los resultados después de ejecutar el código se muestran en la figura 29. La gráfica a) muestra una porción de la señal inicial y la gráfica b) muestra su correspondiente respuesta en frecuencia. La gráfica c) muestra la señal resultante obtenida después de haberla pasado a través del Codificador de Voz en Fase, y la gráfica d) muestra su correspondiente respuesta en frecuencia.

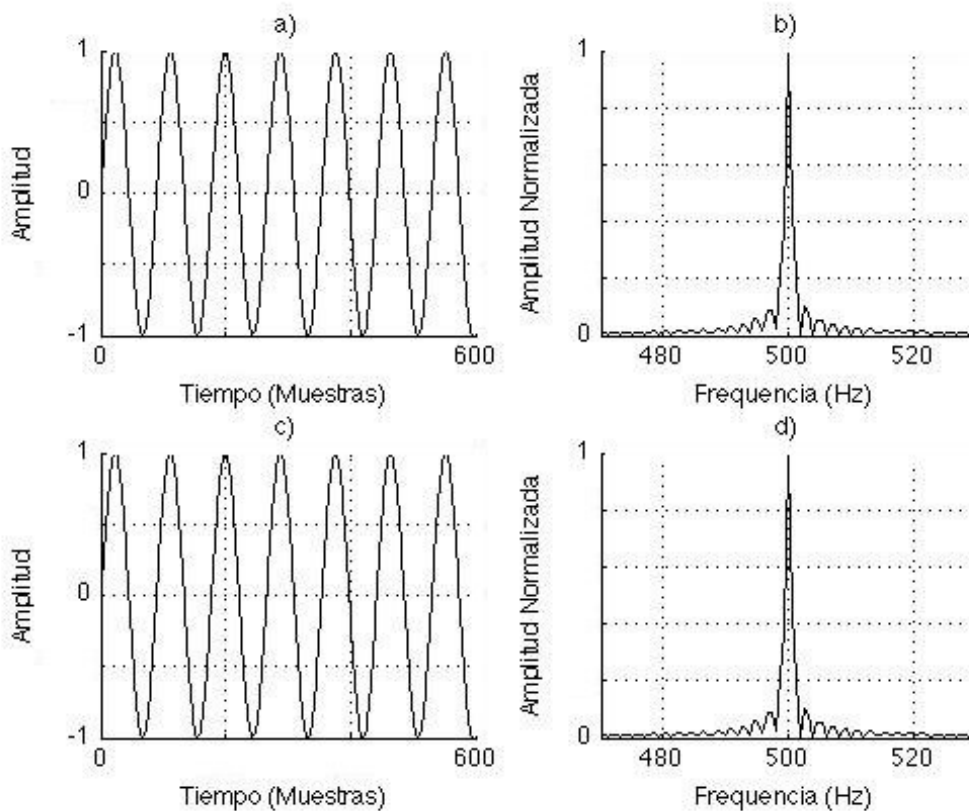


Figura 29. Señales iniciales a) y b). Señales resultantes c) y d)

La figura 29 muestra que ambas señales inicial y resultante, gráficas a) contra c) y b) contra d), son exactamente las mismas, mostrando así la efectividad del algoritmo. A continuación se analiza gráficamente la implementación del PV mostrando los resultados de las primeras cinco iteraciones. Se apreciará cómo la señal inicial es analizada y cómo la señal resultante es sintetizada.

En la primera iteración los índices sRa y sRs valen 0 , además, considerando que a la señal inicial x_{in} se le agrega N ceros al principio y al final, entonces el tramo llamado **grain** contendrá únicamente ceros. El tramo de análisis se muestra en la gráfica a) y la señal resultante se muestra en la gráfica b), de la figura 30.

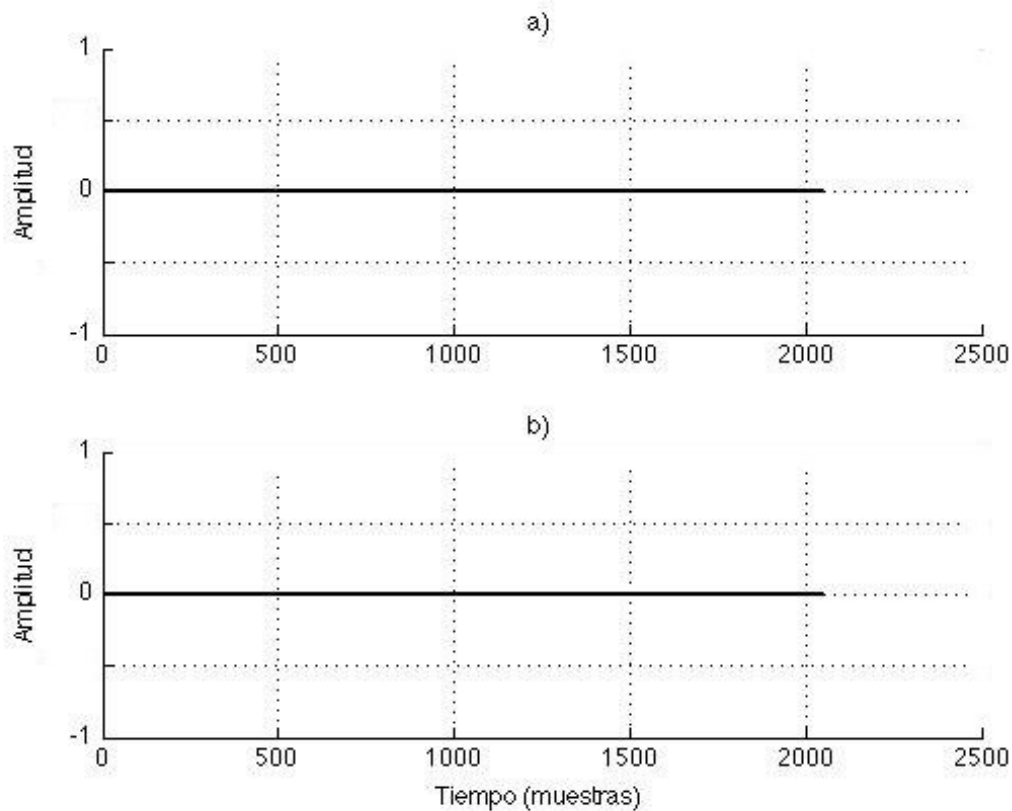


Figura 30. Primera iteración del Codificador de Voz en Fase

De la figura 30 podemos ver que las muestras que abarca el eje de las coordenadas x es de 0 a 2500 . Por otro lado, como las ventanas de análisis $w1$ y de síntesis $w2$ son de tamaño N muestras (2024), entonces ambos tramos de análisis y síntesis contienen 2024 muestras de ceros en la primera iteración.

En la segunda iteración los índices sRa y sRs ahora valen Ra y Rs , respectivamente. Es decir la ventana $w1$ de análisis y la ventana $w2$ de síntesis, de tamaño N , ahora avanzan Ra y Rs muestras respectivamente. El segundo tramo de análisis junto con el previo (primero) se muestra en la gráfica a) y la señal resultante después de traslapar los dos tramos, se muestra en la gráfica b), de la figura 31.

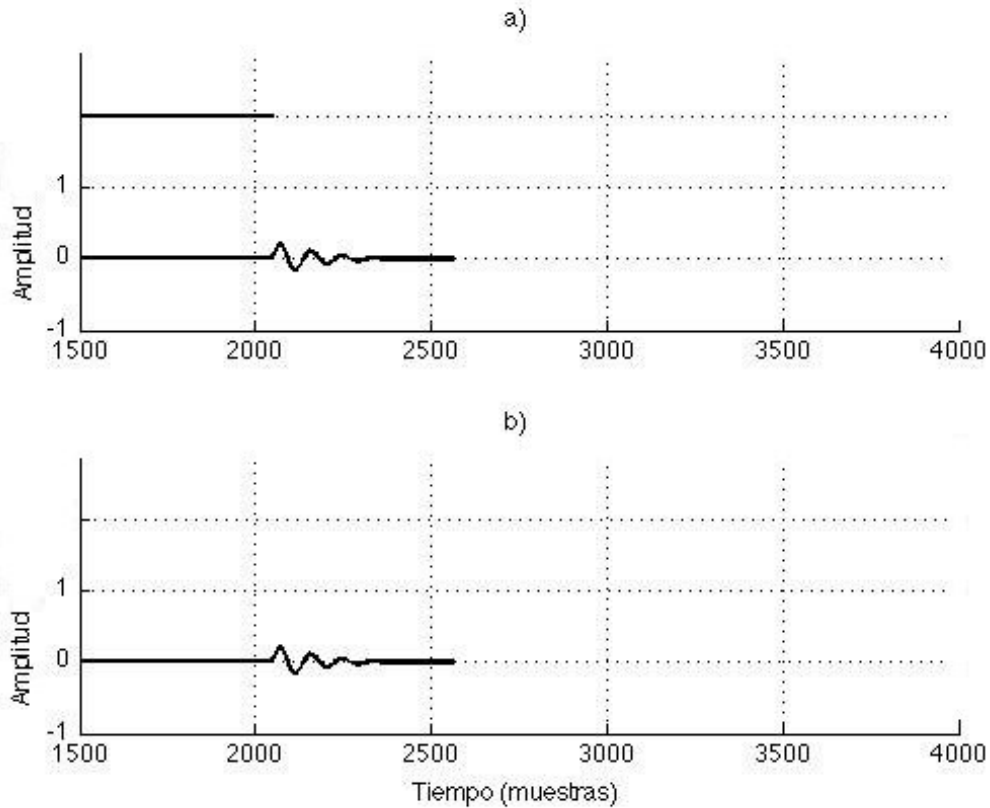


Figura 31. Segunda iteración del Codificador de Voz en Fase

De la figura 31 tenemos lo siguiente. Aquí podemos ver que ahora las muestras que abarca el eje de las coordenadas x es de 1500 a 4000, esto con el fin de observar más de cerca los tramos de la señal. De la gráfica b) podemos ver que el traslape de los tramos mostrados en la gráfica a), es exactamente el mismo tramo de la segunda iteración, lo cual se debe a que las muestras del tramo de la primera iteración fueron todos ceros.

En la tercera iteración los índices sRa y sRs ahora valen 2 veces Ra y Rs , respectivamente. Es decir, la ventana $w1$ de análisis y la ventana $w2$ de síntesis, ahora avanzan $2 \cdot Ra$ y $2 \cdot Rs$ muestras, respectivamente. El tercer tramo de análisis junto con los previos (primero y segundo) se muestra en la gráfica a) y la señal resultante después de traslapar los tres tramos, se muestra en la gráfica b), de la figura 32.

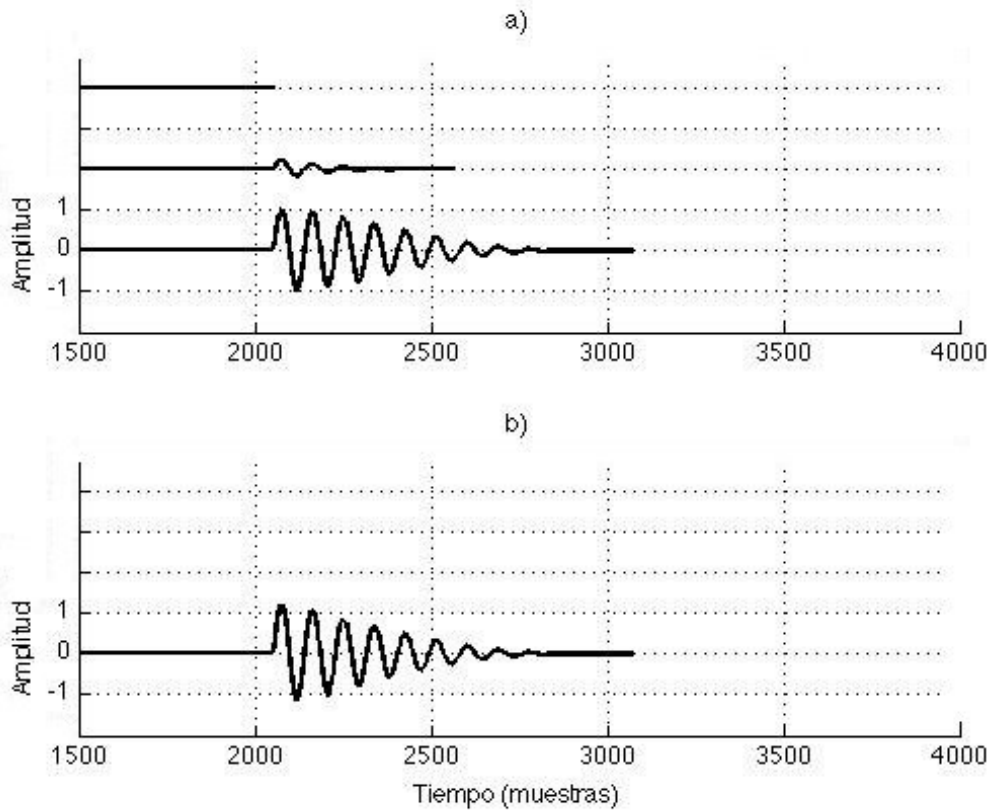


Figura 32. Tercera iteración del Codificador de Voz en Fase

De la figura 23 tenemos que, igualmente, después de traslapar los tramos mostrados en la gráfica a), como resultado obtenemos la gráfica b). Siguiendo el mismo procedimiento, las figuras 33 y la 34 muestran las cuarta y la quinta iteración, respectivamente.

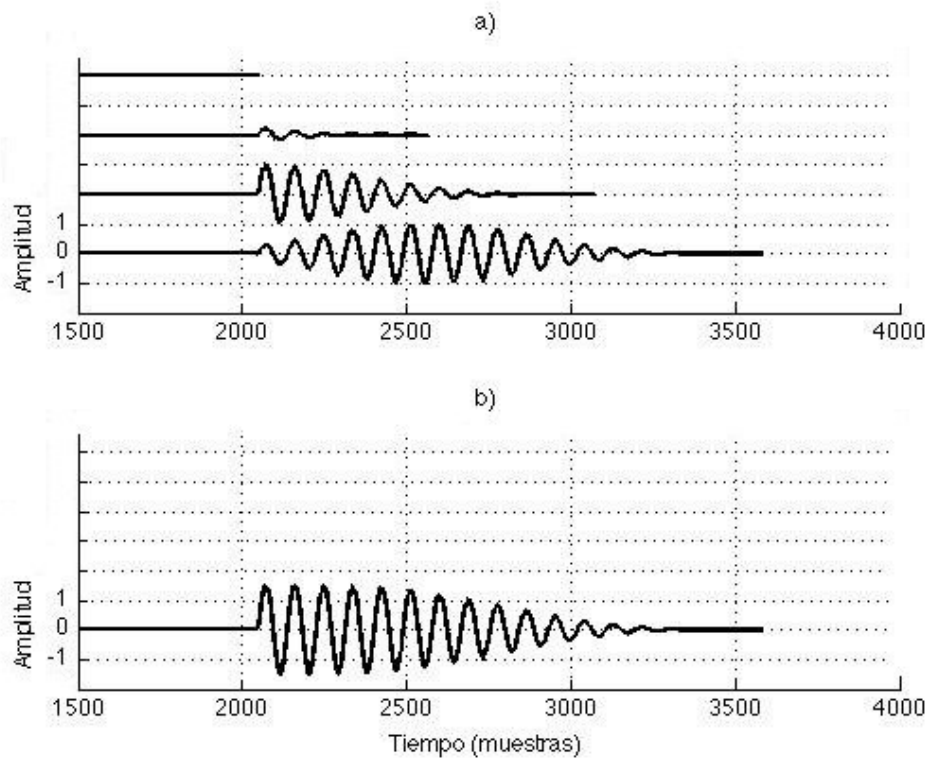


Figura 33. Cuarta iteración del Codificador de Voz en Fase

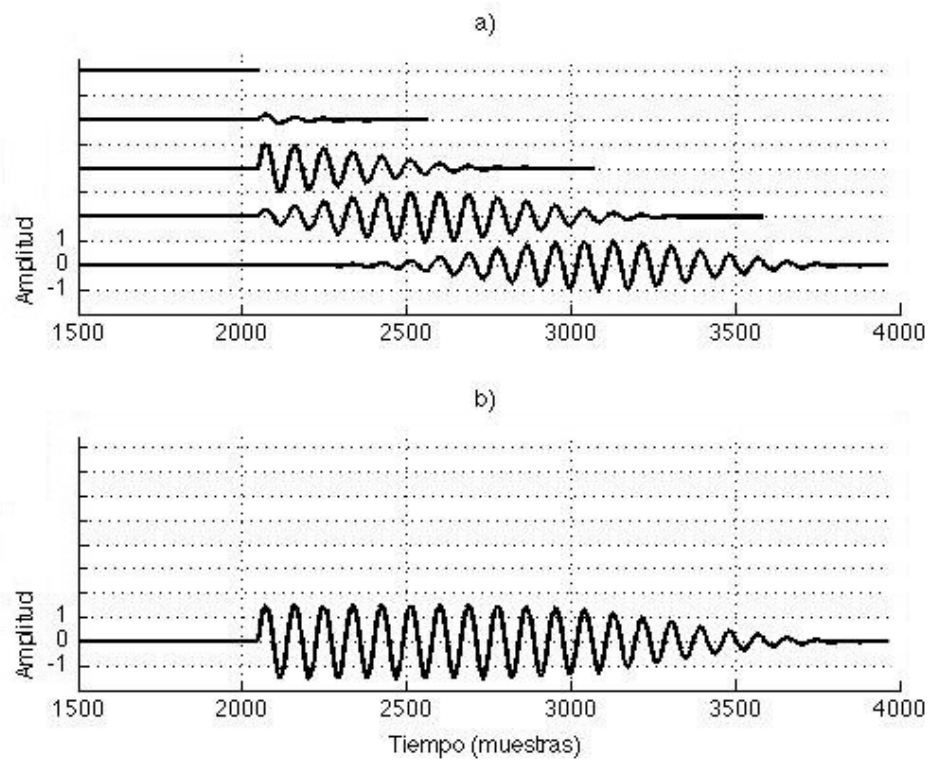


Figura 34. Quinta iteración del Codificador de Voz en Fase

Si ampliamos la gráfica b) de la figura 34, obtenemos la gráfica mostrada en la figura 35.

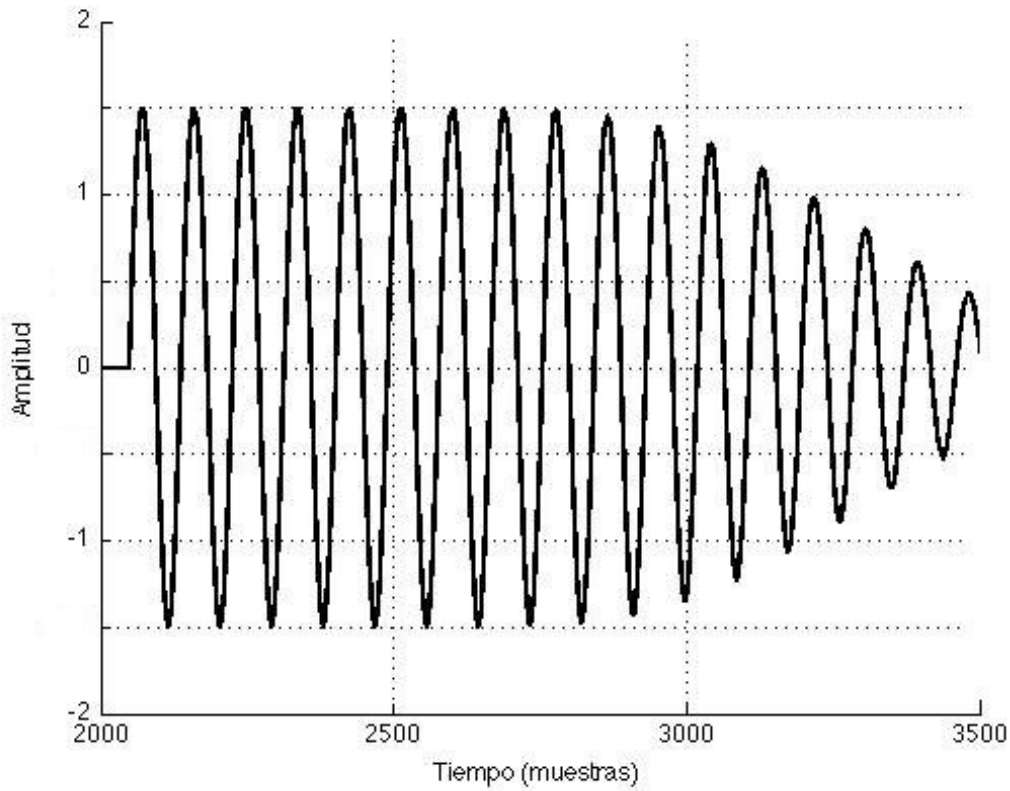


Figura 35. Toma ampliada de la quinta iteración

La figura 35 muestra que en este punto, la señal resultante x_{out} ya ha tomado la forma de una señal senoidal, la cual es igual a la señal inicial x_{in} (figura 29) y al completarse el número total de iteraciones, se formará la señal resultante completa. De la misma figura 35, podemos notar que la amplitud pico a pico es de -1.5 a 1.5, no obstante, como parte del algoritmo, al final del mismo la señal se normalizará y así la amplitud quedará entre -1 y 1.

6.2 Escalamiento en Tiempo

El código **TS.m** utiliza como ejemplo la misma señal utilizada al implementar el PV, además el factor de escalamiento con el que la señal se escala es **1.5**, es decir la señal se alarga un 50 % más. Los resultados después de ejecutar el código se muestran en la figura 36; igualmente, las gráficas a) y b) corresponden a la señal inicial y las gráficas c) y d) corresponden a la señal resultante.

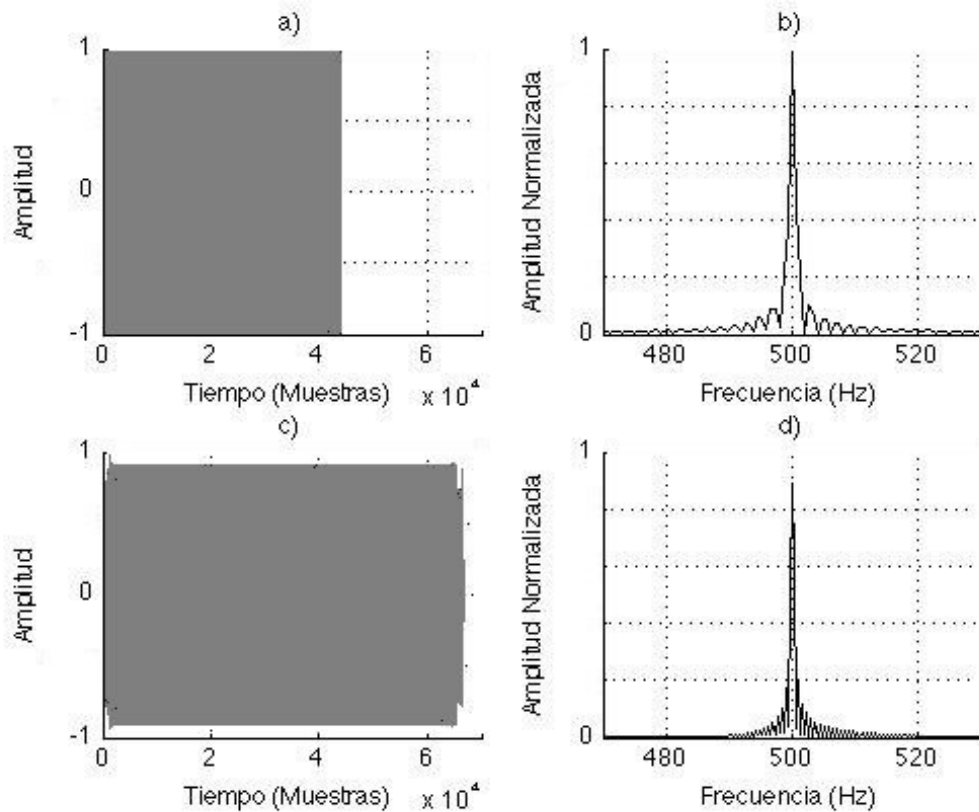


Figura 36. Escalamiento en el tiempo de la señal inicial

De la figura 36 tenemos lo siguiente. La gráfica c) muestra cómo la señal resultante es mayor en tamaño (dominio del tiempo) que la señal a). Por otro lado, al comparar la gráfica d) podemos notar que se produce una distorsión respecto a la gráfica b), la cual se debe al método de escalamiento.

La figura 37 muestra un acercamiento y traslape de la figura 36 entre las gráficas a) contra c) y entre las gráficas b) contra d); la gráfica de línea gruesa representa la señal inicial x_{in} y la gráfica de línea delgada representa la señal resultante x_{out} . La figura 37 muestra cómo la señal resultante x_{out} sufre una ligera distorsión, debido al escalamiento; aquí se puede apreciar la eficiencia del método empleado en la sección 4.3 (Descripción del Escalamiento en Tiempo) para escalar una señal. Si apareciera otra componente distinta a la de 500 Hz, o si la señal se atenuará o amplificará considerablemente, por ejemplo al 30% o más, entonces se podría decir que el método empleado no es eficiente. Parte de esta ligera distorsión se puede apreciar al inicio y al final de la señal escalada, mostrada en la figura 38.

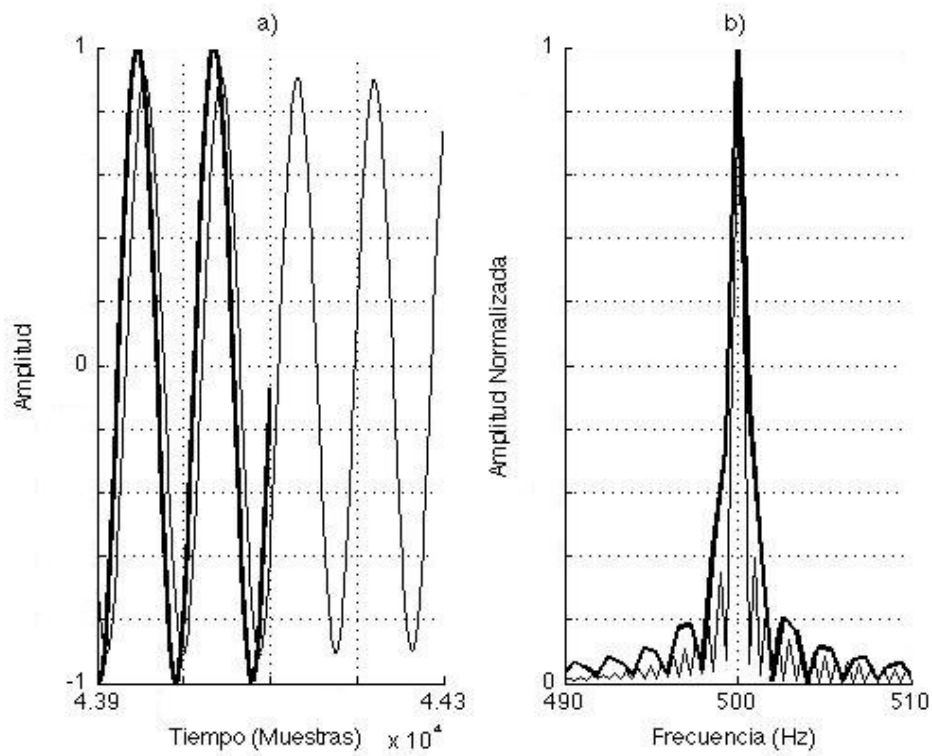


Figura 37. Toma ampliada de la señal inicial contra la señal resultante

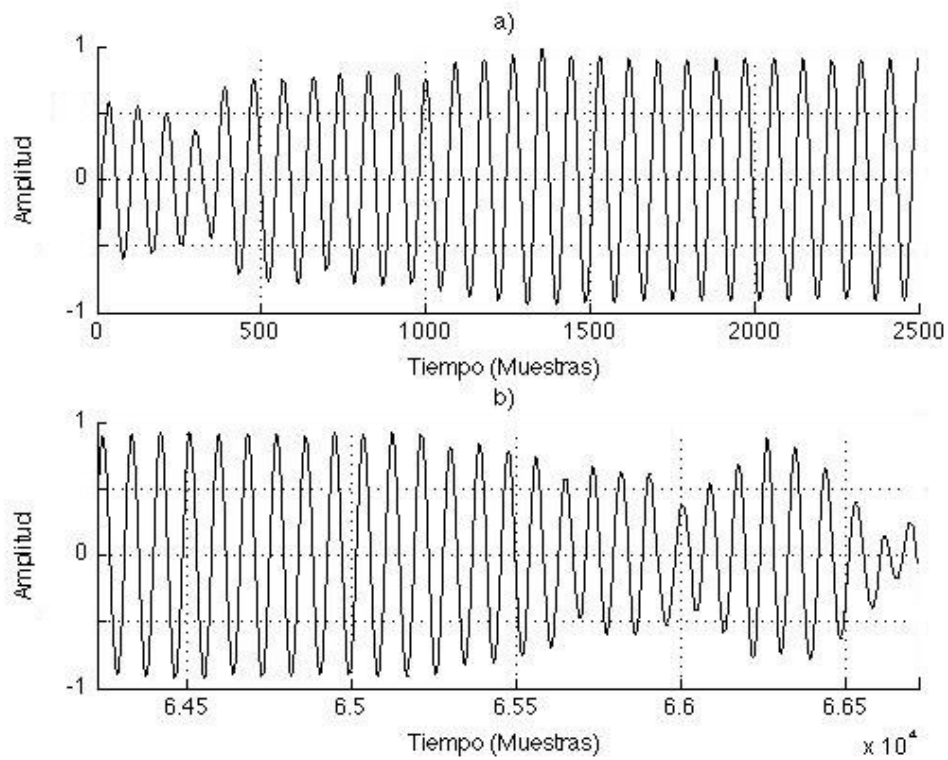


Figura 38. Liger distorsión al principio a) y final b) de la señal escalada

De igual manera que la sección 6.1 (Codificador de Voz en Fase), a continuación se analiza gráficamente la implementación del Escalamiento en Tiempo mostrando los resultados de cinco iteraciones. La finalidad es visualizar cómo la señal resultante es sintetizada. Para fines prácticos, el análisis se hace desde un punto medio de la señal inicial. La figura 39 muestra la primera iteración.

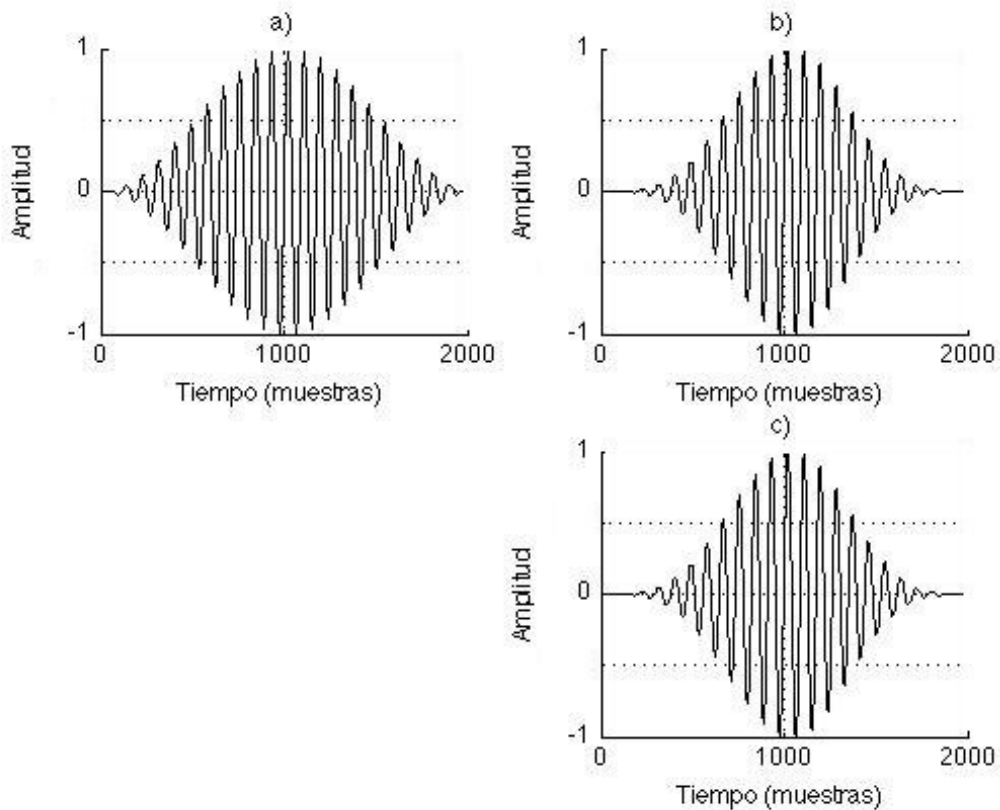


Figura 39. Primera iteración del método Escalamiento en Tiempo

De la figura 39 se tiene lo siguiente. El tramo mostrado en la gráfica a) muestra la ponderación de la primera iteración (etapa de análisis). El tramo mostrado en la gráfica b) muestra el resultado de procesar ese tramo, es decir, después de haber manipulado la fase debido al método Escalamiento en Tiempo (etapa de síntesis). Aquí se nota un cambio en la forma del tramo, resultado de la alteración de la fase. El tramo mostrado en la gráfica c) es tal cual el tramo de la gráfica b), ya que en este momento no se ha traslapado con otro tramo aún. A continuación, la figura 40 muestra la segunda iteración.

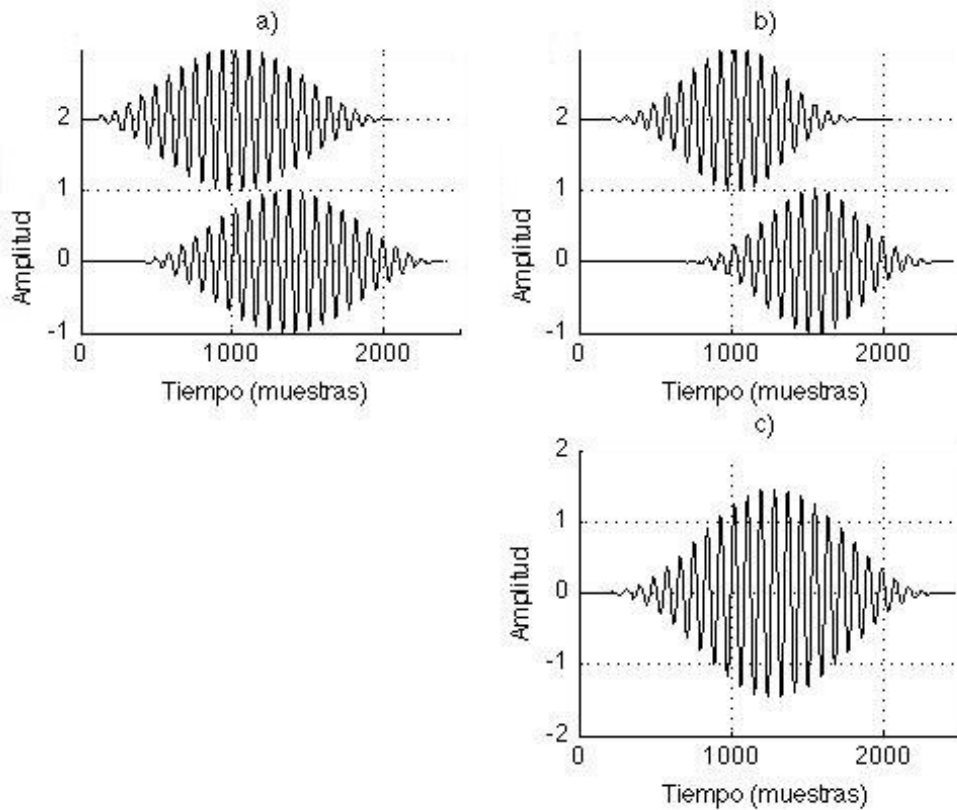


Figura 40. Segunda iteración del método Escalamiento en Tiempo

De la figura 40 tenemos lo siguiente. En la segunda iteración, el tramo avanza más muestras en la etapa de síntesis (gráfica inferior del inciso b)) que en la etapa de análisis (gráfica inferior del inciso a)). Esto se debe a que, para este ejemplo donde se escala la señal con un factor de **1.5**, **R_a** vale **341** y **R_s** vale **512**. El resultado de traslapar los tramos de las primeras dos iteraciones de la gráfica b) (después de que hayan sido alteradas por el método Escalamiento en Tiempo) se muestra en la gráfica c).

La manera de obtener más valores de fase es avanzando más puntos en la etapa de síntesis, sin embargo, hay que recordar que no sólo se trata de avanzar más puntos, si no que también la fase debe ser alterada (escalada) previamente. A la hora de traslapar los tramos consecutivamente, es cuando se obtienen más valores de fase de la señal original (aproximadamente, utilizando este método de escalamiento).

Las figuras 41, 42 y 43 muestran las iteraciones tercera, cuarta y quinta. La figura 44 muestra un acercamiento de la gráfica c) de la figura 43 (quinta iteración).

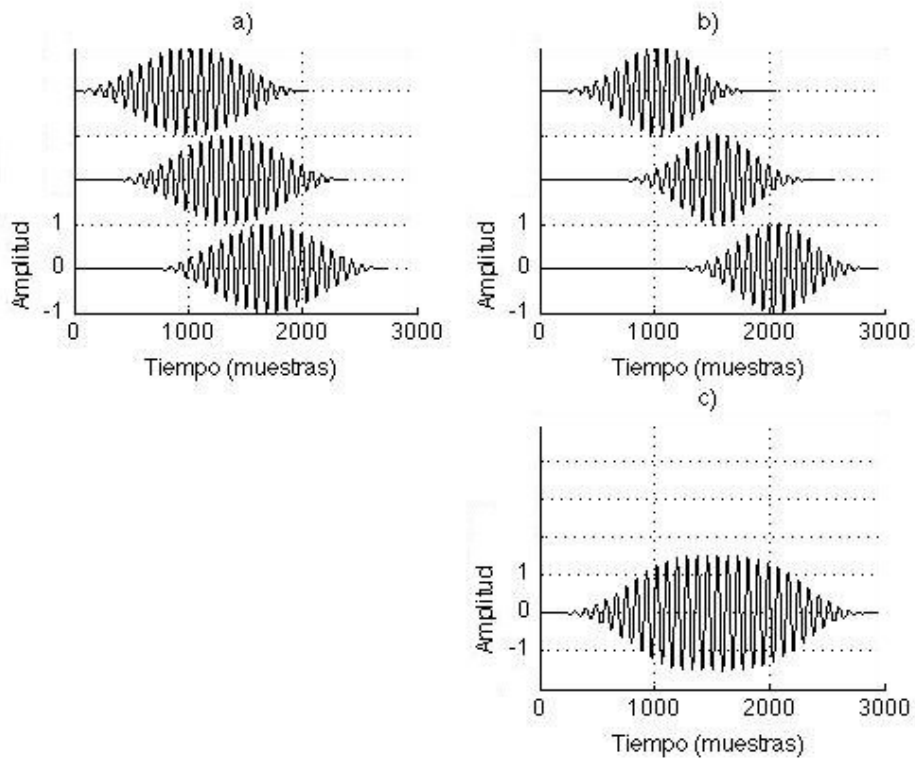


Figura 41. Tercera iteración del método Escalamiento en Tiempo

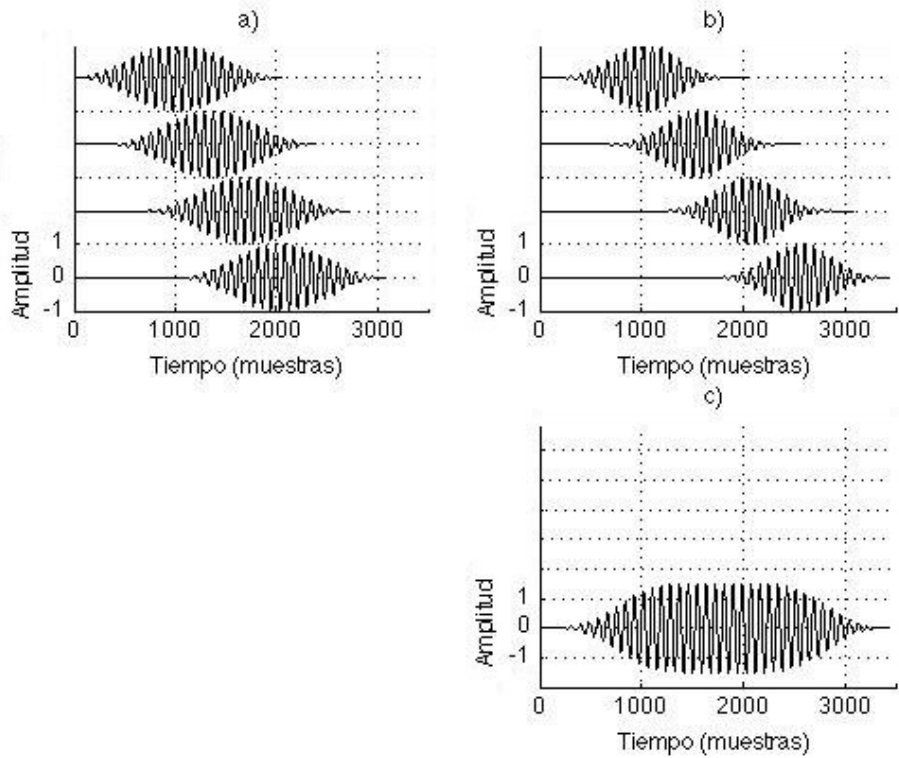


Figura 42. Cuarta iteración del método Escalamiento en Tiempo

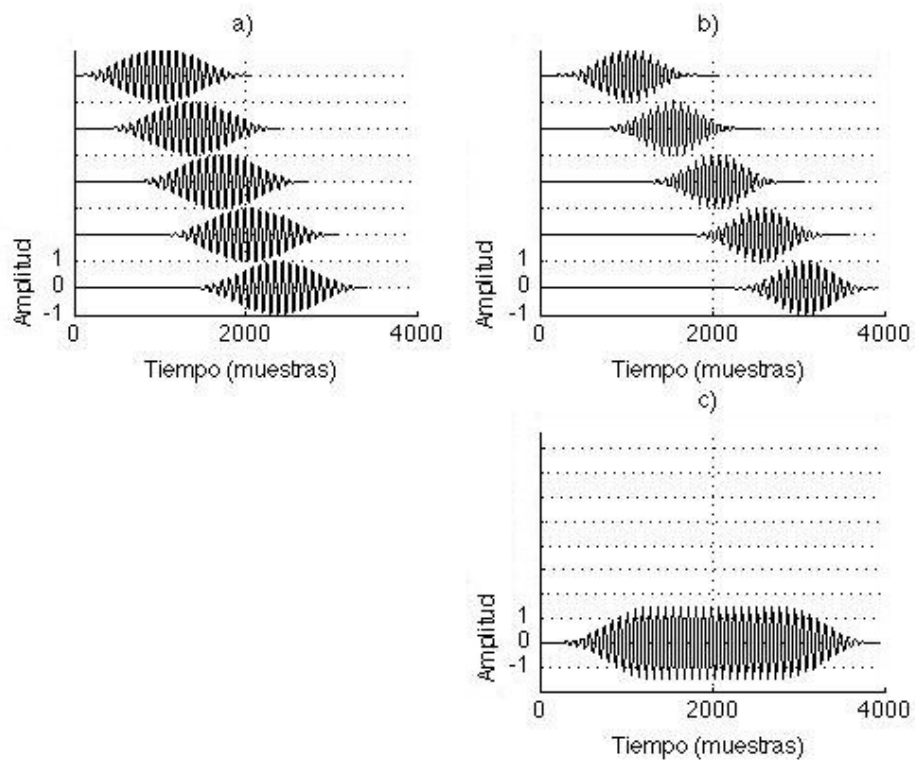


Figura 43. Quinta iteración del método escalamiento en tiempo

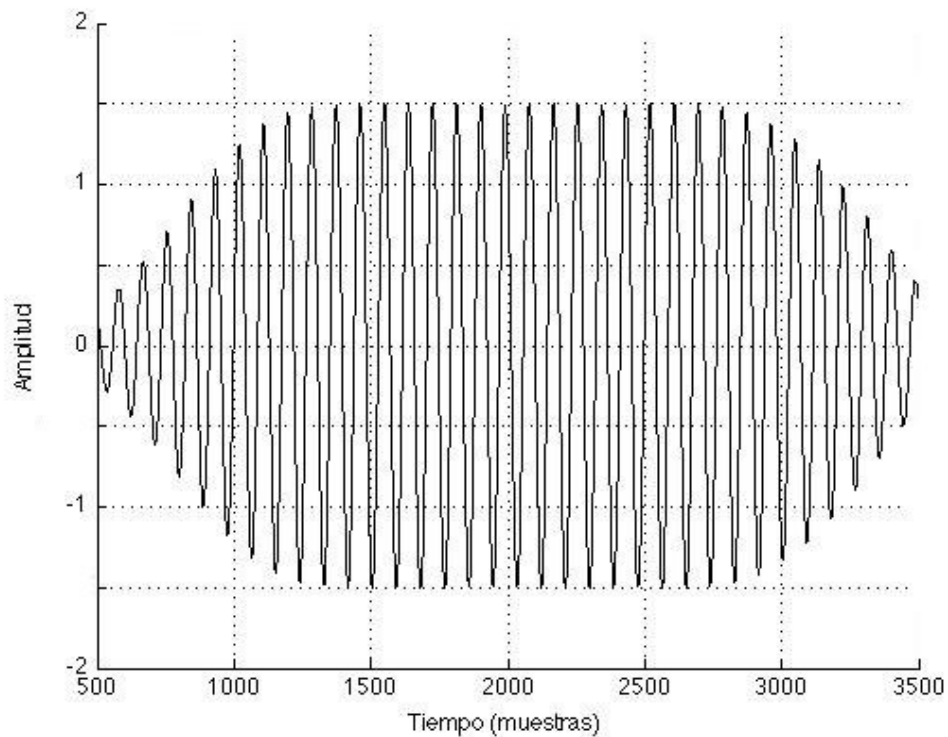


Figura 44. Toma ampliada de la gráfica c) de la figura 43

La figura 44 muestra cómo ya se puede apreciar una señal senoidal resultado de traslapar los tramos en la quinta iteración. Es decir ya se está formando la señal original, pero en este caso alargada en el dominio del tiempo, con una ligera distorsión como lo muestra la figura 37.

6.3 Ecuación

La respuesta en frecuencia de la señal que se ocupa para ilustrar la implementación del ecualizador en el código **Ecua.m**, se muestra en la figura 45, la cual es una señal compuesta por ondas senoidales de frecuencias: 500, 1000, 2000, 4000 y 8000 Hz; muestreada a $F_s = 44,100$ Hz por segundo y de duración $t = 0.1$ segundos cada una.

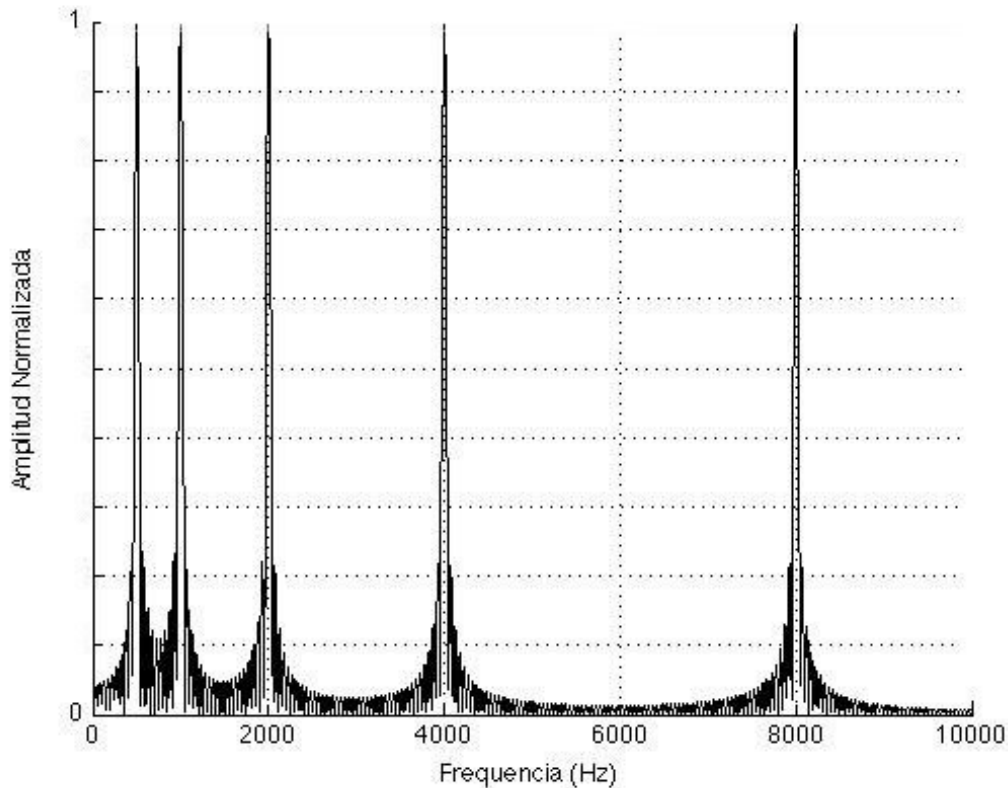


Figura 45. Señal compuesta por ondas senoidales

La respuesta impulsiva y frecuencial de la ecualización se muestra en la figura 46; ésta está formada por 2 filtros de media de octava, 1 filtro de octava y 2 filtros de media de octava. Esta ecualización amplifica el rango 1-4 kHz a 20 dB. Los valores de las ganancias para lograr la amplificación deseada son los siguientes: 10, 2, 16, 2 y 10; éstos se especifican en el código **filteringEph.m**. La señal resultante después de la ecualización se muestra en la figura 47; tanto en el dominio temporal como en el frecuencial.

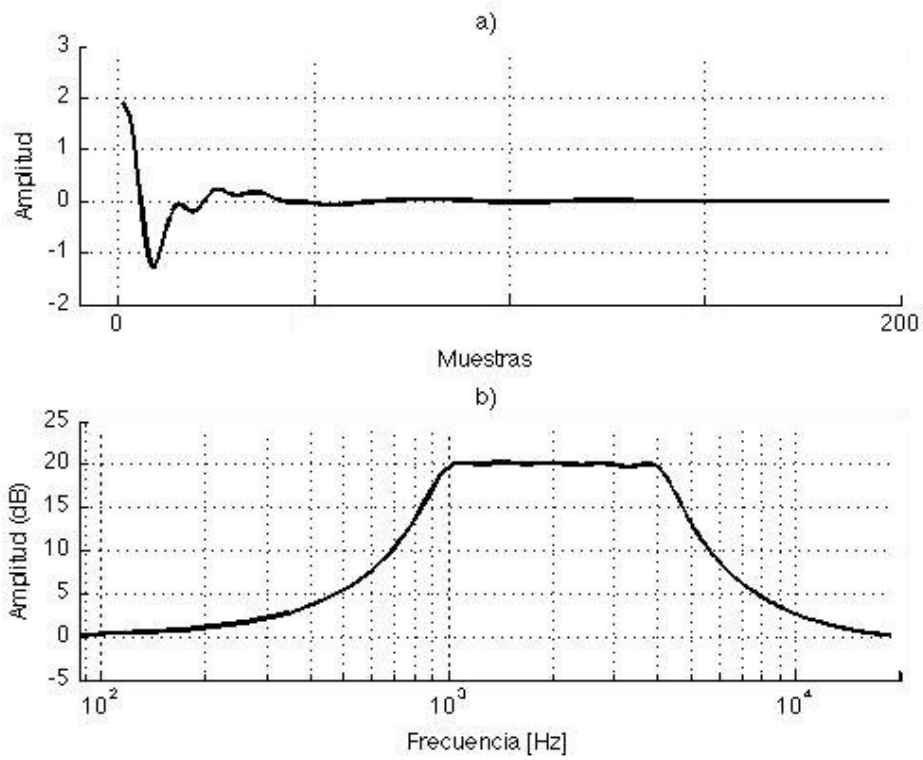


Figura 46. Respuestas impulsiva a) y frecuencial b) de la equalización

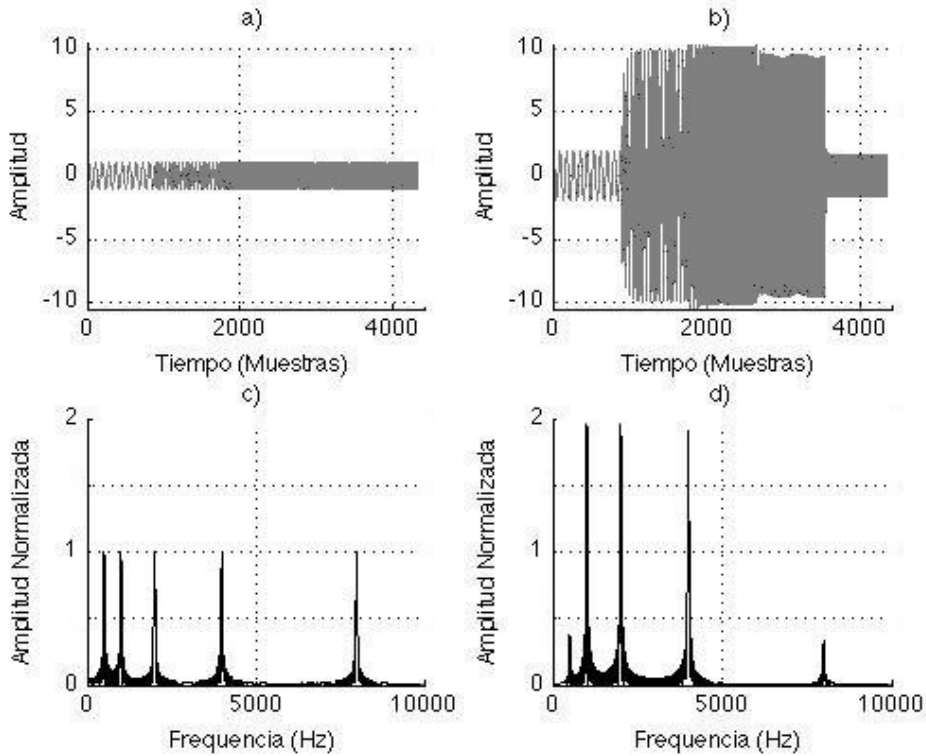


Figura 47. Señal equalizada

De la figura 47 tenemos lo siguiente. La gráfica a) muestra la señal inicial completa, en el dominio del tiempo. La gráfica b) muestra la señal ecualizada, de donde podemos ver que los tramos internos (bandas de frecuencia 1000, 2000 y 4000 Hz) están amplificados a 20 decibeles, también podemos ver que los tramos extremos (bandas de frecuencia 500 y 8000 Hz) están amplificados ligeramente, debido a la naturaleza del ecualizador. La gráfica c) es la respuesta en frecuencia de la señal inicial. La gráfica d) es la respuesta en frecuencia de la señal una vez que ha sido ecualizada (amplitud normalizada), en donde podemos ver cómo únicamente las componentes 1, 2 y 4 kHz son amplificadas.

La gráfica b) muestra cómo la amplitud pico, que originalmente valía 1 Volt, ahora vale 10 Volts. Si queremos representar esos 10 Volts en decibeles, entonces la referencia que se utiliza es de 1 Volt. Teniendo así $20\log_{10}(10V/1V) = 20dB$, que tradicionalmente se representa como $20dBV$. De esta manera, podemos confirmar que la señal original sí se amplificó a 20 decibeles.

La figura 48 muestra un acercamiento y traslape de la figura 47 entre las gráficas a) contra b) y entre las gráficas c) contra d); la gráfica de línea gruesa representa la señal inicial y la gráfica de línea delgada representa la señal resultante. En particular, se hace un acercamiento de la componente 2 kHz. La finalidad es apreciar con detalle la ecualización de dicha señal. A comparación del escalamiento en tiempo, la señal resultante por la ecualización no se distorsiona.

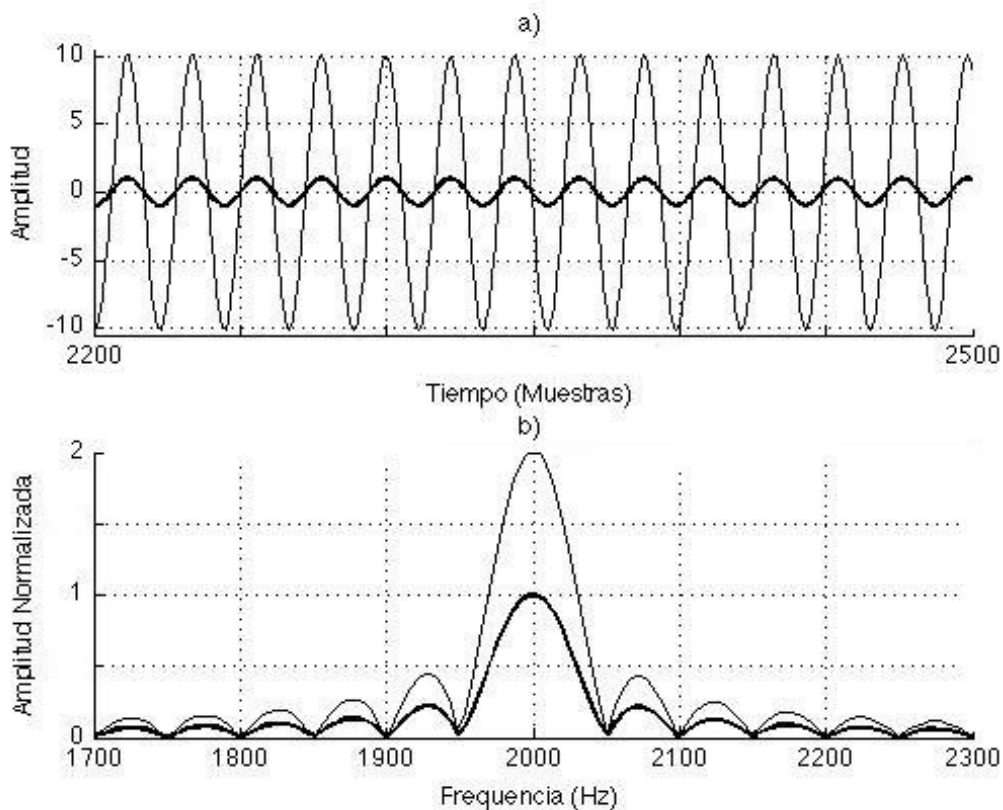


Figura 48. Toma ampliada de la segunda componente de la señal inicial contra la señal resultante

6.4 Aplicación HIPELDA

Para probar la aplicación HIPELDA se ocupa una pista de audio en formato WAV de 12 segundos de duración **grabada a un canal**. El nombre de la pista es **track2c.wav** y contiene grabadas cuatro palabras bisilábicas, las cuales son: cierta, cifra, veinte, dejo.

Primero se compila y ejecuta el código del programa **HIPELDA.java** que se encuentra en la carpeta **hipelda_j** [c], el cual arroja cinco archivos en formato texto, éstos son: **x.txt**, **x_out.txt**, **h.txt**, **Habs.txt** y **filterign.txt**. Estos archivos se pasan a la carpeta **hipelda_m** [b] en donde se encuentra el código **HIPELDA.m**; posteriormente se ejecuta este código para que, además de realizar los mismos cálculos que el programa en Java, éste lea los archivos texto y muestre los resultados de Java y MATLAB encimados en gráficas (figura 49). De esta manera, se puede verificar que los resultados obtenidos con ambas herramientas son los mismos. Verificando así, que los resultados obtenidos en Java son fidedignos.

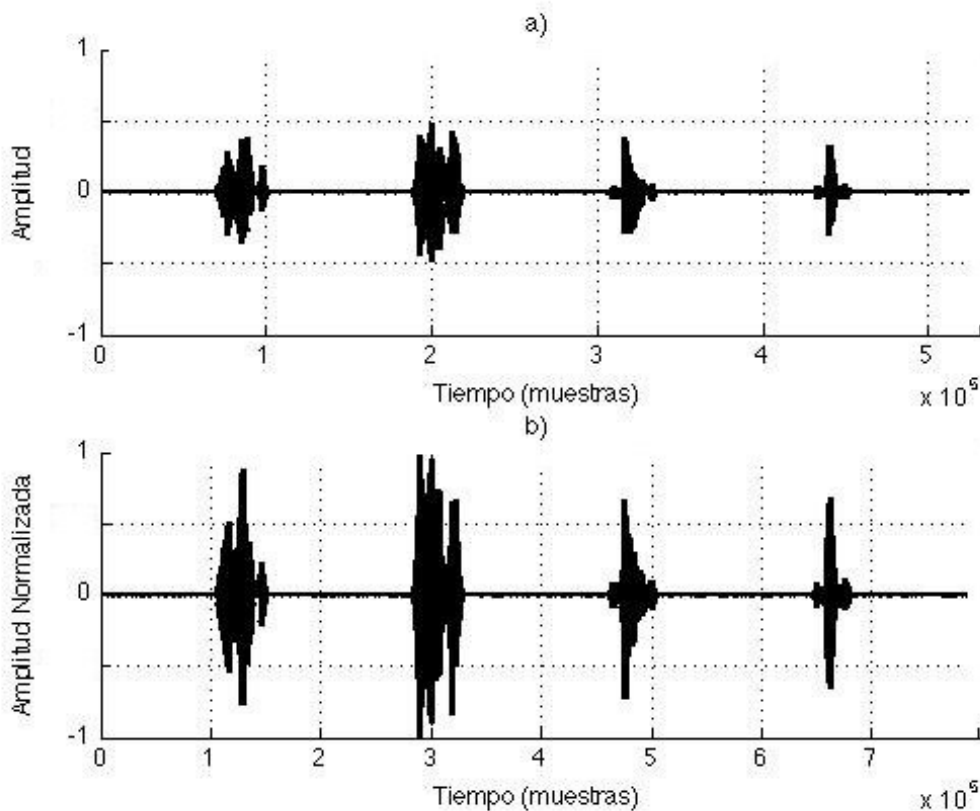


Figura 49. Señales inicial a) y final b) después del escalamiento

De la figura 49 tenemos lo siguiente. Ambas gráficas a) y b) muestran los resultados obtenidos con Java y están encimados con los obtenidos con Java. Aparentemente los resultados son los mismos, sin embargo, para estar seguros de que realmente sí son los mismos, posteriormente se hará un acercamiento de la figura que representa la segunda palabra.

La gráfica a) representa la señal leída de la pista de audio, la cual dura en tiempo 12 segundos y está formada por 5.3×10^5 muestras aproximadamente. La gráfica b) representa la señal escalada en tiempo, la cual ahora es de duración 18 segundos y contiene 8.1×10^5 muestras aproximadamente. También se puede notar que la amplitud es mayor que la mostrada en el inciso a), esto se debe a que parte del método de escalamiento involucra normalizar la señal.

Con la finalidad de apreciar mejor la similitud de los resultados obtenidos por MATLAB y Java, la figura 50 muestra un acercamiento de las gráficas de la figura 49. En particular se hace un acercamiento de la segunda palabra.

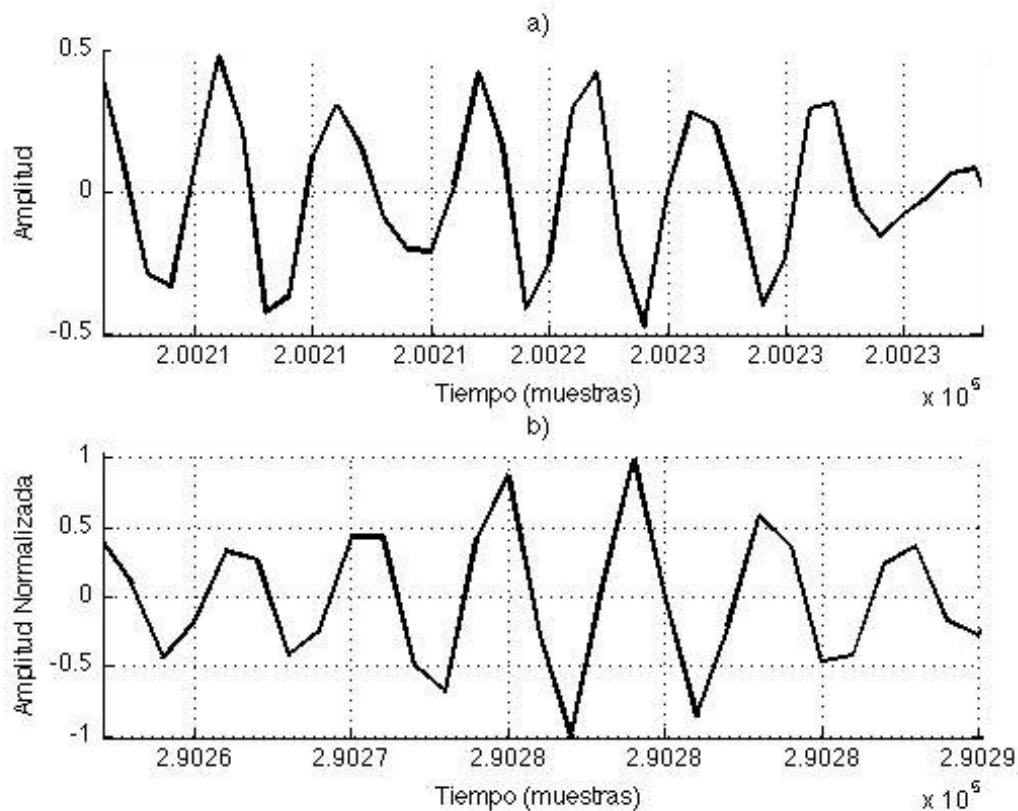


Figura 50. Toma ampliada de la figura 49

La figura 50 muestra cómo en ninguna de las dos gráficas existe alguna disparidad entre los resultados obtenidos por MATLAB ni por los obtenidos por Java. Es decir, los resultados obtenidos por ambas herramientas son tal cual los mismos.

La figura 51 muestra la ecualización de la señal después de haberla escalado; la gráfica a) muestra la señal escalada únicamente y la gráfica b) muestra la señal escalada y ecualizada. De la gráfica b) se puede notar cómo la amplitud de la señal aumenta debido a la ecualización. Igualmente, las gráficas son obtenidas por MATLAB y encimadas con las obtenidas por Java. El acercamiento se hace en la figura 52, donde se vuelve a hacer un acercamiento de la segunda palabra.

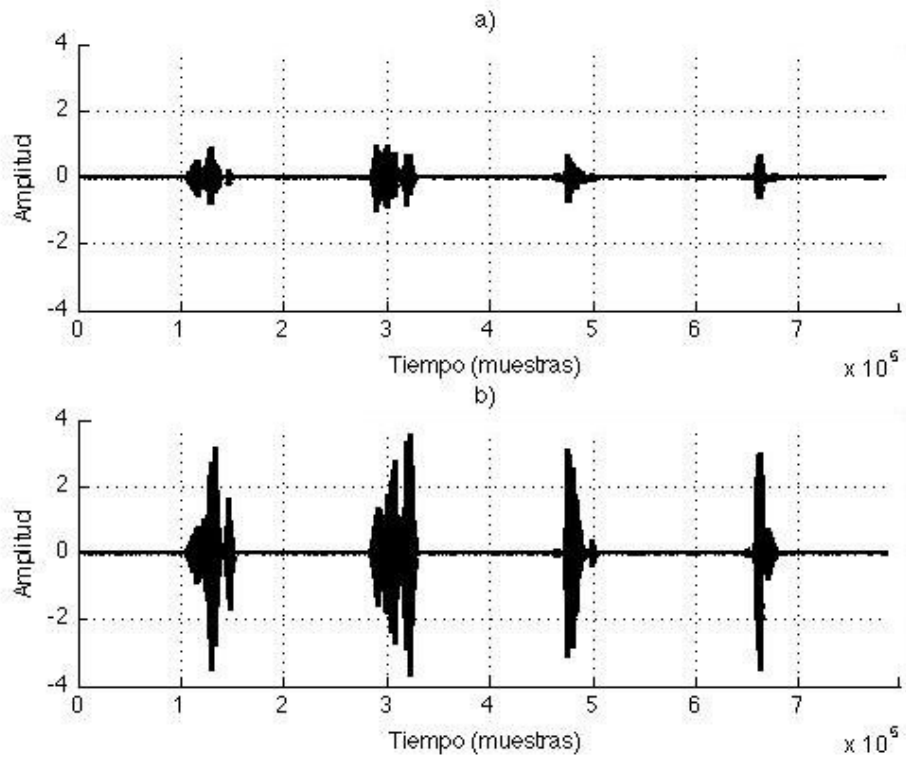


Figura 51. Señales inicial a) y final b) después del escalamiento y ecualización

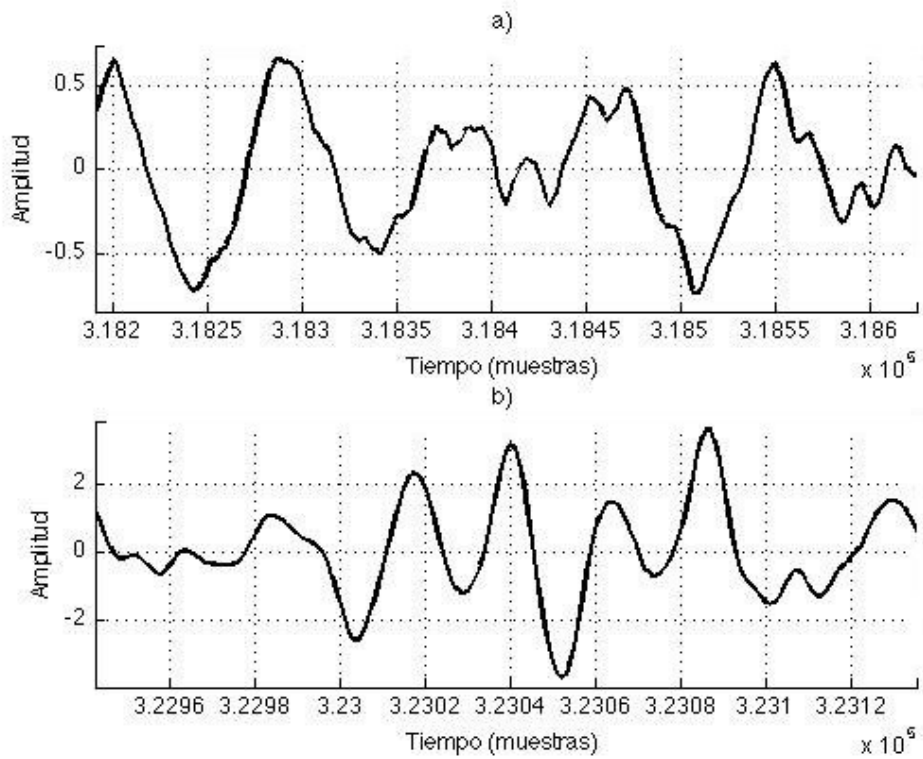


Figura 52. Toma ampliada de la figura 51

La señal sonora de una palabra es compleja compuesta por un gran número de ondas senoidales, entonces al comparar los espectros en el dominio frecuencia contra amplitud de la señal inicial contra la señal ecualizada, éstos no son muy claros en el sentido de notar la diferencia de la ecualización. Una manera de percibir la diferencia es al graficar las señales en el dominio tiempo contra frecuencia. Así, las frecuencias que fueron resaltadas se distinguirán de entre las demás. La figura 53 muestra esta representación, cada franja vertical representa una palabra.

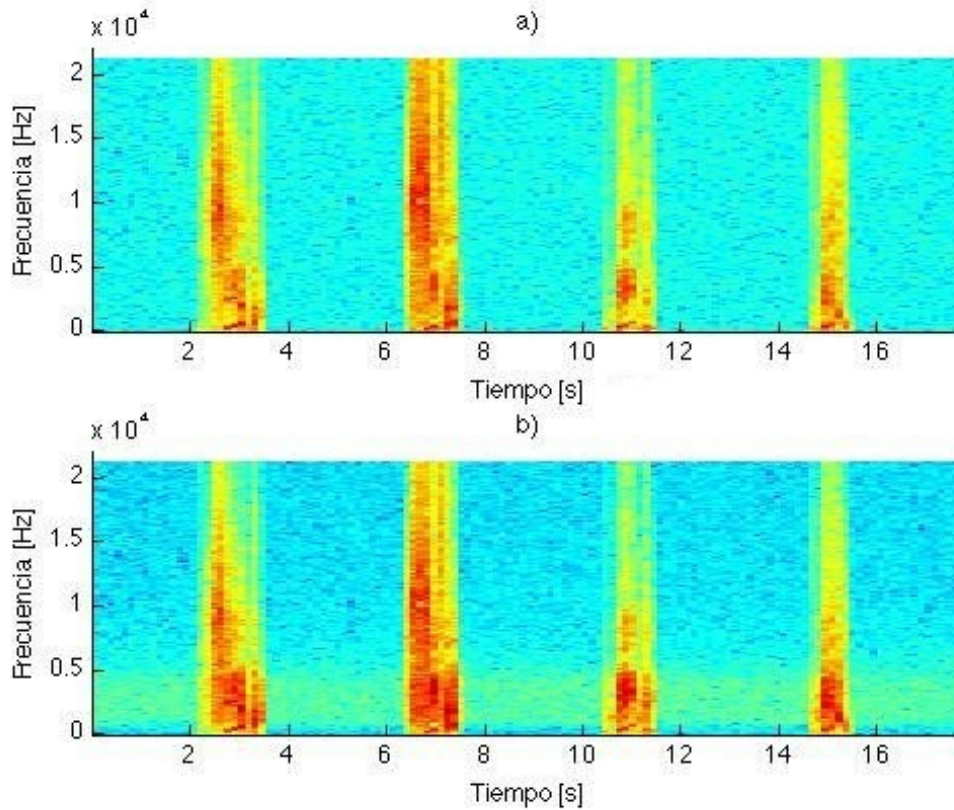


Figura 53. Representación de las señales inicial a) y resultante b) en el dominio tiempo contra frecuencia

De la figura 53, la gráfica b) muestra cómo los colores de las franjas verticales son más intensos en el rango 0-5 kHz, es decir resaltan más. El color rojo indica mayor intensidad (20 dB), el color amarillo indica intensidad media y el color verde claro indica cero intensidad (0 dB). Esto confirma la ecualización que se le hizo al rango frecuencial 1-4 kHz.

7. Aplicación HIPELDA

7.1 Esquema

El nombre de la aplicación HIPELDA se debe a algunas de las iniciales del nombre del proyecto (Herramienta Interactiva para Mejorar el Desempeño en la Percepción del Lenguaje en Débiles Auditivos). La aplicación fue programada utilizando el lenguaje Java y el código del programa principal se llama **HIPELDA.java** que se encuentra en la carpeta *hipelda_j*[c]. El código principal está dividido en seis secciones que aparecen en el siguiente orden: Carga del Archivo, Escalamiento, Ecuación Predefinida, Ecuación Personalizada, Reproducción y Guarda de la Señal.

La Ecuación Predefinida se refiere a la mencionada en la sección 1.2 (Contenido de la Tesis), la cual es de 20 dB en la banda de frecuencia 1-4 kHz. Se incluye la Ecuación Personalizada, en caso de que el usuario desee hacer pruebas con diferentes valores de ganancias; esta ecuación incluye dos bandas de frecuencia más: 500 y 8000 Hz. También se incluye la función Reproducción para escuchar cómo va cambiando la señal después de modificarla. Finalmente, también se incluye la función Guarda de la Señal en caso de que se desee guardar la señal después de modificarla y así poder reproducirla en otro reproductor.

El código del programa principal **HIPELDA.java** se muestra a continuación.

```
import java.io.File;
import javax.sound.sampled.AudioFormat;

public class Hipelda
{
    public static void main (String[] args)
    {
        // Carga del Archivo
        fileChooser fCh= new fileChooser();
        File file_in = fCh.file;
        storage S = new storage(file_in);
        double x[] = S.x;
        double fs = S.Fs;
        double t = S.d;
        System.out.println("\n t = "+t+" seg");
        AudioFormat aF = S.audioFormat;

        // Escalamiento
        int Ra = 228;
        timeSR TSR = new timeSR(x,fs,Ra,aF);
        double x_out[] = TSR.x_out;
        double tm = TSR.d;
        System.out.println("\n tm = "+tm+" seg \n");

        // Ecuación Predefinida
        emphasis Em = new emphasis(x_out, fs);
        double x_out_eph[] = Em.x_out_eph;

        // Ecuación Personalizada
        //int fb[] = {1, 2, 3, 4, 5};
        //int g[] = {0, -4, 10, 3, 0};
        //equa Eq = new equa(x_out,fb,g,fs);
        //double x_out_eq[] = Eq.x_out_eq;

        // Reproducción
        new playing(x_out,aF);
    }
}
```

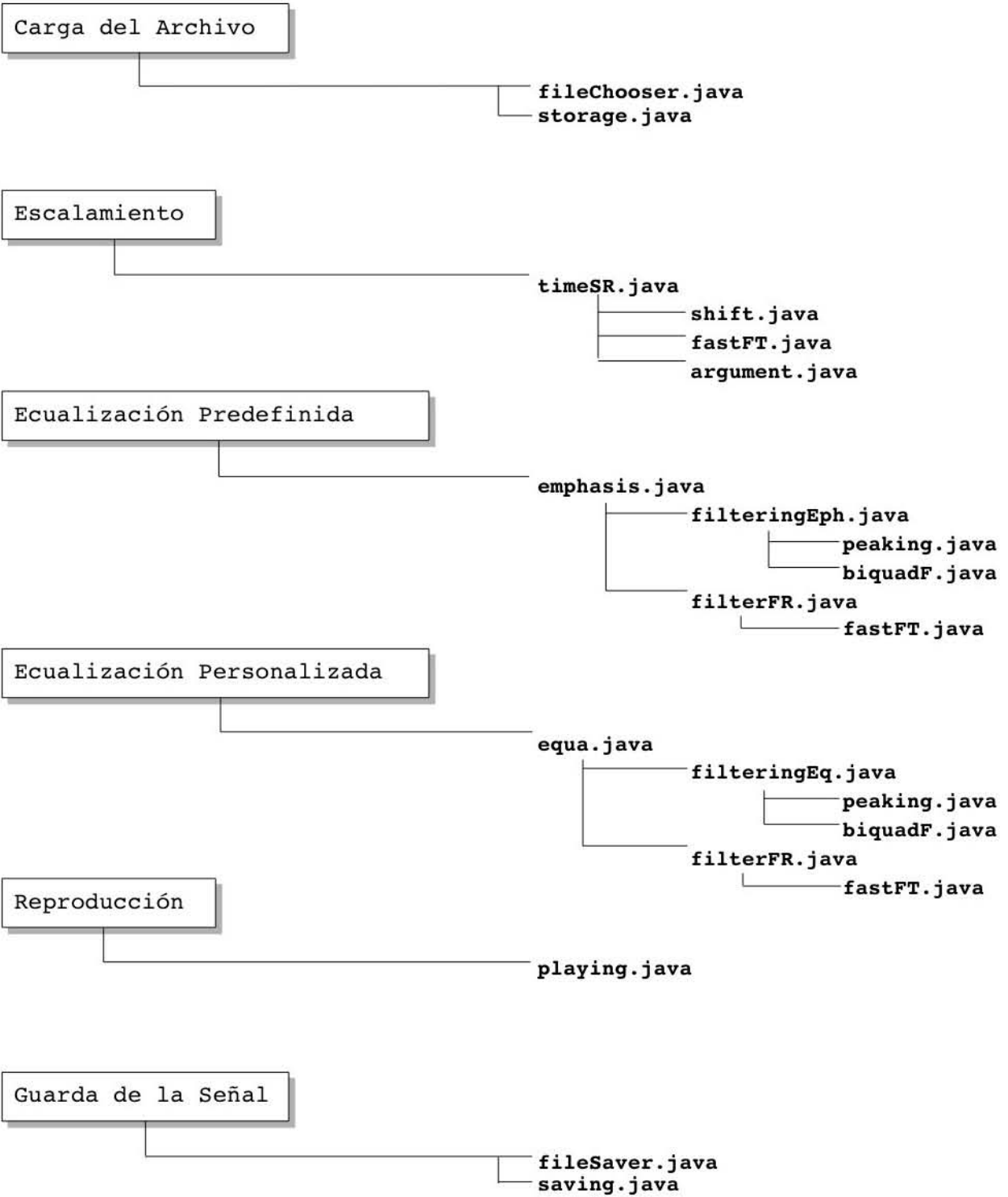
```

    //// Guarda de la Señal
    //fileSaver fS = new fileSaver();
    //File file_out = fS.path;
    //new saving(x_out_eph,aF,file_out);
}
}

```

En la sección Carga de Archivo, el código del programa **storage.java** almacena el archivo de audio, el cual es seleccionado con un Selector de Archivos (tiene la función de un navegador de sistemas de archivos ejecutado por la aplicación). Al ejecutarse el programa el selector se abre y uno puede navegar entre las carpetas hasta encontrar el archivo WAV deseado. La sección Guarda de la Señal también utiliza el Selector de Archivos para seleccionar la carpeta en donde se desee guardar el archivo, al cual se le debe incluir la extensión .wav. El código del programa de la sección Guarda de la Señal se llama **saving.java**.

En las secciones Escalamiento y Ecuilización Predefinida, los código de los programas que llevan a acabo esas funciones son **timeSR.java** y **emphasis.java**, respectivamente. En la sección Ecuilización Personalizada, el código del programa **equa.java** equaliza la señal de acuerdo a los valores de ganancia definidos en el arreglo **g[]**. En la sección Reproducción, el código del programa que realiza esa tarea es **playing.java**. Cada una de estas secciones contiene sus correspondientes programas, los cuales a su vez contienen subprogramas. El siguiente esquema muestra el flujo que sigue la aplicación.



Las fuentes de los sitios y páginas electrónicas de donde se obtuvieron estos códigos se muestran al final del documento, en la sección Códigos Java y MATLAB.

Cada uno de estos códigos, aparte de estar escritos en Java también están programados en MATLAB, excepto las funciones: fileChooser y fileSaver. El contenido del código **HIPELDA.m** que se encuentra en la carpeta *hipelda_m* [b], se muestra a continuación.

```
close all
clear all
clc;

%% Carga del Archivo
file_in = 'track2c.wav';
[x fs] = storage(file_in);

%% Escalamiento
Ra = 341;
x_out = timeSR(x,fs,Ra);

%% Ecualización Predefinida
x_out_eph = emphasis(x_out,fs);

%% Ecualización Personalizada
%fb = [1 2 3 4 5];
%g = [0 -4 10 3 0];
%x_out_eq = equa(x_out,fb,g,fs);

%% Reproducción
playing(x_out,fs)

%% Guarda de la Señal
%wavwrite(x_out_eph,fs,16,'nombre.wav');
```

7.2 Interfaz

La interfaz está dividida en tres secciones principales, estas son: ENTRADA SALIDA, ESCALAMIENTO y ECUALIZACIÓN. La figura 54 muestra con detalle las secciones de la interfaz y sus respectivas componentes.

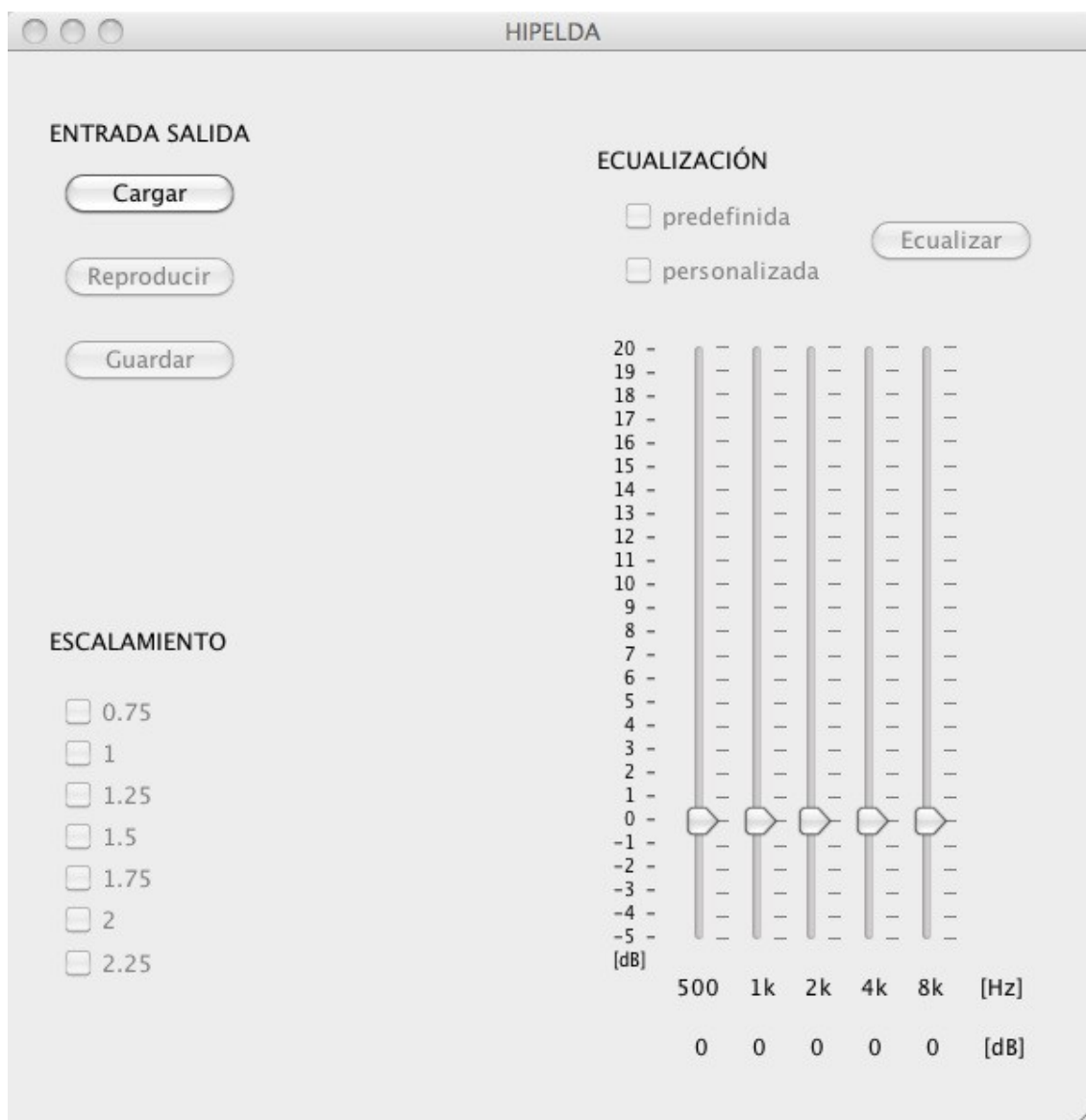


Figura 54. Interfaz HIPELDA

El flujo que sigue el funcionamiento de la interfaz obedece a una serie de validaciones. Por ejemplo, al ejecutar la interfaz todos los botones y casillas de control están deshabilitadas excepto el botón Cargar. Una vez que se carga el archivo, todas las demás funciones se habilitan. La siguiente sección 7.3 muestra con detalle el funcionamiento de dicha interfaz.

7.3 Funcionamiento

Con la finalidad de mostrar cómo funciona la aplicación, a continuación se muestran una serie de ejemplos clave para llevar a cabo dicho funcionamiento. Otro punto importante a abordar es, aprender a interpretar los resultados obtenidos al utilizar esta aplicación, de tal manera que los mismos no sean interpretados erróneamente. Todos los archivos de la aplicación se encuentran en un carpeta de nombre *InterfazHIPELDA* [d]. Los archivos ejecutables para las plataformas Windows, Linux y Mac se encuentran en las carpetas: *estuche_HIPELDA_w* [e]; *estuche_HIPELDA_l* [f]; *estuche_HIPELDA_m* [g], respectivamente. Para abrir la aplicación, simplemente ejecutamos dicho archivo.

Al abrir la aplicación se mostrarán las tres secciones principales. La sección ENTRADA SALIDA está formada por tres botones que cumplen con las siguientes funciones: Cargar, Reproducir y Guardar. La sección ESCALAMIENTO contiene siete casillas de control; cada una corresponde a un factor de escalamiento distinto. La sección ECUALIZACIÓN se explica posteriormente.

Para empezar, cargamos una pista de audio presionando el botón Cargar; inmediatamente después, se abre un selector que nos da la opción de navegar entre las carpetas. Una vez encontrado el archivo, simplemente se lo selecciona y se presiona el botón *Open file* para así cargar la pista. La figura 55 muestra un ejemplo que cargar una pista de nombre **track2b.wav**.

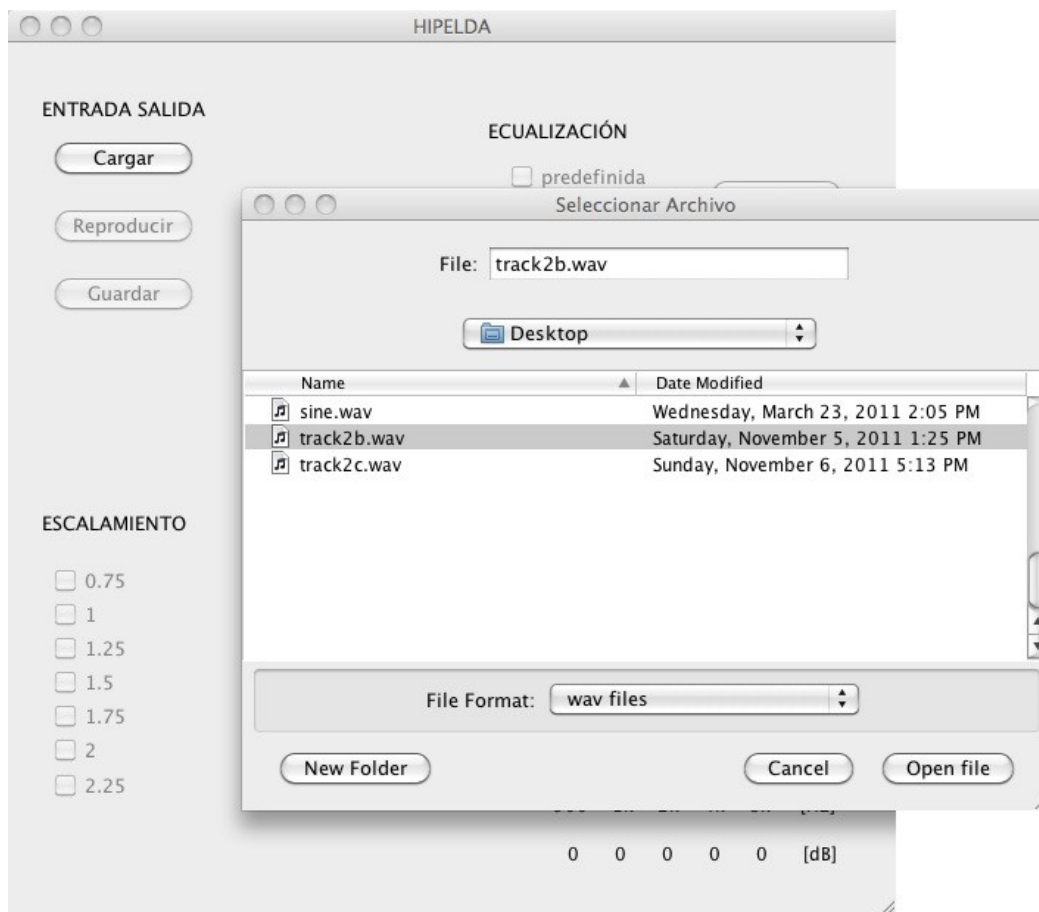


Figura 55. Selector de la interfaz

En este caso, trabajamos con la pista **track2b.wav**, la cual tiene el mismo contenido que la pista **track2c.wav**. La diferencia radica en que la pista **track2b.wav** está **grabada a dos canales**, requisito que es necesario para el funcionamiento de la aplicación. El resultado después de cargar el archivo se muestra en la figura 56.

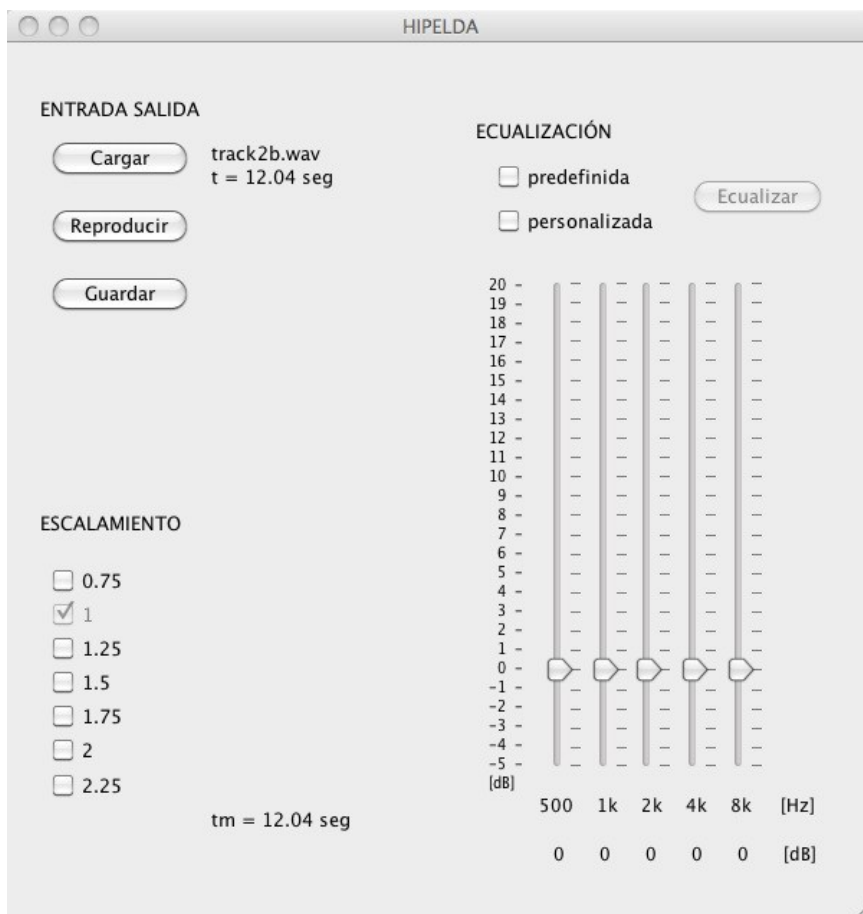


Figura 56. Pista de audio cargada

De la figura 56 podemos ver que el nombre de la pista es **track2b.wav** y su duración es de **t = 12.04 s**; estos resultados se muestran del lado derecho del botón Cargar. Por otro lado, el factor de escalamiento **1** se selecciona automáticamente indicando que la señal no ha sido escalada o de igual manera está escalada con el factor **1**. Finalmente, el tiempo de duración de la señal escalada **tm = 12.04 s** se muestra en la parte inferior izquierda que en este caso, es la misma que el tiempo **t** indicando que la señal no ha sido escalada.

En este punto, el resto de las funciones se habilitan automáticamente e independientemente de que los datos de la pista sean modificados o no, se pueden reproducir e incluso guardar con otro nombre. Para escalar una señal, simplemente seleccionamos la casilla correspondiente y automáticamente se ejecuta la función; al terminar el escalamiento, la casilla se queda seleccionada y deshabilitada. Si queremos escalar la señal con otro factor, simplemente seleccionamos la casilla correspondiente. El tiempo de ejecución que dura escalar la señal dependerá del tamaño de la pista; entre más pequeña sea, más rápida será la duración de ejecución del escalamiento.

La sección ECUALIZACIÓN consta de dos casillas de control, un botón tipo interruptor y cinco botones deslizantes. Las casillas cumplen con la función de ecualizar la señal de audio ya sea Predefinidamente o Personalmente; Predefinidamente quiere decir que la señal será amplificada a 20 dB en el rango de frecuencia 1 a 4 kHz; Personalmente implica que el usuario puede seleccionar los valores de ecualización por cada banda frecuencial en el rango de valores de -5 a 20 dB. Independientemente de cual ecualización se escoja, la correspondiente casilla quedará seleccionada. Si se selecciona la ecualización Predefinida, entonces posteriormente se presiona el botón Ecualizar y listo; si se selecciona la ecualización Personalizada, entonces con los botones deslizantes se seleccionan los valores deseados y después se presiona el botón Ecualizar. Se incluyeron dos bandas frecuenciales más: 500 y 8000 Hz; en caso de que el usuario tenga la necesidad de utilizarlas.

A continuación la figura 57 muestra un ejemplo en donde la señal es escalada con un factor de 1.5 y ecualizada de la manera **Predefinida**.

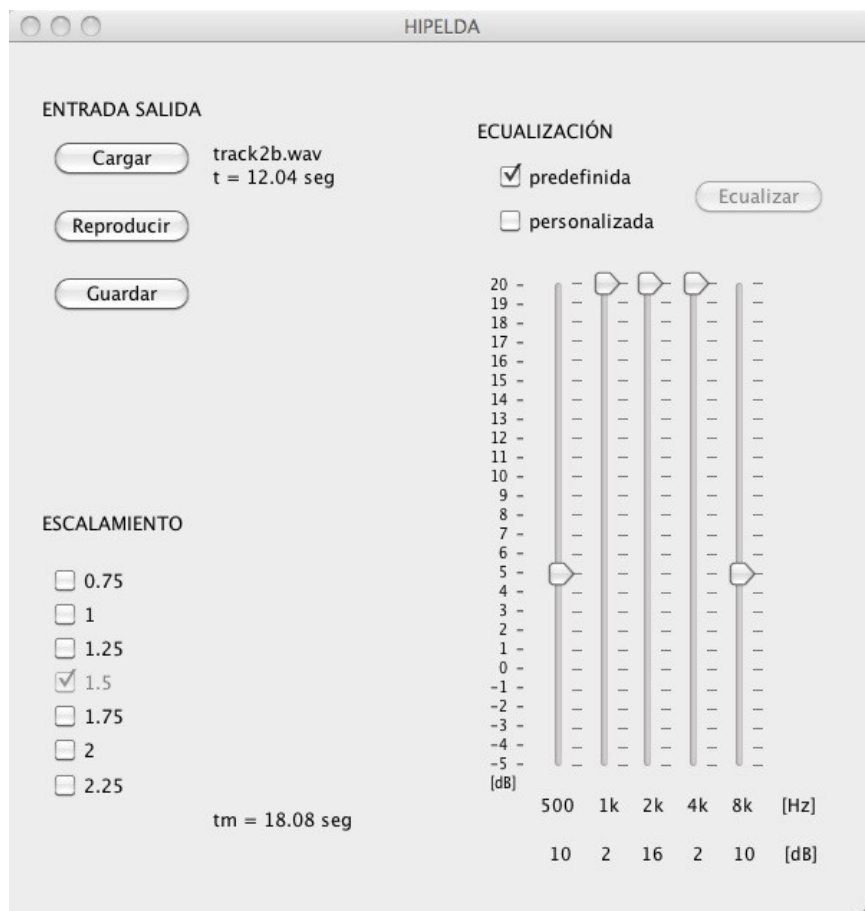


Figura 57. Señal escalada y ecualizada

El primer detalle que debemos de notar de la figura 57 es, que el tiempo de duración de la señal escalada es de **tm = 18.08 seg**, lo cual quiere decir que la señal de audio fue alargada 50 % más y esto implica que la misma se escuchará 50 % más lentamente.

Por otro lado, hay que aclarar que los botones deslizantes tienen dos funciones; una es seleccionar los valores de ecualización por cada banda y otra es representar la respuesta frecuencial del ecualizador (filtrado). Después de presionar el botón Ecualizar, los botones deslizables se reacomodan de tal manera que toman la forma de la gráfica de la respuesta frecuencial del ecualizador. La figura 57 muestra cómo los botones se reacomodan debido a que seleccionamos la ecualización predefinida, dichos botones toman la forma de la respuesta frecuencial mostrada en la gráfica a) de la figura 46. De la figura 46 podemos ver que el valor de la ganancia de las bandas frecuenciales 1, 2 y 4 kHz es de 20 dB y el valor de las bandas frecuenciales 500 y 8000 Hz es de 5 dB; estos mismos valores son los mostrados en la figura 57, representando así la respuesta frecuencial de esta ecualización.

Ahora se muestra un ejemplo en donde se selecciona la ecualización **Personalizada**. Primero seleccionamos la correspondiente casilla; inmediatamente después los botones deslizables se habilitan, tendiendo así la opción de seleccionar las ganancias por cada banda frecuencial. Para mostrar un ejemplo sencillo, ajustamos el botón que le corresponde a la banda frecuencial de 2 kHz a 15 dB, así como se muestra en la figura 58.

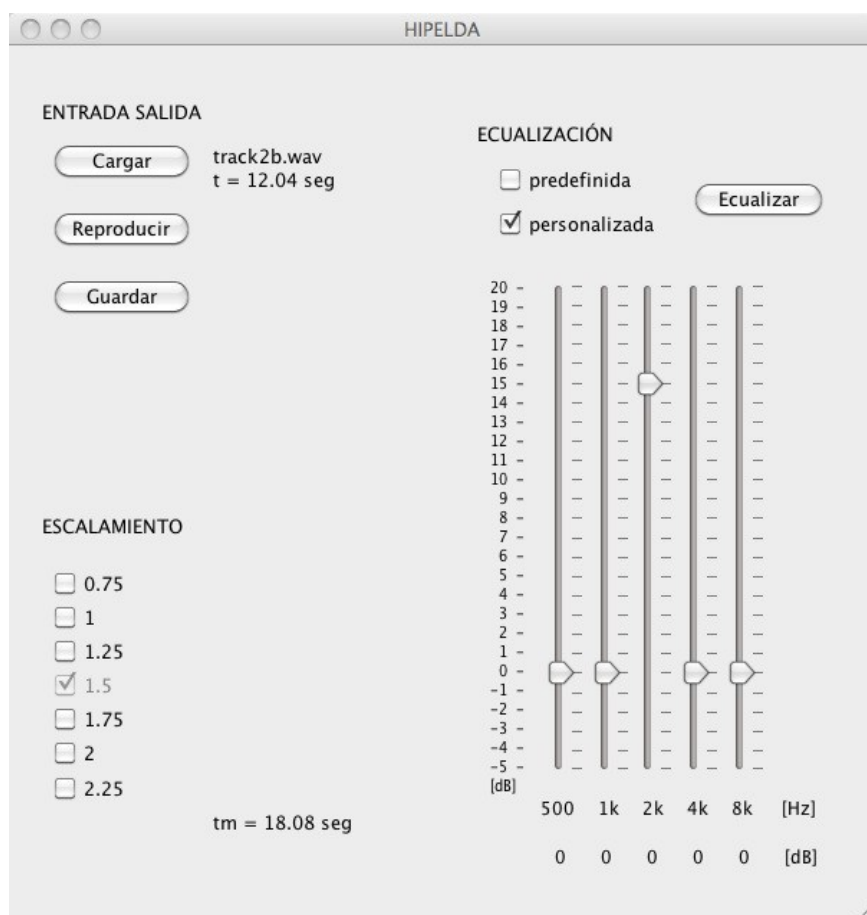


Figura 58. Ganancia de 15 dB en la banda frecuencial 2 kHz

De acuerdo a la sección 4.4 (Implementación del Escalamiento en Tiempo), al diseñar la ecualización predefinida se utilizaron los siguientes valores de ganancias por cada banda frecuencial: 10, 2, 16, 2 y 10; éstos números se muestran abajo del rango de frecuencias de la figura 57. La finalidad de mostrar estos números es para indicar qué valores son necesarios para obtener dicha ecualización. Lo ideal sería que simplemente al ingresar el valor 20 para cada banda frecuencial obtuviéramos la ganancia resultante mostrada en la gráfica a) de la figura 46, sin embargo no lo es así ya que las respuestas frecuenciales de las bandas interactúan entre si. Se tendría que buscar una ecuación que nos arrojara los valores 10, 2, 16 y 10 cuando uno ingresa el valor 20 en cada banda frecuencial; entonces para obtener la respuesta frecuencial deseada, se tuvo que probar ingresando diferentes valores hasta obtener dicha respuesta frecuencial.

Respecto a la ecualización Personalizada, los números seleccionados por cada botón deslizable serán mostrados también, de esta manera se lleva un control de cuales valores fueron seleccionados y estos aparecerán cuando presionemos el botón Ecualizar. En el ejemplo mostrado en la figura 59, al presionar el botón Ecualizar el valor 15 aparecerá en la parte inferior de la respectiva banda de frecuencia, además, los botones se reacomodarán para tomar la forma de la respuesta frecuencial de dicha ecualización.

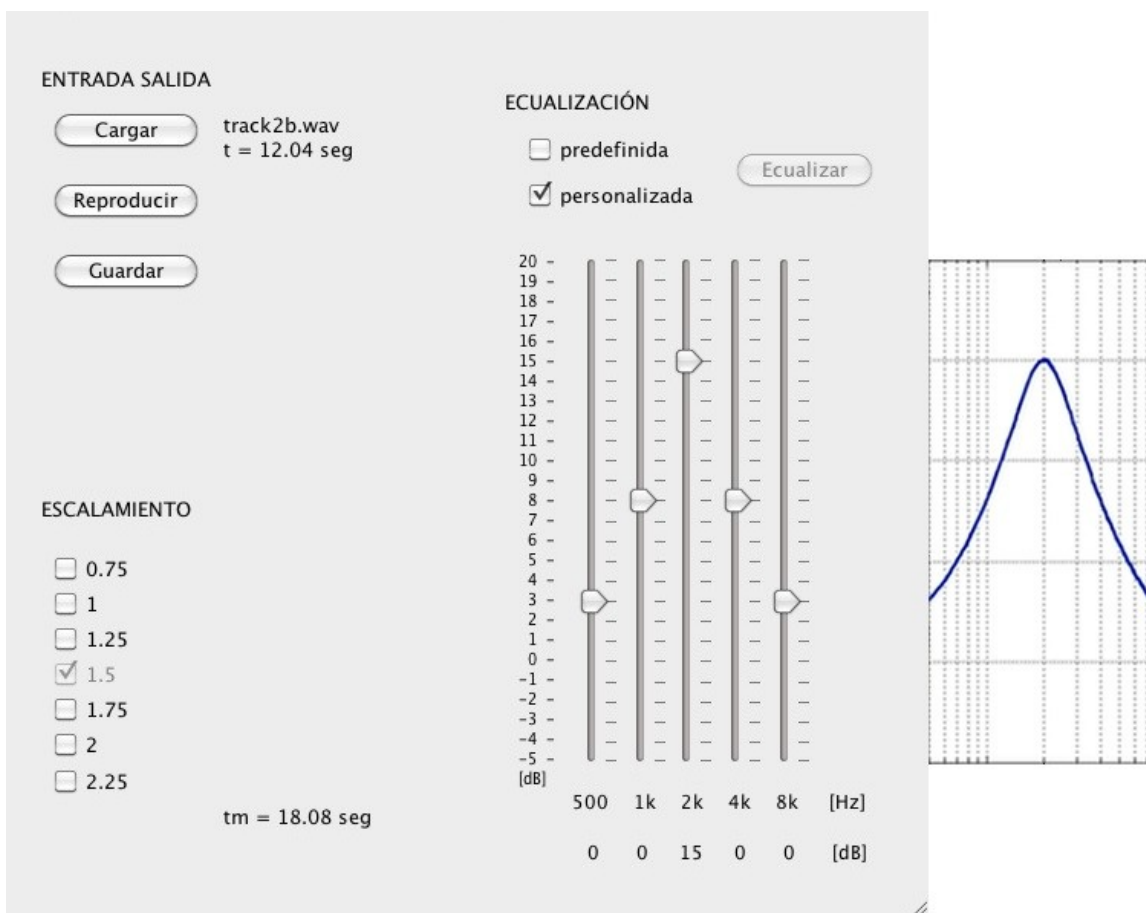


Figura 59. Respuesta frecuencial obtenida al ecualizar la banda frecuencial 2 kHz

La parte derecha de la figura 59 muestra cómo los botones se reacomodan, tomando la forma de la respuesta en frecuencia que obtendríamos al utilizar MATLAB; ésta se muestra a un lado y a la misma altura. El hecho de que el resto de los botones ahora tomen otra posición, no quiere decir que los botones ahora contenga esos valores, simplemente es una representación de cómo la respuesta frecuencial se comporta al haber amplificado la banda en frecuencia 2 kHz a 15 dB. Solamente el botón de la banda en frecuencia 2kHz tomó el valor 15 porque deslizamos ese botón a ese número, y para indicar que ese valor fue seleccionado, en la parte inferior derecha podemos ver dicho valor.

El siguiente paso es mantener la ecualización previa y al mismo tiempo, deslizar los botones de otras bandas. Por ejemplo, ahora deslizaremos los botones de las bandas frecuenciales 1 y 8 kHz a 6 y -5 dB, respectivamente. La figura 60 muestra los botones de dichas bandas deslizados a los valores que se acaban de mencionar.

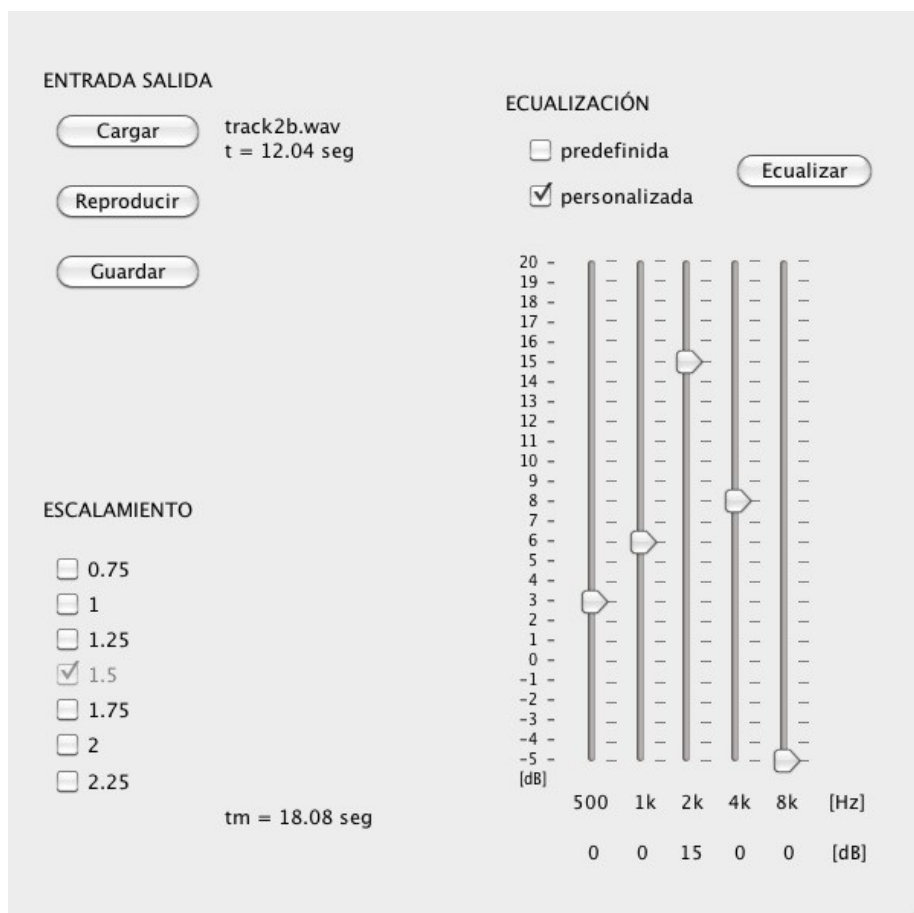


Figura 60. Ajuste de las ganancias de las bandas frecuenciales 1 y 8 kHz

A esta altura ya podemos intuir que al presionar el botón Ecuilizar, los botones de todas las bandas se reacomodaran. El nuevo acomodo representará la forma de la respuesta frecuencial obtenida como resultado de implementar esta nueva ecualización. Es decir, los nuevos valores de ganancias ingresados para estas dos bandas **interactúan** con el valor de ganancia ingresado previamente y como resultado se obtendrá cierta respuesta en frecuencia.

Nuevamente, al presionar el botón Ecuilizar, los botones se reacomodaran para representar la respuesta en frecuencia de esta nueva ecualización. Los valores 6 y -5 aparecerán en la parte inferior de la bandas de frecuencia seleccionadas indicando qué esos valores fueron los valores ingresados, así como lo muestra la figura 61.

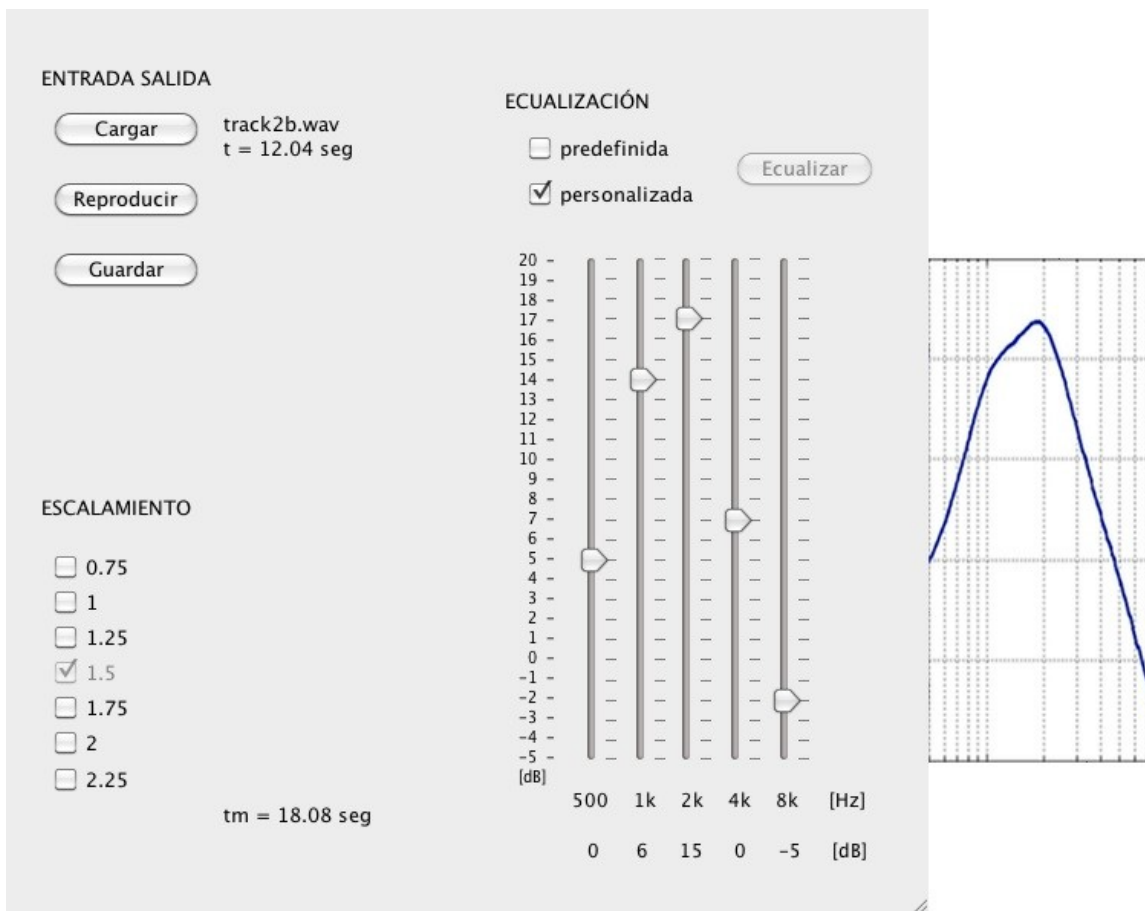


Figura 61. Respuesta frecuencial de la ecualización en cuestión

Si al ingresar ciertos valores de ganancias, obtenemos que el punto máximo de la respuesta en frecuencia rebasa los 20 dB, entonces los botones se regresarán a sus posiciones previas y la nueva ecualización no será calculada. Se tendrán que ingresar valores menores para que a la hora de calcular la nueva ecualización, se evite rebasar en cualquier banda de frecuencia, el punto máximo 20 dB.

Podemos dejar seleccionada la casilla con la última ecualización que se haya hecho y al mismo tiempo seleccionar otro escalamiento; la señal será escalada con el nuevo factor seleccionado y ecualizada con la ecualización que se haya dejado seleccionada previamente. Si se desea eliminar la ecualización, simplemente presionamos la casilla de nuevo y la señal de audio perderá su ecualización, independientemente de que la señal esté escalada o no lo esté. También la casilla se deseleccionará y la respuesta frecuencial del ecualizador regresará a cero en todas las bandas de frecuencia.

La ecualización Predefinida se implementó con cuatro filtros de media de octava y uno de octava, y con los siguientes valores de ganancias: 10, 2, 16, 2, 10. De esta manera, se obtiene la respuesta en frecuencia como las mostradas en las figuras 46 y 57. Sin embargo, si se utilizan éstos valores en la ecualización Personalizada, no se obtendrá la misma respuesta frecuencial. Esto se debe a que en la ecualización Personalizada todos los cinco filtros que se utilizan son de media de octava. Entonces, para obtener una respuesta frecuencial similar a la mostrada en la figura 51 al utilizar la ecualización Personalizada, debemos de ingresar los siguientes valores: -4, 17, 1, 16, -2, los cuales fueron obtenidos al probar diferente combinaciones. La figura 62 muestra estos valores y la respuesta frecuencial obtenida, la cual es similar a la mostrada en la figura 57.

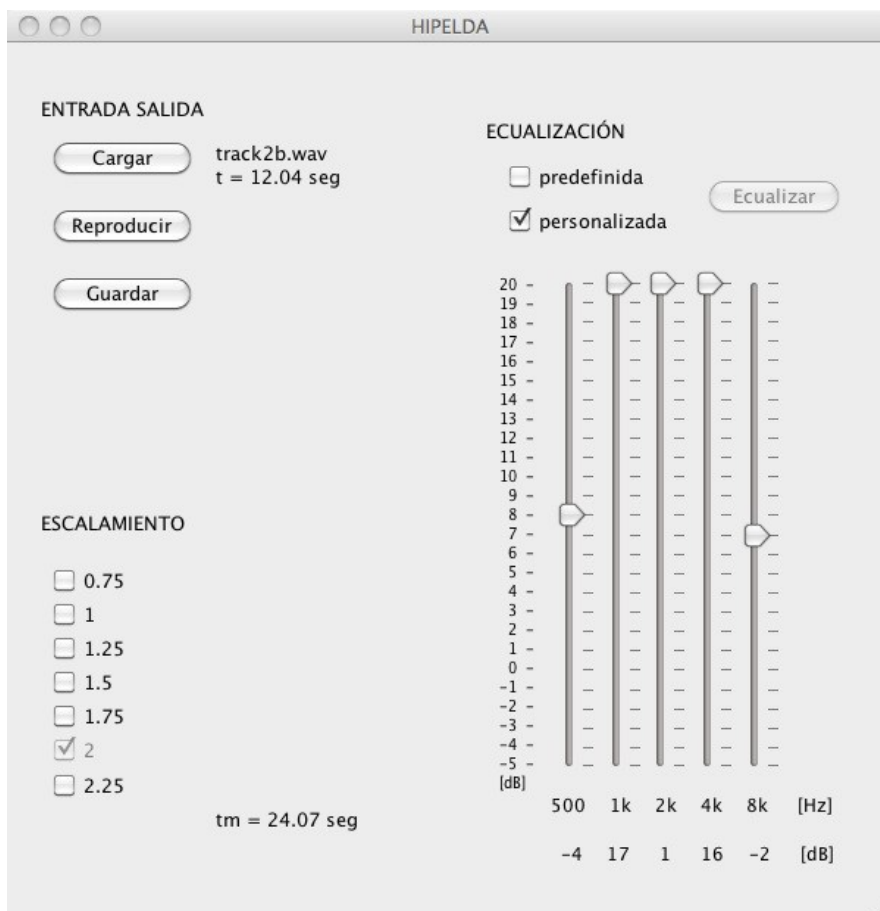


Figura 62. Respuesta frecuencial similar a la mostrada en la Figura 57

Podemos ir probando cómo se escucha la señal una vez que haya sido escalada y/o ecualizada, simplemente presionamos el botón Reproducir. También tenemos la opción de guardar los datos de la señal en un archivo de formato WAV, el botón Guardar se encarga de realizar esa función; al presionar dicho botón, se abre un selector para posicionarse en donde se desee guardar el nuevo archivo. Es necesario incluir la extensión **.wav** al ingresar el nombre. La figura 63 muestra un ejemplo para guardar un nuevo archivo.

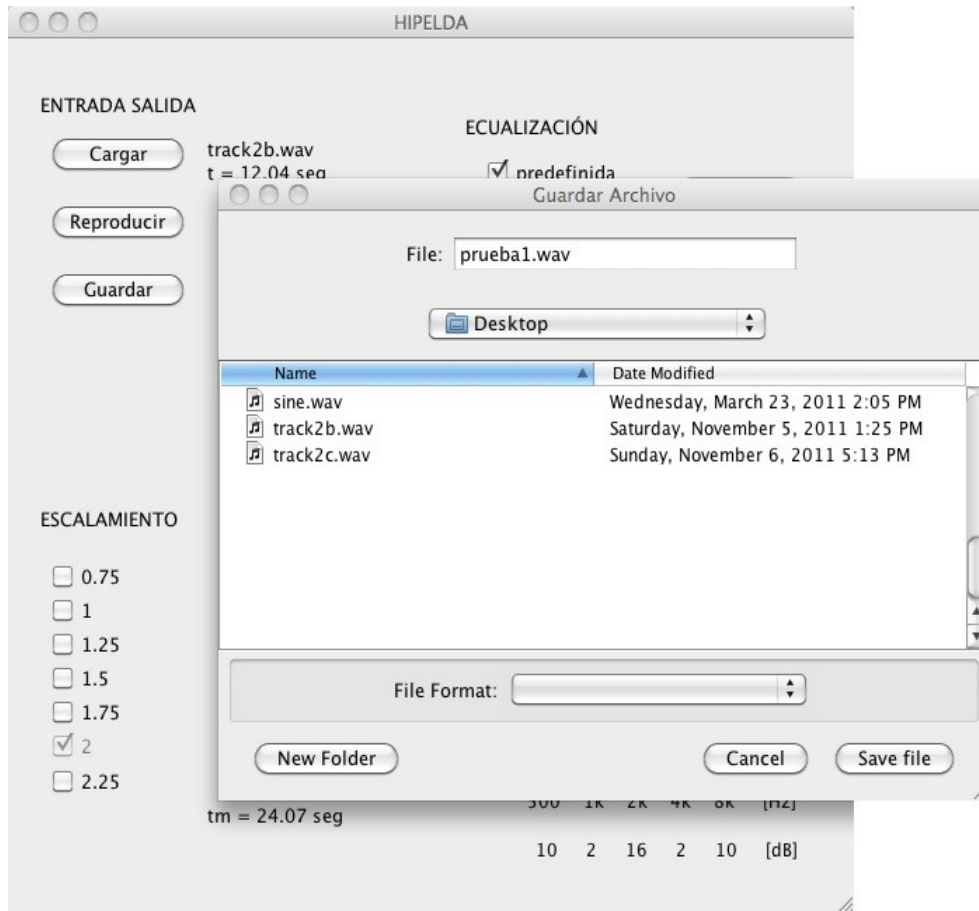


Figura 63. Guarda de los datos en un archivo de formato WAV

En el ejemplo de la figura 63, nos posicionamos en la carpeta **Desktop** y dentro de la casilla *File* escribimos el nombre **prueba1.wav** como nombre del archivo, el cual incluye la extensión .wav. Finalmente, el nombre del nuevo archivo recién creado y su tiempo de duración aparecerán al lado del botón Guardar. Ahora, esta nueva pista de audio la podemos reproducir en cualquier otro reproductor.

Conclusiones

La modificación de señales sonoras del habla que se implementa en el presente trabajo pretende ayudar a niños que padecen de un déficit de audición desde edad temprana. Dado que el problema se puede encontrar en cualquier parte de la vía auditiva, entonces no necesariamente la herramienta funcionará para todos los casos. Al niño se le tendrá que realizar una variedad de pruebas para detectar en donde radica su problema y así decidir si esta herramienta le podrá ser de su ayuda. Se ha estipulado en la literatura especializada que los niños mejoran en su desempeño para adquirir el habla si se entrenan con un programa de este tipo, por lo que vale la pena implementar un programa que incorpore las herramientas que modifican el habla en nuestro país (en español). El siguiente paso sería probar estas herramientas, incluso en gente mayor que también sufre de algún déficit de audición.

Como la meta principal es modificar el habla con el fin de que sea más entendible para personas con déficit de audición, se tuvieron que abordar los temas de fonología y fonética para entender con detalle el porqué, dónde y cómo las señales sonoras del habla se modifican. El procedimiento para modificar las señales, descrito en el capítulo 1 (Introducción), está basado en los trabajos hechos por Talall y colaboradores [6]. En estos trabajos se han obtenido resultados favorables al incluir este tipo de modificación del habla, en el programa de adiestramiento que implementan.

Respecto al procesamiento de señales, la base principal para implementar el Escalamiento en Tiempo es implementar el Codificador de Voz en Fase. Por esta razón en la sección 6.1 (Codificador de Voz en Fase) se analizan paso a paso las primeras cinco iteraciones con el fin de ver cómo la señal inicial se va analizando y cómo la señal final se va sintetizando. De esta manera se tiene una visión clara a la hora de implementar el escalamiento. Además de implementar el porcentaje de escalamiento de 50% a 20% para que la señal de audio se escuche más lentamente, como el reportado en la literatura [6], también se incluye un factor de 25% para que la misma se escuche más rápidamente y hasta un 225% para que se escuche más lentamente.

Como ejemplo ilustrativo, esta versión de la aplicación HIPELDA se tarda en escalar una señal de 12 seg grabada a dos canales, con un factor de 2, aproximadamente 10 segundos. Hay que tomar en cuenta que este tiempo variará dependiendo de la potencia de la computadora y en qué plataforma se este ejecutando la aplicación.

En el caso de la Ecuación, se agregó una Ecuación Personalizada porque quizá el usuario desee experimentar con otros valores y no necesariamente con los descritos en la literatura [6]. Para mostrar la respuesta en frecuencia de la ecuación, con el fin de simplificar la interfaz gráfica, se utilizaron los mismos botones deslizables.

Para implementar tanto el Escalamiento en Tiempo como la Ecuación, se desarrollaron varias herramientas con el lenguaje Java. Entre las principales se encuentran: la Transformada Rápida de Fourier (**fastFT.java**), el envolvimiento de la fase (**argument.java**), el diseño de filtros tipo Pico (**peaking.java**) y la ecuación de segundo grado para realizar el filtrado Biquad (**biquadF.java**). Otras herramientas, en este caso para el manejo de archivos WAV, son: la adquisición de datos desde un archivo (**storage.java**), la reproducción de datos de audio (**playing.java**) y la creación de archivos WAV (**saving.java**). Estas herramientas pueden ser útiles en otras aplicaciones.

Recomendaciones Finales para Trabajar a Futuro

Para que la aplicación sea más eficiente, en el sentido de procesar las señales más rápidamente, se tendrían que hacer algunos cambios en la programación. En este caso el modulo que escala la señal en el tiempo es el más tardado, comparado con el modulo ecuación el cual es inmediato. Una posible solución es reescribir los ciclos iterativos en ciclos recursivos y tomar el tiempo en que se tardan al ejecutarse los dos tipos de ciclos. Si esto acelera el proceso de ejecución, entonces eso sería una solución viable. Otra recomendación es averiguar el porqué la parte del código en Java que reproduce la señal, únicamente lo hace para pistas grabadas a dos canales. Si la aplicación pudiera reproducir y guardar pistas grabadas a un canal, entonces se podría trabajar con pistas de aún mayor duración. La ventaja que esto implicaría es que no se tendrían que seccionar las pistas a tramos de 20 o 25 segundos o incluso al tamaño de una sola palabra. Si se logra reducir el tiempo de ejecución entonces también se podría implementar el procesamiento de escalamiento en tiempo real. La ventaja que esto tendría es que al mismo tiempo que se reproduce la señal también se escala.

Valdría la pena averiguar si este tipo de adiestramiento es utilizado en hospitales o clínicas que traten con problemas de audición, en el país. Si es así, se podrían analizar los resultados que hayan obtenido para evaluar la efectividad del método. Si este método aún no se utiliza, entonces sería una gran oportunidad para realizar pruebas y así se podría adquirir información que ayude a evaluar el método y mejorarlo. La finalidad es ayudar a personas que sufren de cierto déficit de audición.

Bibliografía

- [1] Perception of Auditory Signals; Annals of the N.Y. Academic of Sciences Vol. 1224, pg. 96-108; H. Recanzone.
- [2] Temporal Processing Deficits of Language-Learning Impaired Children Ameliorated by Training; Science Vol. 271, pg. 77-80; M. Merzenich, W. Jenkins, P. Johnston, C. Schreiner, S. Miller, P. Tallal.
- [3] Language Learning Impairment: Integrating Research and Remediation; Scandinavian Journal of Pathology Vol. 39, pg. 197-199; P. Tallal, M. Merzenich, S. Miller, W. Jenkins.
- [4] Language Comprehension in Language-Learning Impaired Children Improved with Acoustically Modified Speech; Science Vol. 271, pg. 81-84; P. Tallal, S. Miller, G. Bedi, G. Byman, X. Wang, S. Nagarajan, C. Schreiner, W. Jenkins, M. Merzenich.
- [5] Developmental Aphasia: The Perception of Brief Vowels and Extended Stop Consonants; Neuropsychology Vol. 13, pg. 69-74; P. Tallal, M. Piercy.
- [6] Speech Modifications Algorithms Used for Training Language Learning-Impaired Children; IEEE Trans. Rehab. Eng. Vol. 6; pg. 257-268; S. Nagarajan, X. Wang, M. Merzenich, C. Schreiner, P. Johnston, W. Jenkins, S. Miller, P. Tallal.
- [7] Aprendizaje y memoria; Fundamentos de Psicología Fisiológica; Neil R. Carlson
- [8] La corteza Cerebral; Biología; Curtis
- [9] Plasticidad Neuronal; La Mente y el Cerebro. Visión orgánica, funcional y metafísica; Amadeo Muntané Sánchez
- [10] Déficit en el Procesamiento Temporal; Habilidades Psicolingüísticas en Niños con Trastorno Específico del Lenguaje de Kinder y Nivel Básico 1; Virginia Varela
- [11] Déficits auditivos; La educación especial en la escuela integrada; Ignasi Puigdelívol
- [12] The Hearing Pathways; Hearing Loss; Patrick J. Antonelli
- [13] Déficit auditivo; Retraso en el Habla de Origen Audígena; J. Fontané-Ventura
- [14] Grado y Gravedad de la Pérdida Auditiva; Déficit auditivo; Retraso en el Habla de Origen Audígena; J. Fontané-Ventura
- [15] Impact of Hearing Loss; Hearing Loss; Patrick J. Antonelli
- [16] Fonología; Introducción a la Fonética y Fonología del Español; Xavier Frías Conde
- [17] Experimental Phonetics; Phonetics; Britannica Encyclopedia
- [18] Los Fonemas del Español; Manual de Introducción al Estudio Fonético y Fonológico; Domingo Roman Montes de Oca
- [19] Vocales; Las Características Acústicas de los Elementos Segmentales; Joaquim Llisterri
- [20] Los Segmentos Consonánticos; Apuntes de Fonética; Alexander Iribar
- [21] Consonantes Oclusivas Orales; Las Características Acústicas de los Elementos Segmentales Joaquim Llisterri
- [22] Sistema Consonántico; Álbum de Espectrogramas de los Principales Alófonos del Castellano de Chile; Mauricio Figueroa Candia
- [23] Time-frequency Processing. Introduction; DAFX; Udo Zölzer
- [24] Phase Vocoder Basics; DAFX; Udo Zölzer
- [25] Phase Vocoder Basics. Block-by-Block Analysis/Synthesis Model; DAFX; Udo Zölzer
- [26] Phase Vocoder Basics. Filter Bank Summation Model; DAFX; Udo Zölzer
- [27] Phase Vocoder Implementations. Direct FFT/IFFT Approach; DAFX; Udo Zölzer
- [28] Phase Vocoder Effects. Time Stretching; DAFX; Udo Zölzer
- [29] Phase Unwrapping; Introduction to Digital Signal Processing. Computer Musical Speaking; Tae Hong Park
- [30] Phase Response; Digital Signal Processing. A Practical Guide for Engineers and Scientists; Steven W. Smith

- [31] Phase Vocoder Implementations. Phase Unwrapping and Instantaneous Frequency; DAFX; Udo Zölzer
- [32] Phase Vocoder Effects. Time Stretching. Block-by-Block Approach (FFT/IFFT); DAFX; Udo Zölzer
- [33] Filters. Equalizers. Introduction and Musical Applications; DAFX; Udo Zölzer
- [34] Implementation Structures for Recursive Digital Filters. The Four Direct Forms; Introduction to Digital Filters with Audio Applications; Julius O. Smith III
- [35] Time Domain Digital Filter Representation. Filter Stability; Introduction to Digital Filters with Audio Applications; Julius O. Smith III
- [36] Equalizers. Shelving Filters; DAFX; Udo Zölzer
- [37] Equalizers. Peak Filters; DAFX; Udo Zölzer

Códigos Java y MATLAB:

- [a] http://ul.to/l4nzz8c5/separados_m.zip
- [b] http://ul.to/xnOmd9xk/hipelda_m.zip
- [c] http://ul.to/10jqc20y/hipelda_j.zip
- [d] <http://ul.to/hrffxpyi/InterfazHIPELDA.zip>
- [e] http://ul.to/h90hp9cg/estuche_HIPELDA_w.zip
- [f] http://ul.to/g9nnvtb9/estuche_HIPELDA_l.zip
- [g] http://ul.to/cccrqfn5/estuche_HIPELDA_m.zip

fileChooser.java

<http://www.java2s.com/Code/Java/Swing-JFC/JFileChooserisastandarddialogforselectingfilefromthefilesystem.htm>

storage.java

<http://www.jsresources.org/examples/AudioDataBuffer.java.html>
http://www.jsresources.org/faq_audio.html#short_to_byte

shift.java

<http://www.dsprelated.com/showmessage/20790/1.php>

fastFT.java

http://www.cs.umbc.edu/~squire/cs455_l18.html

peaking.java

<http://www.mathworks.com/matlabcentral/fileexchange/16567-peaking-notch-iir-filter/content/peaking.m>

biquadF.java

<http://www.musicdsp.org/archive.php?classid=3#174>

playing.java

<http://www.jsresources.org/examples/AudioPlayer.java.html>

fileSaver.java

<http://www.java2s.com/Code/Java/Swing-JFC/SelectadirectorywithaJFileChooser.htm>

saving.java

<http://introc.cs.princeton.edu/java/stdlib/StdAudio.java.html>