UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Diseño e Implementación en Tiempo Real de un Reconocedor de Palabras Aisladas

T E S I S

Que para obtener el título de

Ingeniero Eléctrico Electrónico

P R E S E N T A

Cristian Hernández Escobedo

DIRECTOR DE TESIS

Dr. Rogelio Alcántara Silva

Ciudad Universitaria, Cd. Mx., Octubre de 2016

Agradecimientos

Ahora, que ya he terminado mi trabajo de tesis, me he sentado y me he preguntado, ¿Cómo he logrado llegar hasta este punto? Sin duda, muchos han sido los factores que me han permitido lograrlo. Sin embargo, siempre ha estado presente uno en particular, el apoyo indiscutible de mi familia y amigos.

A pesar de que somos seres de libre albedrío y podemos tomar decisiones por nuestra cuenta, siempre existirá una influencia externa que modificarán nuestras elecciones o que las reafirmarán. Como se expresa en el libro 'La puerta en el muro' de H.G. Wells, siempre podrá haber una puerta cuyo interior nos tiente a cruzarla, pero también, siempre habrán influencias que nos ayuden a decidir si entramos o nos quedamos fuera y seguimos caminado.

Para mí, la influencia más grande que me permitió lograr esta meta ha sido mi familia. Por ello es que aprovecho este segmento para dedicarles unas palabras de profundo agradecimiento. "Muchas gracias a mis padres por haberme educado y apoyado en mis estudios. Han logrado que me convierta en una gran persona. Y a mis hermanos por haberme acompañado y permitirme compartir grandes recuerdos con ustedes. Todos han sido la motivación principal por la que yo he logrado esta gran hazaña. En verdad, MUCHAS GRACIAS"

También, he de agradecer a mis amigos. Tanto viejos como nuevos. Han sido y siempre serán una parte importante de mi desarrollo, tanto como persona, como académico. Asimismo, me gustaría dar las gracias a mis profesores. Todos ustedes me han ayudado a desarrollarme como profesionista, facilitándome todo el conocimiento que ahora poseo y aprecio. Por ello, gracias.

Finalmente, quisiera dar un agradecimiento especial a esa pequeña y agradable influencia, que hace unos años me ayudo a seguir este camino. Y aun cuando no puedas leer esto, te estaré agradecido por siempre.

Jurado asignado

Presidente: DR. FRANCISCO JAVIER GARCÍA UGALDE

Vocal: DR. ROGELIO ALCANTARA SILVA

Secretario: ING. JAIME ERIK CASTAÑEDA DE ISLA PUGA

Suplente 1: DR. CARLOS RIVERA RIVERA

Suplente 2: M.I. HOOVER MUJICA ORTEGA

Universidad Nacional Autónoma de México, Facultad de Ingeniería, Ciudad Universitaria. Ciudad de México, México.

Resumen

Los sistemas de interacción hombre-máquina que podrían ser implementados en el futuro se basarían en el reconocimiento de voz. Esto debido a que es un método que permite intercambiar información fácil, rápida y cómoda para los usuarios. En el presente documento se realiza la propuesta de un reconocedor de palabras aisladas en tiempo real, basado en la comparación de los coeficientes de predicción lineal, con ayuda de la distancia Euclidiana.

En el presente, se puede observar la teoría relacionada con cada uno de los temas que se usan de base para resolver el problema de reconocimiento de palabras aisladas en tiempo real. En los primeros tres capítulos se plantean distintos métodos para procesar señales, tanto determinísticas y como aleatorias. Dentro de los métodos descritos se presenta la obtención de los coeficientes de predicción lineal.

Con ayuda de estos conceptos se plantea un sistema de síntesis de palabras. Este hace uso del método de auto-correlación para extraer el pitch y los coeficientes de predicción lineal de una señal, la cual contendrá una palabra cualquiera. Esta información se emplea para generar una señal de voz similar usando como base una señal aleatoria o un tren de impulsos, según corresponda.

El diseño del reconociendo de palabras aisladas se basa en los conceptos del sintetizador. Se compraran los coeficientes de predicción lineal entre dos señales, haciendo uso de la distancia Euclidiana, para determinar si estas son semejantes. Con este método se genera una base de datos, la cual será usada como referencia para identificar señales con el mismo comportamiento. Asimismo, se describe el algoritmo empleado para realizar esta tarea en tiempo real. Aquí se propone un método por medio del cual se discriminan las señales a analizar de la señal adquirida en tiempo real, que podría considerarse infinita.

Objetivos

- Se diseñará e implementará un sintetizador de voz por el método de auto-correlación con recorte central.
- Se diseñará e implementará un reconocedor de palabras aisladas en tiempo real con base en los parámetros LPC.

Índice general

Capítulo 1

Introducción

En la actualidad el crecimiento tecnológico ha permitido el desarrollo de nuevos instrumentos que han mejorado la calidad de vida. Sin embargo, la interacción hombre-máquina sigue siendo una limitante, en algunos casos. Una interacción óptima entre las tecnologías y las personas sería aquella que se llevara a cabo de forma natural, como lo es el habla. Por medio del habla se podrían intercambiar paquetes de información rápidamente y en tiempos cortos. Además, este método permitiría eliminar la limitación de que el individuo requiera la capacidad de interactuar físicamente con un teclado, un botón o similares.

El habla es uno de los métodos más robustos para la transmisión de la información. Esta habilidad fue adquirida en las primeras etapas del desarrollo de la humanidad y evolucionó de la misma forma que lo hizo la civilización. Durante este largo periodo se han visto una gran cantidad de cambios. Estos han dado como resultado una amplia gama de idiomas y dialectos que se distribuyen alrededor del mundo.

El reconocimiento de voz es un campo muy extenso y a lo largo de su historia se ha abordado de muchas formas. En este texto se expone un método para identificar un número específico de palabras basado en la comparación de los parámetros de predicción lineal (LPC por sus siglas en inglés *Linear Predictive Coefficient*) extraídos de sus señales de voz.

Para poder implementar este reconocedor se requiere analizar algunas de las características de las señales de voz. La potencia y energía son dos características de las señales que nos permiten saber que intensidad poseen la señal; lo cual corresponde al volumen en las señales de audio. La representación frecuencial es otra característica importante de las señales. Con ella se puede observar la distribución que tiene esta en el espectro. A partir de esta información se podría identificar las señales de voz sonora y sus frecuencias que la componen.

Es común que para analizar señales se emplean métodos como la segmentación, el

ventaneo y la aplicación de filtros. Estos ayudan a adecuar las señales para poder ser analizadas. Una señal de voz es una combinación arbitraria de sonidos aleatorios y determinísticos. Por esta razón, se deben de realizar algunas consideraciones al momento de manipularlas. Las señales aleatorias son analizadas a partir de métodos probabilísticos al tratar de predecir su comportamiento. Mientras que las determinísticas poseen un comportamiento estable y predecible.

En el algoritmo propuesto se emplean una comparación de los parámetros LPC de dos señales para determinar si son similares. Estos valores representan el comportamiento codificado de la señal, y pueden ser usados para generarla de nuevo. Los valores pueden ser empleados como base para generar un sintetizador de palabras al usarlos como coeficientes de filtros de ecuaciones en diferencia.

El algoritmo de reconocimiento usa los coeficientes LPC como referencia para determinar si la señal de entrada es válida. Los parámetros obtenidos de las señales de entrada son comparados con los contenidos en la base de datos, por medio de la distancia Euclidiana, para determinar qué tan diferentes son. Para este algoritmo se tiene en consideración que la adquisición de la señal es en tiempo real, por lo que el sistema debe de permanecer pendiente para detectar cuando una señal de voz se presente. Para esto se propone un detector de inicio y fin de la palabra, el cual se encarga de buscar aquellas partes de la señal en tiempo real que poseen una energía y duración mínimas. Sin embargo, dado que hay que realizar el debido análisis de las señales de entrada, habrá un retraso entre la adquisición de la señal voz y la identificación de la misma. Dicho retraso puede ser acortado al optimizar los algoritmos de procesamiento y análisis.

Finalmente, el algoritmo es implementado en el lenguaje de programación Java. Los distintos métodos que se ven implementados en este programa fueron probados en el sistema de desarrollo de Matlab con datos simulados y, en la medida de lo posible, reales.

1.1. Contexto histórico

La historia del reconocimiento de voz se remonta a los años 60's, cuando en los laboratorios Bell se desarrolló el primer sistema que trataba de resolver este problema. Sin embargo, la historia relacionada con la síntesis de voz es mucho más longeva, remontándose al año de 1779 cuando el filósofo alemán C.G. Kratzenstein diseñó un aparato que simulaba los sonidos de las vocales 'A', 'E', 'O' y 'U'. [11]

A lo largo de la historia en ambos métodos se observan distintas propuestas. A pesar de que estos sistemas se desarrollaron para abordar temas distintos, el reconocimiento de voz podría ser resuelto con ayuda de los métodos usados para la síntesis de voz.

1.1.1. Historia de la síntesis de voz

El origen de la síntesis de voz se remonta al establecimiento de los diferentes fonemas. Esto tiene su origen en el año de 1668 cuando B. J. Wilkins publicó en su alfabeto fonético, el cual incluía 8 vocales y 26 consonantes.[11] Cada uno representaría los distintos sonidos que podría emitir el humano.

Uno de los primeros sistemas para realizar el proceso de síntesis de voz fue desarrollada por el fisiólogo alemán C. G. Kratzenstein en 1779. Él desarrolló un instrumento compuesto por un conjunto de cinco tubos, con los cuales se podían simular los sonidos de las vocales (A, E, I, O y U)[11]. Algunos de los aparatos que fueron desarrollados posteriormente se basaban en el mismo principio de hacer circular aire a través de una cavidad que tratase de simular algún fonema.

En el año de 1846, J. Feber construyó una maquina a la que nombraría *Euphonia*. Esta hacía uso del tono fundamental de los sonidos sonoros para generar la voz [11]. El emplear este concepto le permitió el generar vocablos más reales. De esta forma, para los años 70's Alexander Graham Bell lograría llevar a cabo el desarrollo del teléfono, el cual extraía parámetros representativos de una señal de voz para su posterior síntesis.

"C. Paget en 1923 descubrió que había dos componentes frecuenciales en todos los sonidos vocálicos e hizo una tabla con ellas, por observación de su propia voz" [11]. Un año antes, J.Q. Stewart desarrolló el primer sintetizador completamente eléctrico, implementando un circuito que simularía las cuerdas vocales y su correspondiente resonancia.

Uno de los primeros instrumentos que se desarrollaron con base en componentes eléctricos fue el *Voder*. Este poseía dos fuentes de sonido, un zumbido generado por medio de la un oscilador y un ruido aleatorio [11]. Ambas eran empleadas para para producir los sonidos sonoros y no sonoros respectivamente.

Los sistemas de síntesis de voz pueden ser aplicados en varios campos, pero uno de los primeros en los que fue usado fue la telefonía. Aquí se buscaba el poder descomponer la señal de voz en una serie de parámetros de fácil transmisión. De entre los primeros sistemas aplicados en esta área se encuentra el *Vocaderel*, proyectado en 1939 por los laboratorios Siemens de Munich [11]. Este descompone la señal de voz en un conjunto de parámetros codificados cuya transmisión es fácil y de bajo costo. Al llegar la información al receptor, la señal de voz se reconstruye haciendo uso de dichos valores.

En los años 50's del siglo XX, un organista ciego de Guadalajara, México, llamado Ernesto Hill Olvera desarrolló una técnica para simular palabras empleando un órgano *Hammond*. Por medio de la manipulación de distintas palancas, que poseía este instrumento, logró imitar algunas canciones populares de la época.

Con la llegada de las computadoras, la implementación de los métodos para el reconocimiento de voz se ha visto beneficiada. Esto debido a que se pueden procesar

y analizar las señales de voz con gran facilidad. Así como aplicar algoritmos cada vez más complejos para obtener resultados más fidedignos.

1.1.2. Historia de los métodos para reconocimiento de voz

La historia del reconocimiento de voz se remonta al año de 1952 cuando en los laboratorios Bell, Biddulph y Balashek construyeron un sistema de reconocimiento de dígitos aislados para un único usuario [11]. La base de este sistema recaía en la comparación de los espectros de cada una de las señales. Un trabajo similar se presentó en 1956 en los laboratorios RCA, donde Olson y Belar trabajaron sobre un sistema que podía reconocer diez diferentes silabas [11], el cual también se restringía a un único usuario.

En el año de 1959 en el colegio Universitario de Inglaterra, Fry y Denes intentaron construir un reconocedor de fonemas que incluiría cuatro vocales y nueve consonantes [11]. Este se basaba en la comparación del espectro y patrones de la señal usando operaciones estadísticas. En el mismo año, en los laboratorios Lincoln MIT se construyó un sistema capaz de reconocer 10 vocales embebidas en el formato: /b/ - vocal - /t/ [11]. Siendo este un sistema independiente del locutor hacía uso de un banco de filtros para estimar el espectro y las variaciones en el tiempo.

En 1962 Sakai y Soshita, de la Universidad de Kyoto, Japón, diseñaron un hardware para el reconocimiento de fonemas. Este se caracterizaba por la segmentación de la señal y su análisis por el método de cruces por cero. Un proyecto igual de importante se realizó en los laboratorios RCA durante los años 60's en el que se realizaba la detección del inicio y el final de la señal para realizar el reconocimiento. Asimismo, en la Unión Soviética, Vintsyuk propuso el método de programación de Alineamiento Temporal Dinámico de un par de señales. [11]

En los años 70's las investigaciones referentes al reconocimiento de voz vieron su auge. En Rusia Velichko y Zagoruyko realizaron estudios que ayudarían al reconocimiento de patrones. Mientras tanto, en Japón, Sako y Chiba trabajaron sobre los métodos de programación dinámica. Finalmente en Estados Unidos, Itakura realizó investigaciones sobre los LPC, permitiendo codificar voz con una baja taza de bits. [11]

En los años 80's las investigaciones se enfocaron en buscar métodos para concatenar varias palabras. Durante este periodo se desarrolló el modelo de Markov, que fue usado en los laboratorios de IBM y el *Institute for Defense Analyses* [11]. Asimismo en los años 80's se retomó la idea de aplicar redes neuronales para abordar el problema de reconocimiento de voz. Este último había sido descartado en los años 50's dado que se tuvieron problemas con la aplicación práctica.

1.2. El aparato fonador

La producción de los sonidos que constituyen el habla del ser humano está focalizada en el aparato fonador. En este, los sonidos son generados al modificar la vibración de las ondas en el aire cuando transitan a través de distintos módulos del mismo. Las principales componentes del aparato fonador son: los pulmones, la laringe, el paladar blando, y las cavidades nasal y oral. Además de complementarse con otras partes como lo son los dientes, la lengua, los labios, el alveolo, entre otros [13].

En el aparato fonador, los pulmones están a cargo de impulsar el aire sobre el cual se producirán las vibraciones. Este se desplaza a través de la tráquea hasta llegar a la laringe, la cual contiene las cuerdas vocales. Estas últimas vibran con el paso del aire generando distintos armónicos, según la tensión y longitud de las mismas. Al salir de la laringe, el aire tiene como destino la cavidad nasal u oral, según sea la posición del paladar blando. Estas cavidades funcionan como caja resonante, ayudando a intensificar la potencia de las vibraciones, así como caracterizar los sonidos.

Para manipular las vibraciones y generar los patrones de sonido que componen el habla, el aparato fonador emplea principalmente los labios, la lengua, los dientes, el paladar y el alveolo. Dependiendo de cada una de las combinaciones realizadas se generará un sonido determinado. Estos son limitados y en cada lenguaje se emplea una cantidad distinta para formarlos.

En la figura 1.1 se ejemplifican algunas de las partes que componen al aparato fonador, excluyendo a los pulmones. La mayoría de las secciones que articulan las señales del habla se encuentran contenidos en la boca.

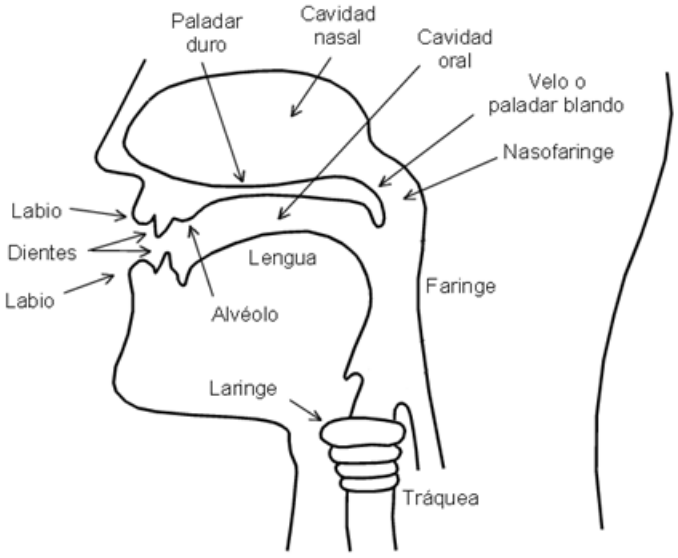


Figura 1.1: Esquema del aparato fonador, sin incluir los pulmones

1.2.1. Clasificación de la voz según su origen

Los sonidos generados por una persona pueden clasificarse de distintas formas. La discriminación suele estar basada en los distintos factores que involucran la generación de la voz. Por ejemplo, algunos sonidos requieren de las cuerdas vocales para ser producidos, o de una determinada combinación de los componentes del aparato fonador.

Los distintos sonidos producidos por el aparato fonador se basan en la forma en que el mismo manipula las vibraciones del aire. En primer lugar se hace pasar una corriente, impulsada por los pulmones, por en medio de la laringe, en donde se encuentran las cuerdas vocales. La primera clasificación se basa en sí estas cuerdas las hacen vibrar o no (sonidos sonoros o no sonoros). Posteriormente el aire llega a las cavidades nasal y vocal, en donde los distintos componentes, y sus combinaciones manipulan, la corriente.

Sonidos sonoros y no sonoros

Los "sonidos sonoros"son todos aquellos que se generan tras haber hecho vibrar las cuerdas vocales. Se caracterizan por tener una representación en el tiempo periódica, mientras que en la frecuencia se puede observar como una serie de impulsos le componen. Por lo general, estas señales contienen una frecuencia cuya relevancia es mayor ante las demás, a la cual se le conoce como frecuencia fundamental o pitch, y que suele caracterizar a ese sonido.

En la figura 1.2 podemos observar un ejemplo de una señal de voz sonora. En el tiempo es posible apreciar la periodicidad que posee, aun cuando cada ciclo parece diferente al resto. De la misma forma, podemos observar que en frecuencia la señal está compuesta por una serie de impulsos de magnitud diferente. En esta última parte es posible observar un impulso cuya magnitud sobrepasa a los demás, a este se le podría considerar como la frecuencia fundamental de esta señal de voz.

Los "sonidos no sonoros.º "sonidos sordos"son aquellos cuyo origen no involucra las cuerdas vocales. Estos tienen una representación en el tiempo no periódica que se asemeja a una señal de ruido. Asimismo, en frecuencia su representación está compuesta por un gran número de valores. Esto último nos indica que la señal no puede ser representada mediante una combinación específica de frecuencias y que su semejante más próximo es el ruido blanco Gaussiano, el cual contiene todas las frecuencias [6].

En la figura 1.3 se puede observar un ejemplo de una señal no sonora. Se observa que su representación en el tiempo asemeja a una señal de ruido, mientras que la representación en frecuencia parece contener todas las posibilidades del espectro disponible.

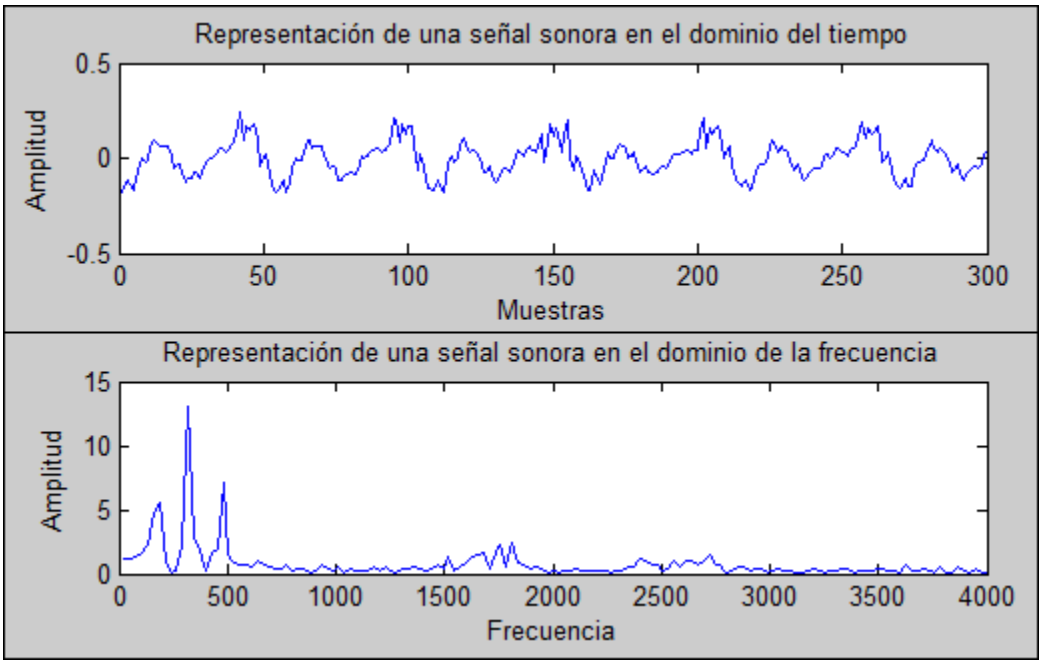


Figura 1.2: Representación en el tiempo y en frecuencia de una señal sonora

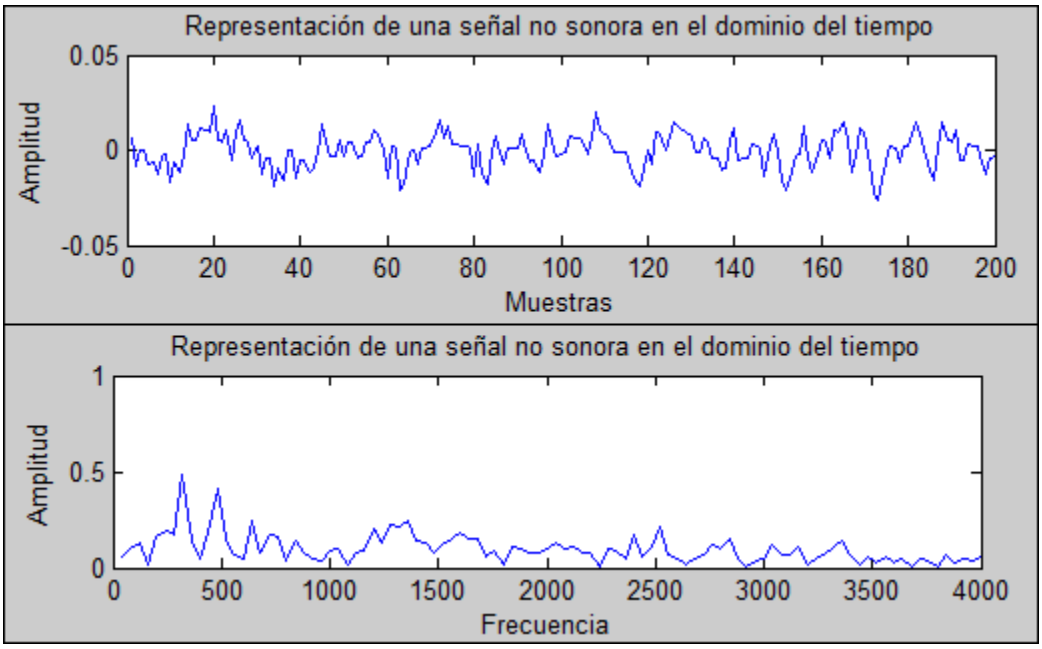


Figura 1.3: Representación en el tiempo y en frecuencia de una señal no sonora

Sonidos según su articulación

Se pueden clasificar los sonidos según la articulación de los mismos, es decir, la forma en que interactúan la lengua y los labios con los dientes, el alveolo, la glotis y el paladar. Las distintas combinaciones alteran las vibraciones del aire generando todos los fonemas

del lenguaje. La clasificación según las combinaciones se presenta a continuación [13].

- Bilabiales: oposición de ambos labios. /b/
- Labiodentales: oposición de los dientes superiores con el labio inferior. /f/
- Linguointerdental: la lengua entre los dientes. /z/
- Linguodentales: colocación de la punta de la lengua con la parte trasera de los dientes superiores. /d/
- Linguoalveolar o Alveolares: oposición de la punta de la lengua con la región alveolar. /s/
- Linguopalatal o Palatales: oposición de la lengua con el paladar duro. /ch/
- Linguovelar o Velares: oposición de la parte posterior de la lengua con el paladar blando o velo. /k/

Sonidos según el modo de articulación

La articulación de los sonidos implica el manipular las vibraciones del aire de distintas formas. Las dos ramas principales se denominan como "sonidos plosivos" y "Sonidos no plosivos" [13]. Los sonidos plosivos son aquellos en los que la corriente de aire se emite como si fuera una pequeña explosión, mientras que los no plosivos son aquellos cuya emisión es continua. A continuación se listan las clasificaciones derivadas de estas dos ramas.

- Plosivas
  - Oclusivos: la salida del aire se cierra por completo de forma momentánea, es decir, se produce una explosión. /p/
  - Fricativos: el aire sale atravesando un espacio estrecho, produciendo un roce. /f/
  - Africados: oclusión seguida por fricación. /ch/
- No Plosivas
  - Nasales: Parte del aire sale por la nariz. /m/
  - Laterales: la lengua obstruye el centro de la boca y el aire sale por los lados. /l/
  - Vibrantes: la lengua vibra cerrando el paso del aire intermitentemente. /r/

Fonemas

Un fonema se considera como la unidad mínima del lenguaje hablado. Cada uno de estos representa un sonido único que puede ser emitido por el aparato fonador y que en combinación forman los distintos lenguajes hablados alrededor del mundo. En la figura 1.4 se pueden observar todos los posibles fonemas existentes a nivel internacional. Sin embargo en el español solo se usa un total de 24, donde 5 corresponden a las vocales.

Figura 1.4: Clasifición de los fonemas (consonantes)

Las vocales, a diferencia de las consonantes, se clasifican en una tabla distinta a la mostrada en la figura 1.4. Estas se consideran por separado ya que para ser emitidas no hay ninguna obstrucción del flujo del aire a través del aparato fonador. En la figura 1.5 se pueden observar las variantes de las vocales que se tienen registradas a nivel internacional.

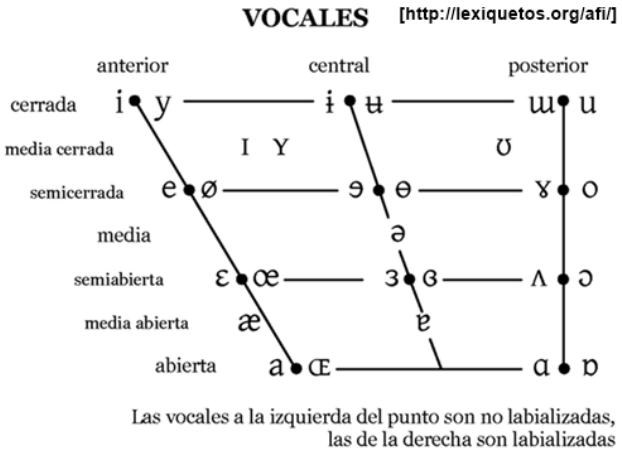


Figura 1.5: Clasifición de los fonemas (Vocales)

1.2.2. Frecuencia fundamental de la voz o pitch

El pitch es un valor exclusivo de los sonidos sonoros que representa la frecuencia base bajo la que están generados. Este valor define el tono de la voz de las personas. Debido a esto último es que incluso para sonidos iguales emitidos por la misma persona se pueden tener valores de pitch diferentes.

El valor del pitch está en función de la longitud y la tensión de las cuerdas vocales. Cuando las cuerdas vocales tienen una vibración alta, estas se tensan y producen sonidos más agudos. En el caso contrario, si las vibraciones son bajas la tensión es baja y se producen sonidos graves. Por lo general, las características de las cuerdas vocales varían con forme la persona crece, dificultando la tarea de poder establecer un valor único que caracteriza el pitch de una misma persona. Sin embargo, es posible establecer un rango que englobe las variaciones que realiza.

La variación del valor del pitch con mayor notoriedad es apreciada cuando se comparan los datos obtenidos de una población de hombres contra una de mujeres. "Unificando las cifras que ofrecen los diferentes autores, podemos acotar la frecuencia de la voz masculina entre 50 y 200 hercios y la femenina entre 150 y 350 hercios." [1]

En la figura 1.6 se presentan las señales correspondientes a las vocales, que son sonidos sonoros. Se puede observar que su representación en frecuencia está constituida principalmente por un conjunto de impulsos de diferentes magnitudes. Entre estos impulsos se puede observar que existe uno cuyo tamaño es mayor que el resto, este corresponde al pitch asociado a cada ejemplo.

1.3. Definición señales y sistemas

Las señales son propagaciones de información en el tiempo en forma de algún tipo de energía a través de uno o varios medios. Por ejemplo, la voz es un tipo de señal que se propaga con facilidad en el aire como vibraciones; la luz visible se puede propagar por distintos medios como parte de la gama de ondas electromagnéticas; asimismo, en electrónica, las señales empleadas para transmitir información se basan en una serie de impulsos eléctricos que se propagan por los distintos circuitos eléctricos.

Una señal cualquiera está compuesta por una serie consecutiva de valores que se presentan para cada instante de tiempo. Cada uno representa una magnitud correspondiente al tipo de energía que compone a la señal. Para cada instante de tiempo solo puede existir un correspondiente dato.

Cuando una señal tiene la capacidad de proporcionar un dato para cada instante de tiempo, es decir, puede tener un número infinito de valores, se dice que la "señal es continua". Si la señal solo proporciona información para determinados momentos, y

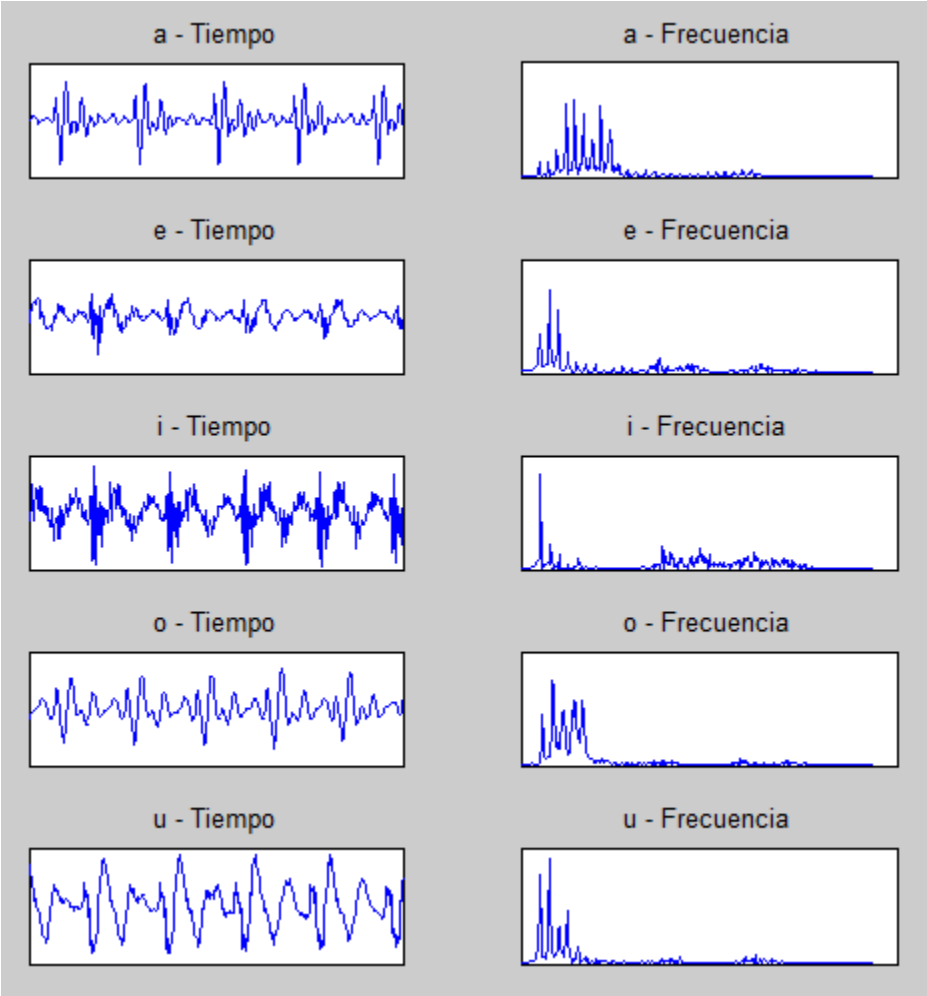


Figura 1.6: Representación en el tiempo y en frecuencia de las vocales

entre cada uno de ellos no es posible obtener datos, se dice que la "señal es discreta". Las señales discretas suelen ser extraídas de las señales continuas al recolectar información en determinados intervalos de tiempo. Como consecuencia se pierde mucha de la información de la señal continua, agilizando los procesos de análisis.

La información de una señal puede ser representada de forma analógica o digital. En la primera se pueden tener valores para cualquier instante de tiempo, mientras que la segunda solo proporcionará datos en determinados momentos. Las señales analógicas son similares a las señales continuas, pero estas pueden o no ser finitas. De la misma forma, las señales digitales son similares a las señales discretas, y también pueden ser finitas o infinitas.

Cuando un señal cualesquiera es modificada por un medio con el que tubo interacción, se dice que este se trata de un sistemas. Estos suelen recibir como entrada una o más señales para proporcionar una respuesta acorde a ellas. Los sistemas pueden ser tanto continuos como discretos.

1.4. Sistemas en tiempo real

Se considera que un sistema ocurre en tiempo real cuando la disposición y manipulación de la información se realiza inmediatamente después de que esta se ha generado. Estos sistemas se caracterizan por tener un conjunto de restricciones temporales que deben de respetarse. Los sistemas en tiempo real están compuestos por un conjunto de eventos embebidos distribuidos en varios hilos.

En computación un sistema en tiempo real puede ser clasificado de dos formas [14]

- Hard real-time system o sistemas en tiempo real duros: Estos procuran que todos los procesos siempre se culminen dentro de las restricciones temporales establecidas.
- Soft real-time system o sistemas en tiempo real blandos: En estos los procesos que se llevan a cabo no siempre se culminan dentro de los límites de tiempo establecidos. Se procura el dar prioridad a los eventos que se consideran críticos para que estos si cumplan con la restricción temporal.

Los sistemas en tiempo real duros se suelen emplear en procesos en los que se precisa que cada evento se culmine antes del tiempo establecido. Asimismo, se procura que los límites establecidos sean siempre los más bajos posibles. Por ejemplo, un radar requiere que la información capturada se analice con rapidez para poder visualizar los objetos con la mayor exactitud posible.

Los sistemas en tiempo real blandos se suelen emplear en aquellos procesos en los que no se requiere que todos los eventos terminen antes del límite establecido para que funcione el conjunto. Estos son empleados en sistemas multimedia, realidad aumentada, sistemas de adquisición de datos, entro otros.

Los sistemas en tiempo real están compuestos por un conjunto de eventos embebidos distribuidos en varios hilos, es decir, son concurrentes. Estos suelen necesitar la ejecución de dos o más procesos en el mismo periodo de tiempo. Por lo general cada uno se lleva a cabo de forma aislada, con la finalidad de evitar interferencias entre ellos. En muchos sistemas se suele implementar hardware dedicado para llevar a cabo estas actividades.

Una alternativa que se suelen emplear en los sistemas que no tienen la capacidad de ejecutar más de una actividad en el mismo periodo, son los 'Thread'. Estos son mecanismos por medio de los cuales dos programas simulan una ejecución en paralelo. La función 'Thread' segmenta los hilos, con base en sus líneas de código, e intercala las partes. Posteriormente los ejecuta uno a uno, almacenando el estado de cada programa, para determinar en qué punto ha terminado la sección anterior.

Capítulo 2

Análisis y procesamiento de señales determinísticas

Se consideran como señales determinísticas a todas aquellas que pueden representarse mediante relaciones matemáticas explicitas, tablas o listas. En este tipo de señales los valores futuros pueden ser obtenidos a partir de sus valores pasados, respetando siempre la relación de una única magnitud para cada instante de tiempo.

El análisis de este tipo de señales se basa en los distintos métodos algebraicos que puedan operar las expresiones matemáticas que las representan. Dependiendo del comportamiento que presenten las señales, será de mayor utilidad el emplear algún método en específico.

2.1. Señales periódicas y aperiódicas

Una señal determinística puede comportarse de dos formas, como una "señal periódica" o como una "señal aperiódica". La primera maneja a aquellas cuyo comportamiento en el tiempo es repetitivo. El lapso de tiempo (t) que transcurre entre las repeticiones se conoce como periodo (T), y se suele medir en segundos para las señales analógicas o en número de muestras (N) para las señales digitales. Todas las señales periódicas están definidas mediante la siguiente estructura

$$Tiempo\ continuo : x(t) = x(t+T) \ \ \forall \ \ t = 0...\infty$$
$$Tiempo\ discreto : x[n] = x[n+N] \ \ \forall \ \ n$$

Donde n son las muestras de la señal.

En las señales periódicas se pueden definir un número ilimitado de periodos con

distinta magnitud, sin embargo a aquel cuya duración es la más corta se le conoce como periodo fundamental (T0). Cuando una señal carece de periodo, para cualquier instante de tiempo, se le conoce como señal aperiódica o señal no periódica. Por ejemplo, los dos casos de señales periódicas más simples corresponden al seno y al coseno. Los cuales están definidos mediante las siguientes ecuaciones

$$Seno : x(t) = Asin(2ft + \alpha)$$
$$Coseno : x(t) = Acos(2ft + \alpha)$$

Donde A es la amplitud de la señal, t es la variable independiente del tiempo, α es la fase y f es la frecuencia, que se define como el inverso del periodo.

$$f = 1/T$$

En el tiempo, su representación se observa como una ondulación infinita, con tantas repeticiones por segundo como lo denote *f*. En el dominio de la frecuencia su representación se observa como un impulso localizado en la frecuencia indicada por *f*. En la imagen 2.1 se ejemplifica el comportamiento de una función seno en una frecuencia cualquiera.

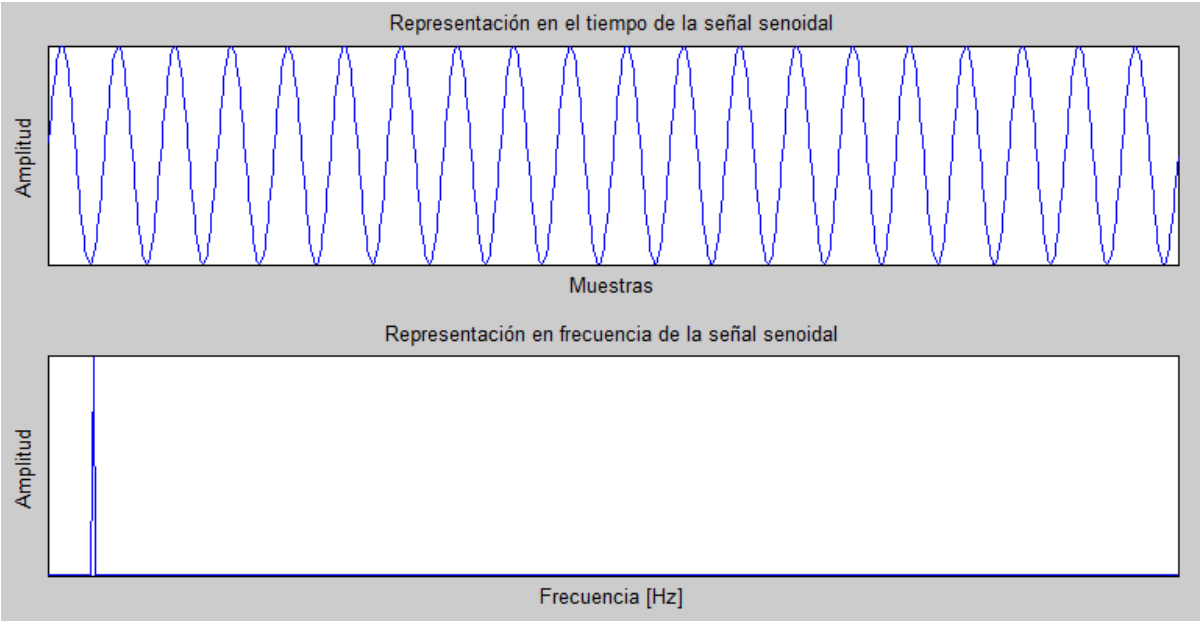


Figura 2.1: Representación en el tiempo y en frecuencia de una señal seno

Gracias al comportamiento en frecuencia de estas dos funciones, es posible el representar cualquier otra función periódica como una combinación de senos y cosenos con distintas frecuencias y amplitudes. Esto debido a que toda señal periódica está compuesta de un conjunto definido de impulsos en el dominio de la frecuencia. Por ejemplo,

una señal cuadrada representa en el dominio de la frecuencia una serie de impulsos separados la misma distancia con magnitud decreciente, como se presenta en la figura 2.2.

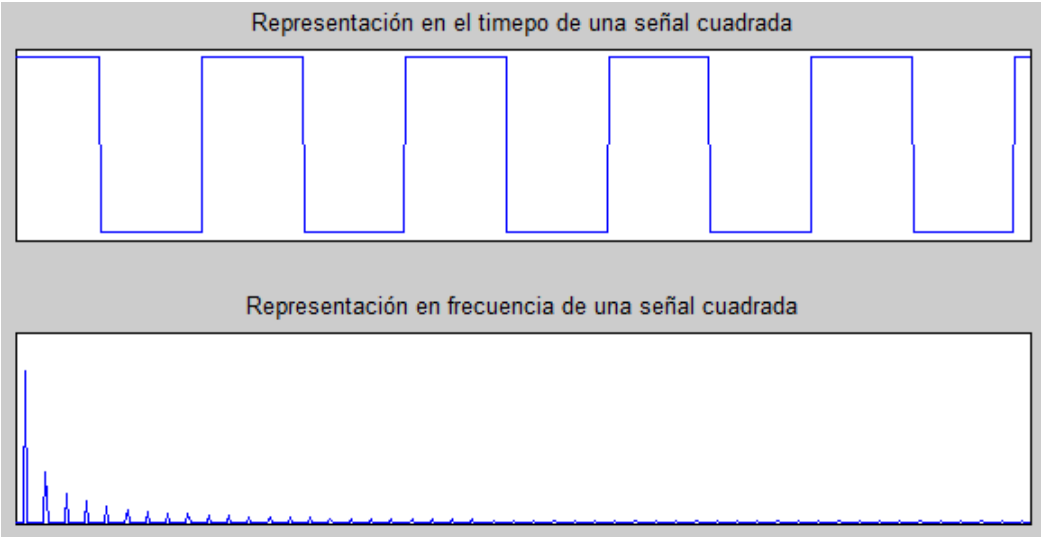


Figura 2.2: Representación en el tiempo y en frecuencia de una señal cuadrada

Una señal no periódica también puede ser representada como una combinación de senos y cosenos. Para esto se debe de considerar que dicha señal es finita en el tiempo, y que se repetirá el mismo patrón tras finiquitar dicho intervalo. Es decir, se fuerza a la señal no periódica a ser periódica.

2.2. Señal de energía y potencia

"Las señales pueden representar una amplia variedad de fenómenos. En muchas aplicaciones [. . .] las señales que examinamos están directamente relacionadas con cantidades físicas que capturan potencia y energía" [2].

Las señales se pueden caracterizar por tener una energía o potencia finita, según sea su comportamiento. Se dice que una señal de energía existe si para cualquier intervalo de tiempo, aun cuando este sea infinito, posee una magnitud de energía finita. Además, las señales energía siempre presentan una potencia promedio igual a cero. La energía de una señal es calculada empleando la ecuación 2.1 [2].

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt \tag{2.1}$$

Una señal de potencia es aquella cuya potencia promedio es finita, mientras que su energía es infinita. "[...] ya que si hay una energía promedio diferente de cero por

unidad de tiempo [...], entonces integrando o sumando ésta sobre un intervalo de tiempo infinito se obtiene una cantidad infinita de energía" [2]. La potencia de una señal está definida por la ecuación 2.2

$$P = \lim_{N\to\infty} \frac{1}{2T} \int_{T}^{T} |x(t)|^2 dt \tag{2.2}$$

La representación análoga de las ecuaciones de energía y potencia en tiempo discreta quedan representadas por las ecuaciones 2.3 y 2.4, respectivamente [2].

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2 \tag{2.3}$$

$$E = \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2 \tag{2.4}$$

"En la práctica siempre transmitimos señales que tienen energía finita. Sin embargo, para describir señales periódicas, que por definición existen para todo tiempo t y tienen energía infinita [se emplean las señales de potencia...] Como regla general, las señales periódicas y las señales aleatorias se clasifican como señales de potencia, mientras que las señales determinísticas no periódicas se las clasifica como señales de energía" [4]

2.3. Representación en frecuencia

La frecuencia se define como el número de repeticiones de un evento periódico en cada unidad de tiempo. En señales, este valor se asocia al número de veces que la misma forma de la señal se repite cada unidad de tiempo.

En el sistema internacional la unidad de la frecuencia es el hercio o hertz (Hz o segundo^{-1}) en honor al físico Alemán Heinrich Rudolf Hertz. Este valor equivale al número de repeticiones de un evento en un segundo. La frecuencia también se suele representar mediante la frecuencia angular ω cuyas unidades en el sistema internacional son los radianes/segundo. Estos representan el ángulo desplazado cada segundo en una rotación infinita, donde cada rotación equivale a una repetición en la señal. La relación existente entre la frecuencia y la frecuencia angular se denota de la siguiente forma

$$\omega = 2\pi f$$

De forma análoga, al inverso de la frecuencia se le conoce como periodo y se mide en segundos. Este valor representa el tiempo mínimo que se tarda en repetirse el mismo patrón de la señal.

$$T = \frac{1}{f}$$

El valor de la frecuencia, junto con la función seno o coseno, permite crear señales periódicas en forma de onda. Cada una de estas solo podrá representar una frecuencia. Sin embargo al realizar la adición de nuevos conjuntos con otras características se tendrá como resultado señales periódicas más complejas.

Sin embargo, pocas veces las señales periódicas que se presentan en la práctica tienen una estructura que se puede representar con una única función seno. Para estos casos se suele aplicar el análisis de Fourier. Este método permite descomponer las funciones complejas en una suma de senos y cosenos.

El análisis de Fourier trata de representar una función periódica compleja como una suma de funciones periódicas más sencillas. La base de este análisis recae en la función de Euler con la estructura mostrada en la ecuación 2.5

$$e^{j\alpha} = cos(\alpha) + j\ sin(\alpha) \tag{2.5}$$

a partir de esta "se puede demostrar de forma sencilla que dada una secuencia de entrada $x[n] = e^{j\omega n}$ para $-\infty < n < \infty$, la correspondiente salida de un sistema lineal e invariante con el tiempo con respuesta al impulso es

$$y[n] = H(e^{j\omega})e^{j\omega n}$$

Siendo

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}$$

[...] podemos ver que $H(e^{j\omega})$ describe el cambio en función de la frecuencia ω de la amplitud compleja de una señal de entrada de tipo exponencial compleja." [3]

Como se presentó en la ecuación 2.5, la función exponencial puede expandirse en una suma de valores reales e imaginarios

$$H(e^{j\omega}) = H_R(e^{j\omega}) + j\ H_I(e^{j\omega})$$

A partir de esta estructura se puede calcular la representación en frecuencia de la función. Para ello se realiza el cálculo de su amplitud y su ángulo como se muestra a continuación.

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j\angle H(e^{j\omega})}$$

2.3.1. Transformada discreta de Fourier

La transformada de Fourier, cuyo nombre recibe en honor a Jean Baptiste Joseph Fourier (1768-1830), permite obtener la representación de una señal en dominio del tiempo en el dominio de la frecuencia. La Transformada de Fourier en tiempo continuo para una función no periódica, o con periodo infinito, se define mediante la integral mostrada en la ecuación 2.6 [2]

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{2.6}$$

Donde t es el tiempo en segundos, ω es la frecuencia angular, x(t) es la señal que se desea analizar en el tiempo y $X(\omega)$ es el equivalente en frecuencia de la señal de entrada $x(\omega)$. Asimismo, este método nos permite el recuperar la señal en el dominio del tiempo a partir de la señal en frecuencia por medio de la integral de la ecuación 2.7 [2].

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}dt \tag{2.7}$$

En muchas ocasiones, y para efectos de notación, se suele representar la conversión entre dominios mediante la siguiente expresión

$$x(t) \leftrightarrow X(\omega)$$

El algoritmo de la Transformada de Fourier suele ser aplicado con mayor frecuencia en señales discretas. Esto debido a la versatilidad que presentan en la actualidad las computadoras. Por esta razón, se debe de discretizar la función. Para facilitar la aplicación se suele considerar que todas las señales que serán analizadas con este algoritmo siempre comenzarán en el tiempo cero y que tendrán un tamaño N de muestras. De esta forma se puede definir la función de la Transformada Discreta de Fourier (DFT por sus siglas en inglés *Discrete Fourier Transform*) como se menciona en la ecuación 2.8 [2].

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}, k = 0, 1, 2, ..., N-1 \tag{2.8}$$

Asimismo, si se discretiza la ecuación 2.7 bajo las mismas consideraciones se obtendrá la función de la transformada Inversa de Fourier Discreta (IDFT por sus siglas en inglés Inverse Discrete Fourier Transform) representada en 2.9 [2].

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N}, n = 0, 1, 2, ...N-1 \tag{2.9}$$

De esta forma, al emplear la ecuación 2.8 se puede generar un algoritmo con la capacidad de obtener la representación en frecuencia de una señal discreta de tiempo finito. El resultado que nos proporcionará el algoritmo de la Transformada Discreta de Fourier es una señal simétrica compleja que representa las distintas frecuencias contenidas en la señal de entrada como una serie de impulsos.

Dado que la señal obtenida es compleja, su representación gráfica no suele ser útil para analizar la información obtenida. Es común que se emplee el valor absoluto de la señal para interpretar los datos. Asimismo, de forma similar que $x^2(t)$ nos proporciona una noción de la energía de la señal en el tiempo, $|X|^2(\omega)$ nos proporciona una noción de la energía de las distintas frecuencias que se contienen. Finalmente, gracias a la simetría de la señal se podría ignorar la segunda mitad del conjunto de valores.

Por ejemplo, en la figura 2.3 se observa una señal simulada muestreada a 8000 [Hz] que contiene cuatro señales seno a 100, 500, 2000 y 3500 [Hz]. La representación en frecuencia que se obtiene con ayuda de la DFT se muestra en la primera parte de la figura, en donde se puede apreciar la distribución de las frecuencias en el espectro. Como se ha mencionado con anterioridad, la señal que nos proporciona la DFT muestra en su segunda mitad un equivalente de la primera, actuando como si fuera un espejo. Esto da la posibilidad de ignorar alguna de las dos partes, enfocando los análisis la que nos sea más conveniente. La segunda y tercera graficas de la figura 2.3 ejemplifica esta última mención.

2.3.2. Transformada rápida de Fourier

Una de las principales desventajas que contiene la DFT es su tiempo de ejecución en los algoritmos computacionales. Se observa en la ecuación 2.8 que el cálculo de cada valor k de la DFT implica N multiplicaciones complejas y $N-1$ sumas complejas. Por lo tanto, los N valores de la DFT pueden ser obtenidos tras N^2 multiplicaciones complejas y $N^2 - N$ sumas complejas. Esto último implica que el costo computacional para llevar a cabo el cálculo sea más caro mientras mayor sea la cantidad de valores en la señal. Por esta razón, se suele reemplazar este algoritmo por uno de menor costo y más rápido como lo es la Transformada Rápida de Fourier (FFT por sus siglas en inglés *Fast Fourier Transform* [2]).

La FFT considera las múltiples operaciones que se repiten a lo largo del desarrollo de

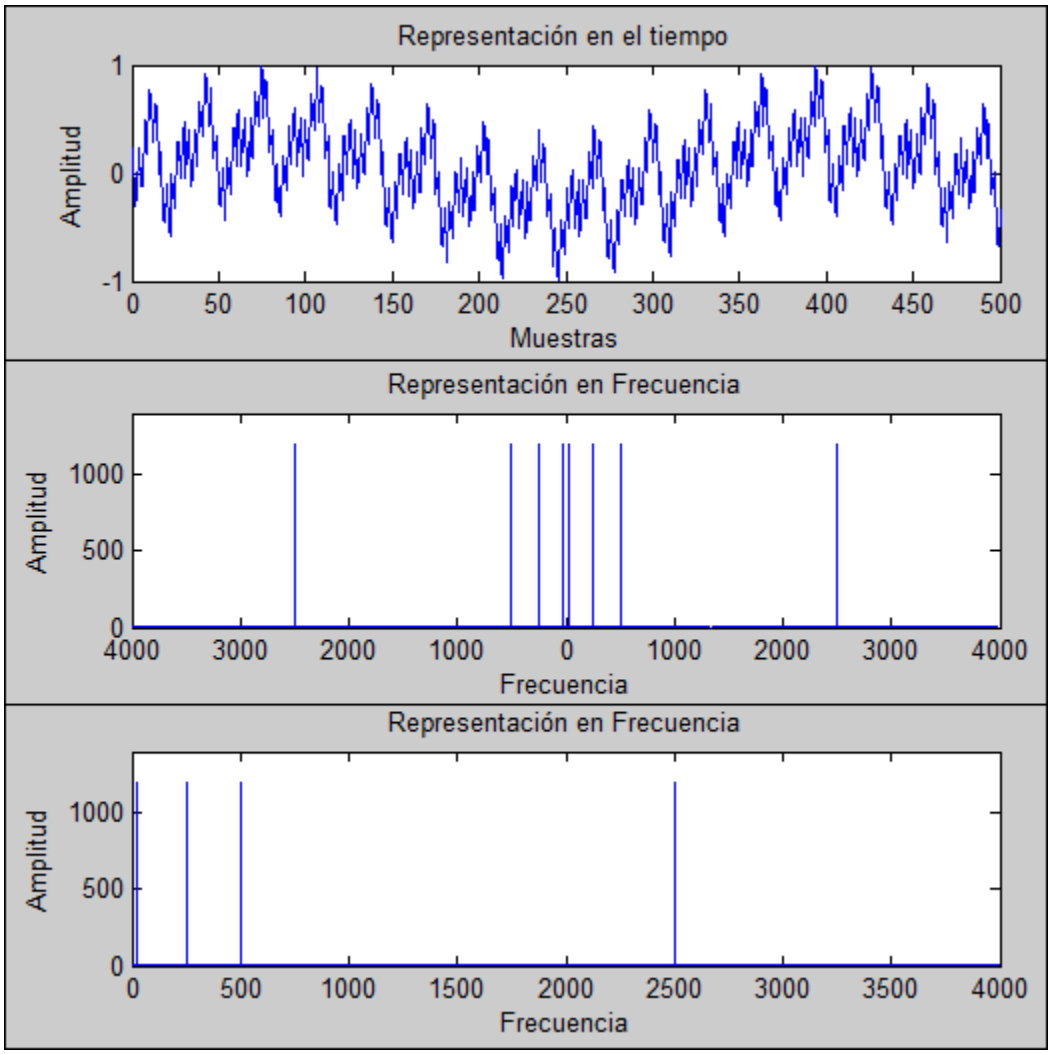


Figura 2.3: Ejemplo de la Transformada de Fourier

la DFT para agilizar el cálculo. Esto nos permite realizar combinaciones de operaciones para obtener los diferentes resultados. Dichas combinaciones se suelen representar en un diagrama de mariposa ofreciendo dos posibilidades para operar la información. La primera alternativa (figura 2.4) [3] se caracteriza por recibir la señal de entrada en orden y proporciona los valores de la salida en desorden (FFT con diezmado en frecuencia). Mientras que la segunda opción (figura 2.5) [3] requiere que los valores de la señal de entrada se presentan desordenados pero a la salida estos se encontraran ordenados (FFT con diezmado en el tiempo). El optar por alguno de los dos diagramas en específico dependerá de los requisitos al momento de emplear la FFT.

La forma en la que se operan los diagramas de mariposa parte del concepto de que el valor proveniente de las rectas horizontal se le sumará el valor proveniente de la flecha correspondiente. Para aquellas flechas que suben se mantendrá el signo, mientras que a las flechas que descienden se les multiplicará por un -1 el segundo valor. El valor de las Wk será multiplicado según se observa en la figura 2.6 [3].

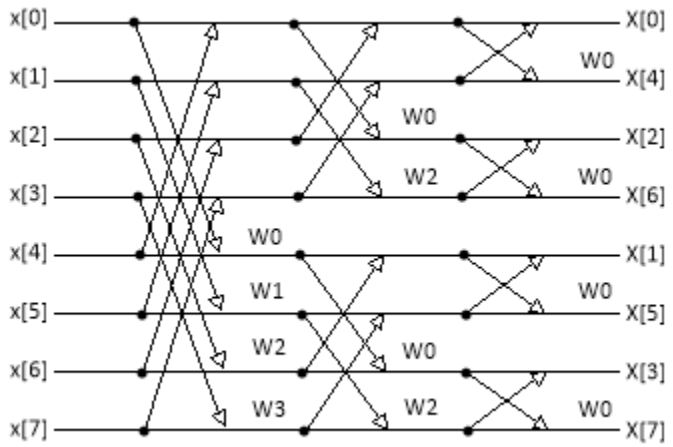


Figura 2.4: Diagrama de mariposa de la FFT con diezmado en frecuencia

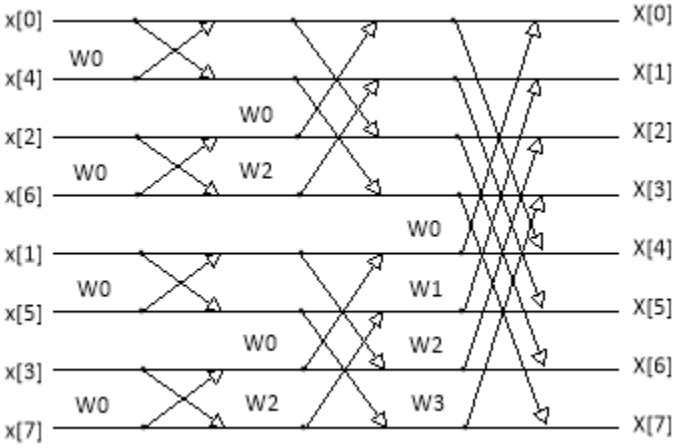


Figura 2.5: Diagrama de mariposa de la FFT con diezmado en el tiempo

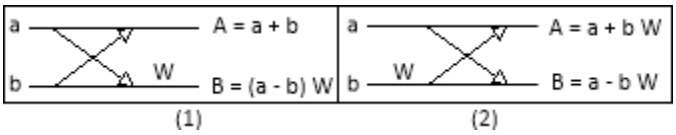


Figura 2.6: Interpretación de diagrama de mariposa para diezmado frecuencia (1) y en el tiempo (2)

Cabe mencionar que debido a la forma en que se realizan las combinaciones en los diagramas de mariposas, se requiere que el vector de entrada sea una potencia de dos. Una forma de cumplir con este requisito es quitando valores o añadiendo ceros a los datos de la señal para redondear la longitud al valor más cercano de potencia de dos.

La ecuación 2.8 también puede ser escrita como se presenta en la ecuación 2.10

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, k = 0, 1, 2, ..., N-1 \tag{2.10}$$

Donde, por definición

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

Que es la raíz n-ésima de la unidad [5].

Por tanto, podemos definir a las variables Wk como

$$W_k = (W_N)^k = (e^{-j2\pi/N})^k$$

Con estas se pueden calcular las combinaciones que se observan en las figuras 2.4 y 2.5, de cualquier orden. Como se observa en las estas imágenes, para el cálculo de la FFT se requiere obtener las Wk equivalentes a la mitad del número de datos de la señal de entrada.

2.4. Filtros digitales

Los filtros son sistemas que modifican determinadas características de su entrada. Un sistema se componen de un conjunto de elementos relacionados y que interactúan con una o más perturbaciones para producir una respuesta acorde a ella. En señales, la perturbación que reciben estos sistemas se conoce como señal de entrada, mientras que la respuesta que generan se denomina como señal de salida. Los filtros de señales generalmente son modelados en el dominio de la frecuencia ya que las características que alteran son las frecuencias que contienen. A continuación se plantea la función de transferencia que modela este tipo de sistemas en el dominio de la frecuenica.

$$H(s) = \frac{V_o(s)}{V_i(s)} \tag{2.11}$$

Donde s es la variable obtenida al transformar un sistema del dominio del tiempo al dominio de la frecuencia, y t representa la variable del tiempo. Mientras que $V_0(s)$ corresponde a la salida del sistema y $V_i(s)$ a la entrada. En la figura 2.7 se ejemplifica un diagrama de un sistema que responde a esta función de transferencia.

La función de transferencia del filtro puede ser expresada en términos de su magnitud y fase, al considerar la definición $s = j\omega$.

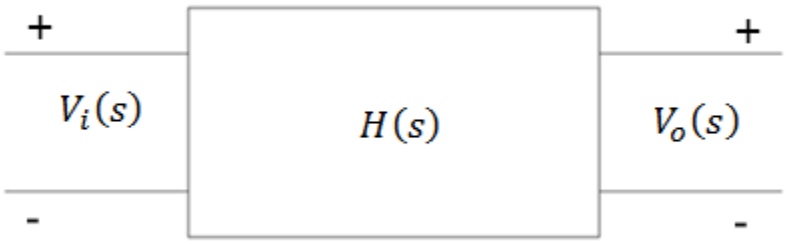


Figura 2.7: Diagrama de sistema con entrada y salida

$$H(j\omega) = |H(j\omega)|e^{j\phi(\omega)}$$

Esta forma de visualizar la función nos permite comprender que frecuencias son las que tienen mayor influencia dentro del sistema. Asimismo, permitirá realizar calculos con mayor facilidad, en algunos casos.

2.4.1. Análisis en el dominio del tiempo

Cuando se adquiere una señal, por lo general, esta estará representada en el tiempo. Esto dado que es la forma más fácil de almacenar la información. Basados en esto, se han desarrollado diversos métodos con los cuales se puede analizar las señales mientras permanecen en esta representación.

En el tiempo, la relación entrada – salida de los filtros puede ser representado por medio de las ecuaciones en diferencias de coefientes constantes, como la observada en la ecuación 2.12. Estas establecen que la suma de ciertos valores de entrada es igual a la suma de un conjunto valores de la salida, siempre que cada uno de ellos sea multiplicado por determinados coeficientes. En la estructura general de las ecuaciones en diferencia se considera que tanto para los datos de entrada como para los datos de salida se tendrán vectores de coeficientes b_n y a_m, respectivamente, con tantos valores como se estén sumando.

$$\sum_{n=0}^{N-1} b_n x_{k-n} = \sum_{m=0}^{M-1} a_m y_{k-m} \tag{2.12}$$

La representación en el dominio de z de las ecuaciones en diferencias nos permite manipular los ceros y polos del sistema como una combinación de los coeficientes bm y an. La representación general del sistema en función de estos valores se puede observar en la función 2.13 [3].

$$H[z] = \frac{Y[z]}{X[z]} = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{\sum_{n=0}^{N-1} a_n z^{-n}} = \frac{b_0 + b_1 z^{-1} + ... + b_{M-1} z^{-(M-1)}}{a_0 + a_1 z^{-1} + ... + a_{N-1} z^{-(N-1)}} \tag{2.13}$$

Para resolver el sistema se despeja de la ecuación 2.12 el valor presente de la señal de salida. Así se podría obtener el valor deseado o presente de la salida. Este quedará en función de los valores pasados de la salida y los valores presentes y pasados de la entrada, como se muestra en la ecuación 2.14.

$$a_0 y_k = \sum_{n=0}^{N-1} b_n x_{k-n} - \sum_{m=1}^{M-1} a_m y_{k-m} \tag{2.14}$$

A partir de la ecuación 2.14 es posible generar dos tipos de filtros, aquellos cuya respuesta al impulso es infinita y los que tienen respuesta al impulso finita. Donde los primeros se caracterizan por poseer parámetros a_m como b_n, es decir que su función de transferencia se compone de polos y ceros. Mientras que los segundos solo poseen parámetros b_n, lo que equivale a que la ecuación del filtro solo posee polos.

Los filtros de respuesta infinita al impulso (IIR, por sus siglas en inglés *Infinite Impulse Response*) se consideran recursivos ya que requieren tanto de los valores presentes y pasados de la señal de entrada como de los valores pasados de salida del sistema. Su función está representada por una ecuación en diferencias de coeficientes constantes con $a_0 = 1$, como se observa a continuación.

$$y_k = \sum_{n=0}^{N-1} b_n x_{k-n} - \sum_{m=1}^{M-1} a_m y_{k-m} \tag{2.15}$$

También se presenta el caso en el que los valores a_m de la ecuación en diferencias son considerados cero, con excepción de a_0. A este se le conoce como filtro de repuesta finita al impulso (FIR, por sus siglas en inglés *Finite Impulse Response*). Debido a la consideración realizada, la ecuación ya no involucra los valores pasados de la salida del sistema, por lo que este filtro no es recursivo. La ecuación de este filtro se puede observar a continuación.

$$y_k = \sum_{n=0}^{N-1} b_n x_{k-n} \tag{2.16}$$

Los filtros IIR y FIR, al estar basados en ecuaciones en diferencias, nos permiten el manipular las señales sin tener que transformar su información al dominio de la frecuencia. Los coeficientes a_m y b_n pueden ser calculados resolviendo la ecuación 2.13 para distintos lugares de los polos y ceros que la componen. Sin embargo, en la práctica estos

valores suelen ser obtenidos al modelar en el dominio discreto (z) los filtros analógicos. Para esto, se debe de considerar la función que se desea que el filtro realice, es decir, como se desea que se discriminen las frecuencias.

2.4.2. Análisis en el dominio de la frecuencia

Clasificación de filtros según la discriminación de frecuencias. Los filtros que se basan en la discriminación de frecuencias pueden presentar cuatro tipos de esquemas básicos: paso bajas, paso altas, rechaza banda y paso banda. Cada uno de ellos permite el paso de un conjunto específico de frecuencias, mientras que el resto son suprimidas.

Filtro paso bajas

Este filtro permite el paso de frecuencias bajas, inferiores a la frecuencia umbral o frecuencia de corte w_0. En la figure 2.8 se puede apreciar el comportamiento ideal de este filtro, en donde las frecuencias no son anuladas hasta después de pasar la frontera establecida por la frecuencia w_0.

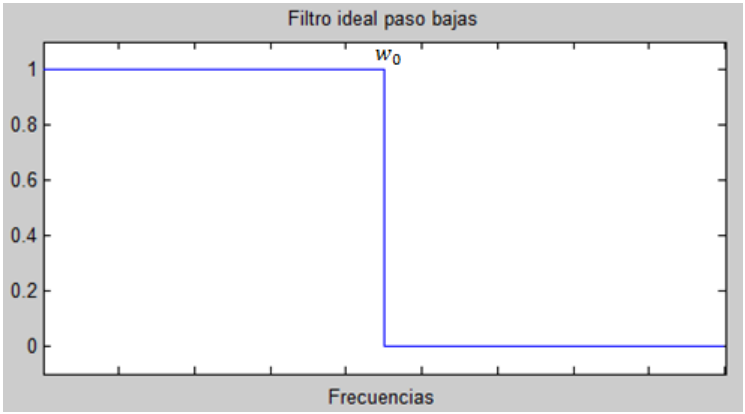


Figura 2.8: Comportamiento ideal de un filtro paso bajas

Filtro paso altas

Esta filtro permite el paso de frecuencias altas, superiores a la frecuencia de corte w_0, es decir, vuelve nulas todas las frecuencias menores a w_0 como se puede apreciar en la figura 2.9.

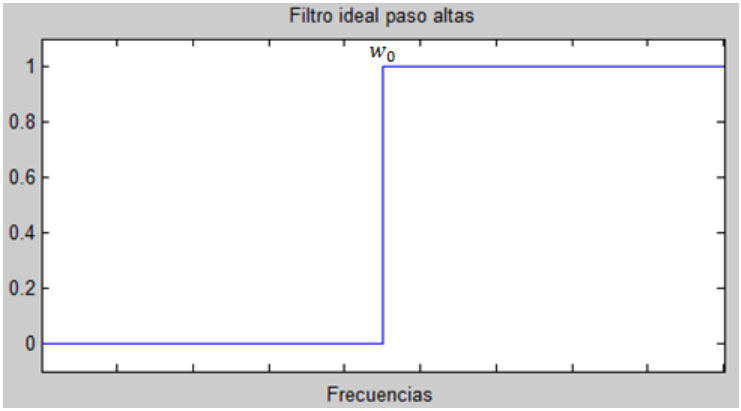


Figura 2.9: Comportamiento ideal de un filtro paso altas

Filtro paso banda

Este filtro permite el paso de determinadas frecuencias localizadas entre un rango especificado. Puede considerarse como la combinación de un filtro paso bajas con un filtro paso altas, donde la frecuencia de filtrado del primero es mayor que el del segundo. Su comportamiento ideal se puede observar en las figura 2.10 donde solo las frecuencias localizadas entre w_1 y w_2 no son atenuadas.

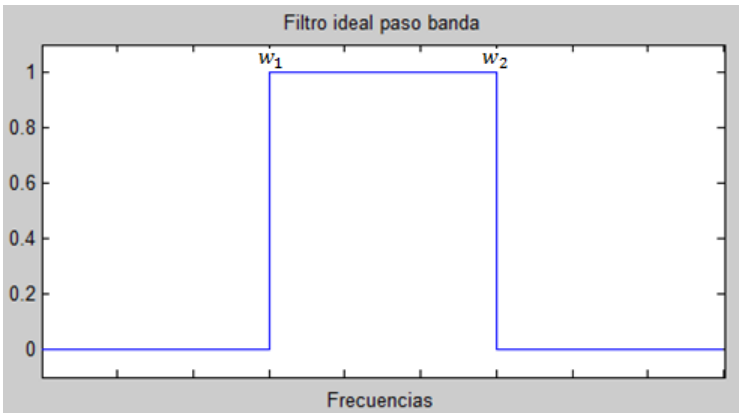


Figura 2.10: Comportamiento ideal de un filtro paso bandas

Filtro rechaza banda

Este filtro suprime las frecuencias contenidas en un rango especificado. En la Figura 2.11 se puede apreciar en comportamiento ideal de este filtro, donde las frecuencias entre w_1 y w_2 son eliminadas.

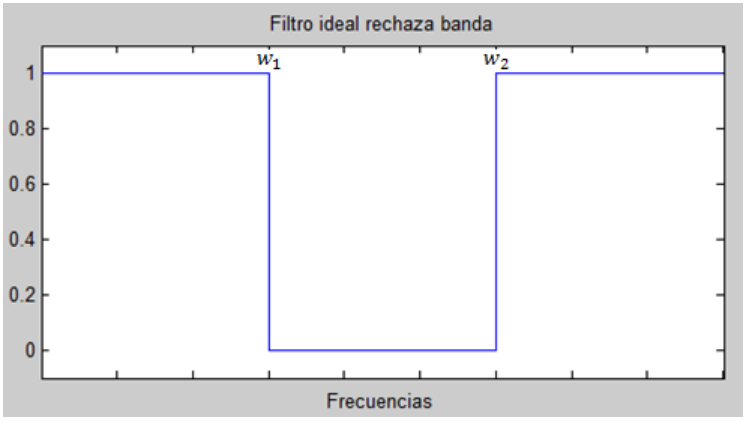


Figura 2.11: Comportamiento ideal de un filtro rechaza banda

2.4.3. Aproximaciones de los filtros analógicos

Los filtros basados en ecuaciones en diferencias de coeficientes constantes dependen de los valores que toman a y b para definir su comportamiento. Para filtros digitales, los coeficientes se suelen obtener mediante la resolución de las ecuaciones aplicadas en los filtros analógicos llevados al dominio de z. Ya que estos sistemas han sido largamente estudiados, es conveniente el poder aplicarlos. Dos de los diseños que se pueden aplicar se basan en los filtros analógicos *Butterworth* y *Chebyshev*.

Filtros Chebyshev

Los filtros diseñados a partir de Chevyshev (en honor al matemático ruso Pafnuti Lvóvich Chebyshov) se caracterizan por tener una banda de transición con caída rápida, pero con oscilaciones en la banda de paso o rechazo, como se ejemplifica en la figura 2.12. En estos, mientras mayor sea el orden del filtro, mayores serán las oscilaciones y menor será la banda de transición. El módulo de su función de transferencia está dado por la siguiente ecuación.

$$|H(j\omega)| = \frac{K_1}{\sqrt{1+\epsilon^2 C_n^2(\omega/\omega 0)}} \tag{2.17}$$

Donde K_1 y ϵ son valores contantes y $C_n(\omega)$ es un polinomio de Chebyshev de grado n.

En la figura 2.12 se puede observar el valor RW el cual corresponde a la distancia definida como $1 - 1/\sqrt{1+\epsilon^2}$, y representa la amplitud de las perturbaciones que se generan. Tomando la diferencia respecto de 0 [dB] (decibeles), queda:

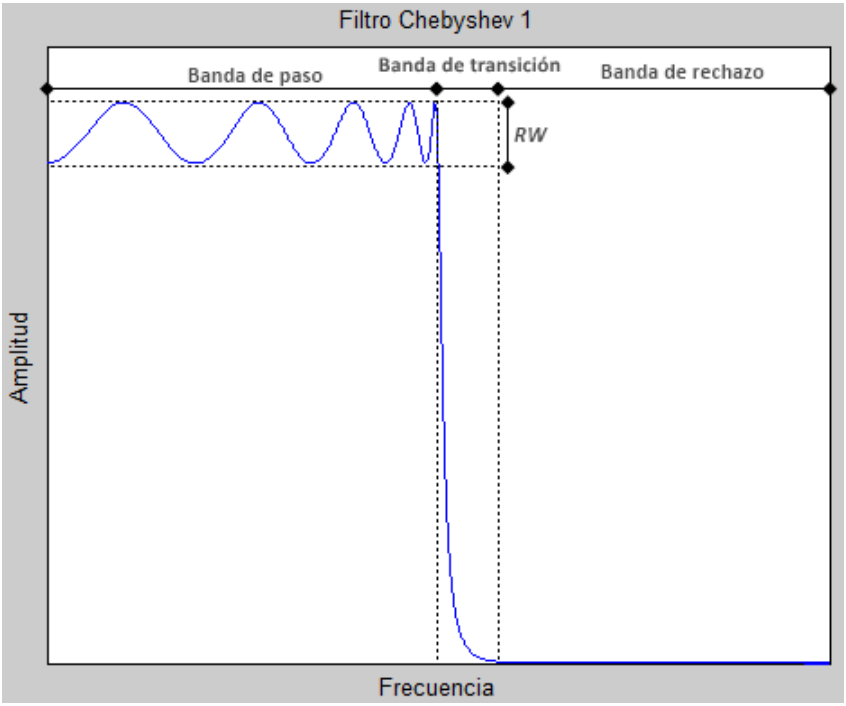


Figura 2.12: Comportamiento generalizado del filtro Chebyshev

$$RW|_{dB} = 20log\sqrt{1+\epsilon^2} \tag{2.18}$$

Así, el RW queda determinado por la elección de ϵ, el cual fluctua entre 0 y 1, entonces $RW = 1 - 1/\sqrt{2}$, es decir que el valor de RW varía entre 0 y 3[dB]. [7]

Este filtro es útil para los casos en que se desea una rápida anulación de las frecuencias no deseadas. Sin embargo, en aquellas frecuencias no suprimidas se podrán ver alteradas sus magnitudes. Dependiendo de la aplicación, estas podrían ser o no de relevancia.

Filtro Butterworth

El filtro Butterworth (en honor al ingeniero británico Stephen Butterworth) se caracteriza por tener una respuesta plana tanto en su banda de paso como en su banda de rechazo. Sin embargo posee una banda de transición con caída lenta, esta es de 20 [dB/Década] por cada polo, es decir, habrá una caida de 20[dB] cada vez que la frecuencia se incremente 10 veces respecto a la última caida (10, 100, 1000, etc. [Hz]). El incremento de polos en la función de transferencia también incrementa el tiempo de calculo del filtro. En la Figura 2.13 se pude observar un ejemplo de las componentes de este tipo de filtro. El módulo de la respuesta en frecuencia de este sistema está definido por la siguiente función.

$$|H(j\omega)| = \frac{G}{\sqrt{1 + (\omega/\omega_c)^{2n}}} \tag{2.19}$$

Donde G es la ganancia del filtro, w_c es la frecuencia de corte y $n = 0, 1, 2, ..., k$ es el orden del filtro.

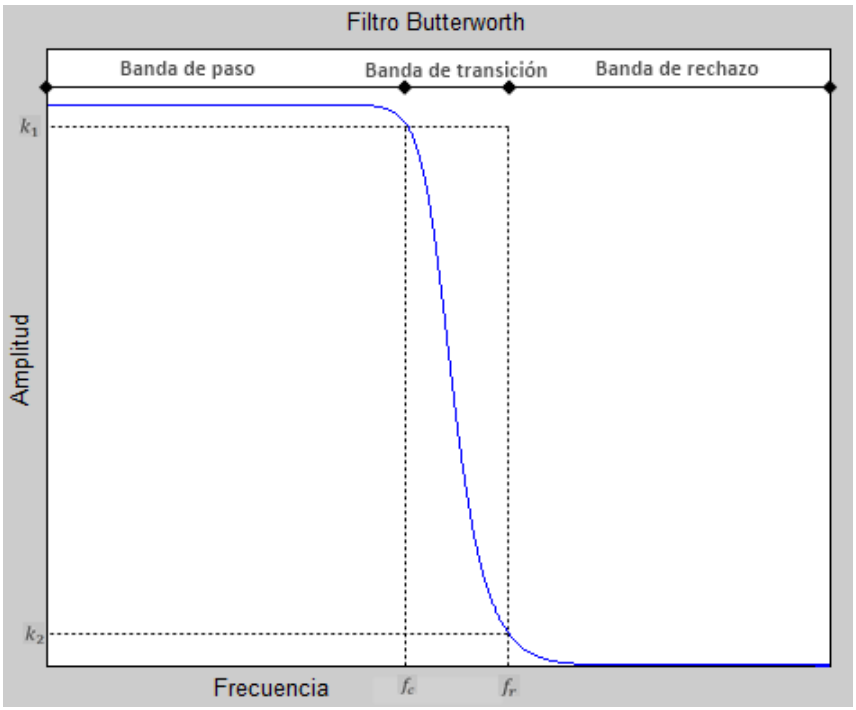


Figura 2.13: Comportamiento generalizado del filtro Butterworth

En la figura 2.13 podemos observar que se tienen dos frecuencias principales, la frecuencia de corte (f_c) y la frecuencia de rechazo (f_r), junto con sus respectivas ganancias, k_1 y k_2. El valor de f_c se suele localizar cuando hay una caída en la ganancia de equivalente a 3 [dB] del máximo. Mientras que f_r se suele asociar al punto en que la caída de la magnitud máxima de la señal es de 20 [dB].

Este filtro sele ser aplicado cuando se trata de no alterar la información de la señal original. Sin embargo, la disminución de la banda de transición implicaría el incremento en el tiempo de calculo del filtro.

2.5. Filtro de pre-énfasis y post-énfasis

El filtro de pre-énfasis y post-énfasis son una aplicación particular de los filtros. El primero de estos se usa para acentuar determinadas frecuencias de una señal. Mientras que en el segundo es aplicado para realizar el acto inverso [10].

En algunos casos se posee información frecuencia dentro de una señal que tiene una baja ganancia debido a su origen. Sin embargo esa información es de relevancia para la persona que la está analizando. Por esta razón, se suele aplicar un filtro de pre-énfasis, el cual aumenta la magnitud de esas frecuencias de relevancia sin suprimir alguna otra.

La función de transferencia del filtro de pre-énfasis se define de la siguiente forma

$$H[z] = 1 + az^{-1} \tag{2.20}$$

Donde el valor de a es una constante que define el comportamiento del filtro. Por ejemplo en la figura 2.14 se puede observar el caso para el cual $a = -0{,}9$, cuando la señal de entrada es un impulso en el tiempo [10]. Como se puede observar con este valor negativo las frecuencias altas son incrementadas, mientras que las frecuencias bajas son atenuadas pero sin llegar a suprimirlas.

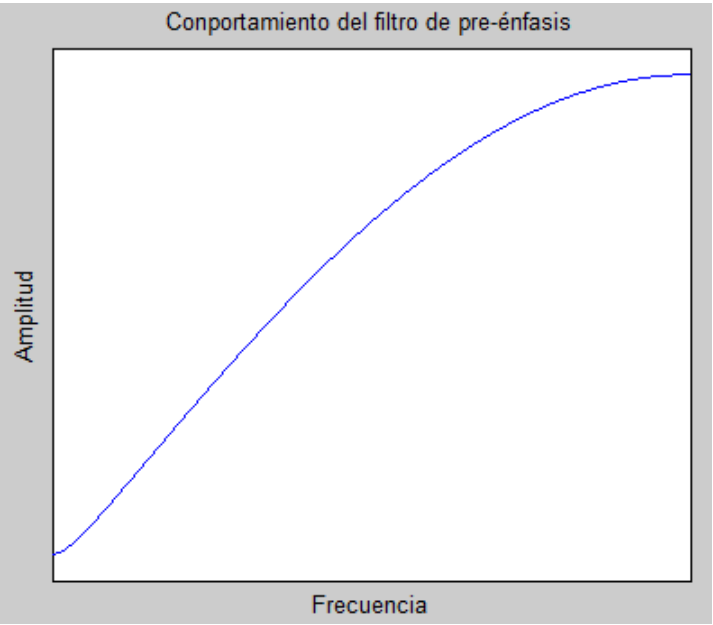


Figura 2.14: Comportamiento del filtro de pre-énfasis, con $a = -0{,}9$

Para el filtro de post-énfasis, la función de transferencia es la siguiente

$$H[z] = 1 + az \tag{2.21}$$

Para este caso las frecuencias serán mayormente atenuadas sin ser suprimidas. La acción que realiza es exactamente el contrario del filtro de pre-énfasis, por lo que si se aplicará este filtro a la señal que se obtuvo con el primer método se recuperaría la señal original. En la figura 2.15 se puede observar el comportamiento de dicho filtro para una entrada impulso, con $a = -0{,}9$.

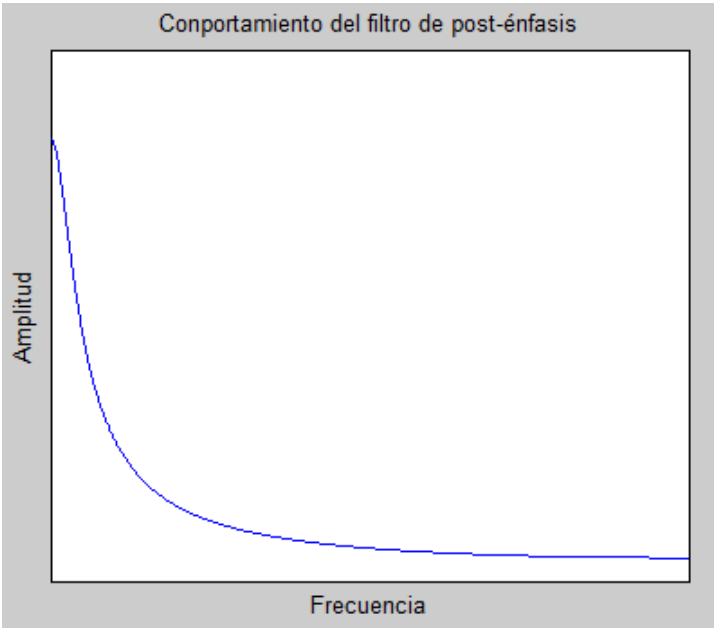


Figura 2.15: Comportamiento del filtro de post-énfasis, con $a = -0{,}9$

Capítulo 3

Análisis y procesamiento de señales aleatorias

En la práctica, muchos de los fenómenos que se analizan suelen tener comportamientos poco predecibles, es decir, los sistemas suelen actuar de forma aleatoria. En señales se observa que el comportamiento de la onda no presenta un periodo o un patrón en específico. A estos fenómenos se les suele denominar como aleatorios o estocásticos.

Los métodos de análisis aplicados a este tipo de fenómenos se suelen basar en los conceptos de probabilidad. A partir de estos es posible obtener información relevante de las señales aleatorias de forma similar que con los métodos aplicados a las señales determinísticas.

Dentro de la rama de las señales aleatorias, uno de los fenómenos más recurrentes es el ruido. Este tipo de señal se caracteriza por no contener información relevante y permanecer de forma parasita en otras señales. Pueden tener su origen en el mismo sistema que la señal a analizar, estar presente de forma natural en el medio, generarse como consecuencia de los instrumentos de medición, originarse como consecuencia de la forma como se manipula la información capturada, entre otras.

El ruido blanco es un tipo de ruido que se caracteriza por contener todas las frecuencias en su espectro. Cuando esta señal tiene una distribución Gaussiana se dice que es un ruido blanco Gaussiano. En este caso se tiene que las potencias de las distintas frecuencias del espectro tienen la misma potencia, es decir que posee una densidad de potencia constante.

3.1. Variables aleatorias

"Una variable aleatoria es una función que asigna un número real a cada resultado del espacio muestra de un experimento aleatorio" [8]. En otras palabras, una variable aleatoria puede interpretarse como el conjunto de valores que un sistema genera y cuyo comportamiento no posee un patrón específico. Sin embargo, esto no significa que los datos no representen información relevante.

"Una variable aleatoria discreta es una variable aleatoria con un rango finito" [8], es decir que solo puede tomar valores enteros de un conjunto especificado. Mientras que "una variable aleatoria continua es una variable aleatoria que tiene como rango un intervalo de números reales" [8], por lo que los posibles valores que puede tomar son infinitos. En probabilidad a una variable aleatoria se le suele denominar con una letra mayúscula (X), mientras que a sus posibles valores con una letra minúscula (x).

3.2. Conceptos básicos de probabilidad

Los métodos de análisis y procesamiento de las variables aleatorias se basan en conceptos de probabilidad y estadística. Esto a causa de que los valores proporcionados no pueden ser modelados por una función matemática explicita.

Distribución de probabilidad

"La distribución de probabilidad de una variable aleatoria X es una descripción de las probabilidades asociadas con los valores posibles de X. Para una variable aleatoria discreta, es común especificar la distribución como una lista de valores posibles junto con la probabilidad de cada uno." [8]. La expresión matemática que representa este concepto se puede observar a continuación. En esta el resultado solo puede variar entre los límites cerrados de 0 y 1.

$$P(X = x_i) = y_i \tag{3.1}$$

Rango

El rango se define como la diferencia entre el valor máximo y mínimo registrado en el conjunto de datos.

$$Rango = valor_{maximo} - valor_{minimo} \tag{3.2}$$

Distribución discreta uniforme

"Una variable aleatoria X es una variable aleatoria discreta uniforme si cada uno de los valores de su rango tienen la misma probabilidad" [8]. Esta probabilidad se puede encontrar al sacar el inverso del número de valores (n) que tiene la variable aleatoria.

$$f(x_i) = 1/n \tag{3.3}$$

Media de una variable aleatoria discreta

La media o promedio es la ponderación de un conjunto de valores finitos con una distribución uniforme.

$$\bar{x} = \mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.4}$$

Varianza de una variable aleatoria discreta

La varianza es una medida de dispersión que compara la diferencia al cuadrado de cada valor del conjunto de datos con la media de estos bajo un mismo factor. "Su fórmula matemática para el caso de datos referentes a una muestra es:

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{3.5}$$

Y para el caso de datos de una población es dada por" 3.6.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{3.6}$$

Donde n corresponde al número de datos que se están operando.

Desviación estándar

La desviación estándar es una medida de dispersión de los valores respecto a su media geométrica. Esta muestra si los valores están proximos o no de la media cuando el valor resultante es bajo o alto, respectivamente. Esta se define como la raíz cuadrada de la varianza, e igual que esta está definida tanto para una población o un conjunto de muestras

Para el caso de un conjunto de muestras se tiene que

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.7}$$

Asimismo, para el caso de una población sabe que

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.8}$$

Función Gaussiana

La función Gaussiana se caracteriza por su simetría y su representación en forma de campana (campana de Gauss) y queda definida mediante la siguiente función

$$f(x) = ae^{\frac{-(x-c)^2}{2b^2}} \tag{3.9}$$

donde a controla la altura de la función, b define lo ancho de la misma y c la localización del centro. El comportamiento general de esta función es el representado en la siguiente figura 3.1.

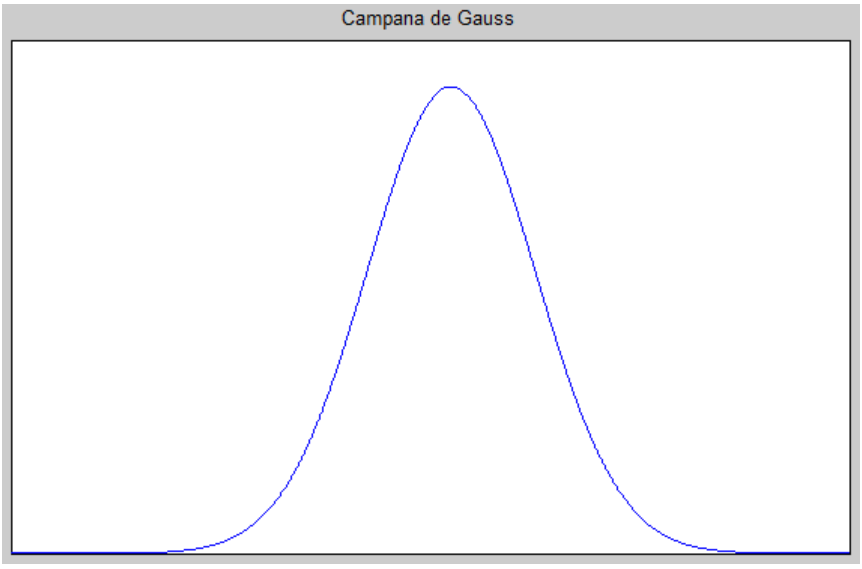


Figura 3.1: Comportameinto de la función Gausiana

Distribución de Poisson

"Distribución de probabilidad discreta que se aplica a las ocurrencias de algún suceso durante un intervalo de tiempo específico [... que] puede ser tiempo, distancia, área, volumen o alguna unidad similar" [8].

$$P(x) = \frac{\mu^x e^{-\mu}}{x!} \tag{3.10}$$

Donde la variable μ indica el promedio de veces que el fenómeno se presenta en el intervalo, y x los éxitos de los que se desea saber su probabilidad de ocurrencia.

Distribución normal

La distribución normal establece la probabilidad de que un determinado valor pertenezca al conjunto estudiado. Para ello emplea la función de la campana de Gauss junto con la media y la desviación estándar. La función se centra en la media y establece que los valores más cercanos a esta tendrán una mayor probabilidad de éxito. La función empleada para calcular estas probabilidades es la siguiente.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.11}$$

A partir de este concepto se puede establecer un intervalo que cubre la mayoría de los valores del conjunto analizado. Este se genera a partir de la sustracción y suma de la desviación estándar a la media. Mientras mayor sea el agregado que se le hace a la media, mayor será el porcentaje de valores que se cubren. Por ejemplo, a una desviación estándar se estarían cubriendo 68.26 % del total de las muestras, mientras que a dos deviaciones estándar se alcanzaría el 95.4 % y a tres deviaciones el 99.7 %.

Valores atípicos

Los valores atípicos son aquellos datos del conjunto cuyo comportamiento no van acorde al del resto de datos. Suelen presentarse de forma aleatoria alterando el comportamiento general de la información, como el promedio, la varianza o la energía en las que se involucran todos los datos.

En muchas ocasiones se busca el eliminar estos valores al considerarla como información sin utilidad. Pero también se puede presentar casos en los que estos valores atípicos no deban de ser eliminados ya que tienen una relevancia importante en nuestros resultados. Por ejemplo, cuando se realizan mediciones de alguna señal eléctrica,

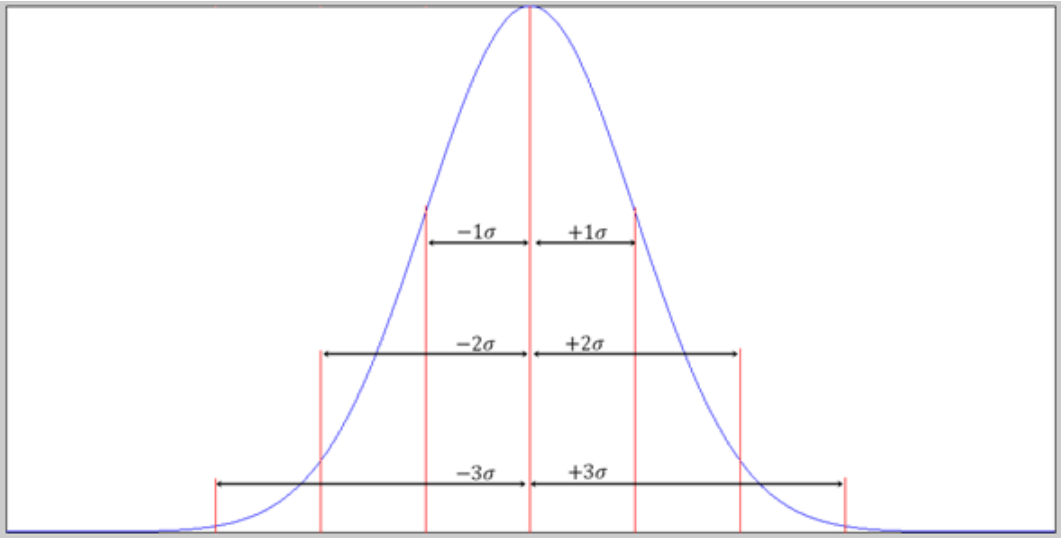


Figura 3.2: Representación de distintas distribuciones en la campana de Gauss

se pueden presentar algunos picos cuyo origen está relacionado con el instrumento de medición. Estos valores podrían ser despreciables ya que no representan información relevante del sistema que se está monitoreando. Asimismo, podríamos registrar picos cuyo origen provenga de nuestro sistema de forma aleatoria. Estos podrían representar información de relevancia y que nos indicaría un mal funcionamiento.

El cálculo de los valores atípicos se lleva a cabo mediante los siguientes pasos

- Se deben de ordenar los datos de menor a mayor
- Se calcula la mediana (Q_2) del conjunto ordenado
- Se calculan el primero (Q_1) y tercer cuartil (Q_3) que equivalen a los datos en la posiciones equivalentes al 25 % y 75 % del total.
- Los rangos internos están definidos de la siguiente forma

$$limiteInterno1 = Q_2 - (Q_3 - Q_1) * 1{,}5 \tag{3.12}$$

$$limiteInterno2 = Q_2 + (Q_3 - Q_1) * 1{,}5 \tag{3.13}$$

- Los rangos externos del conjunto están definidos por las siguientes formulas

$$limiteExterno1 = Q_2 - (Q_3 - Q_1) * 3 \tag{3.14}$$

$$limiteInterno2 = Q_2 + (Q_3 - Q_1) * 1{,}5 \tag{3.15}$$

Aquellos datos que permanecen dentro de los límites internos no serán eliminados. Los datos que se localicen entre el rango interno y el externo serán considerados ligeramente atípicos y podrían o no ser eliminados. Todos aquellos que se localicen por encima del rango externo serán considerados como atípicos y deberán de ser evaluados para determinar si son eliminados.

Error de predicción y error cuadrático

El error de predicción proporciona la diferencia existente entre un sistema real y un sistema que trata de simular el comportamiento del primero.

$$e(n) = x - x' \tag{3.16}$$

Donde $x(n)$ corresponde a los valores originales y $x'(n)$ son los valores del sistema simulado. El error cuadrático medio (ECM) se calcula mediante la siguiente

$$ECM = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x(i) - x'(i))^2} \tag{3.17}$$

3.2.1. Procesos aleatorios estacionarios y no estacionarios

Un proceso estocástico se compone por una colección de variables aleatorias provenientes de una misma fuente, relacionadas por un parámetro en común. En la materia de señales este parámetro es el tiempo. Cada componente del conjunto puede presentar sus propias características estadísticas, es decir que estas propiedades podrían variar respecto al tiempo[9].

El comportamiento futuro de los procesos aleatorios suele ser difícil de predecir, dado que no se posee ningún modelo que los represente. Por esta razón, se emplean las propiedades estadísticas para comparar comportamientos entre conjuntos de la misma naturaleza. Con este método es posible predecir el comportamiento de una señal aleatoria analizando las propiedades de otras similares.

Cuando un proceso estocástico no presenta variaciones en sus propiedades estadísticas respecto al tiempo se clasifica como proceso estacionario y se define de la siguiente forma.

"Se dice que un proceso aleatorio $x(t)$ es estacionario hasta el orden N si, para cualquier $t_1, t_2, ..., t_N$

$$f_x(x(t_1), x(t_2), ..., x(t_N)) = f_x(x(t_1 + t_0), x(t_2 + t_0), ..., x(t_N + t_0))$$

Donde t_0 es cualquier constante real arbitraria. Aún más, se dice que el proceso es estrictamente estacionario si es estacionario al orden de $N \to \infty$" [9]

Un proceso aleatorio es estacionario en sentido estricto "si ninguna de sus propiedades estadísticas son afectadas por un desplazamiento"[9] en el tiempo. Asimismo se

le llama proceso estacionario en sentido amplio "si la media y la auto correlación no varían con un desplazamiento en el tiempo"[9].

Los procesos no estacionarios son aquellos cuyas propiedades estadísticas varían con forma el tiempo cambia.

3.3. Correlación y auto-corrección

La correlación determina la relación o similitud existente entre dos señales. Compara una señal x con una señal y para distintos desfasamientos y expone para cada uno de estos un valor de similitud. En total de datos representa la relación que tendrá la segunda señal con la primera a los largo del tiempo. A continuación se cita un ejemplo de aplicación de la correlación extraído del libro Tratamiento Digital de Señales de John G. Proakis [5].

Supongamos que disponemos de dos señales $x(n)$ e $y(n)$ que deseamos comparar. En las aplicaciones de radar y sonar, $x(n)$ puede representar la versión muestreada de la señal transmitida e $y(n)$ puede representar la versión muestreada de la señal recibida en la salida del convertidor analógico – digital. Si hay un blanco en el espacio en que el radar o el sonar está barriendo, la señal recibida $y(n)$ estará formada por una versión retardada de la señal transmitida y distorsionada por efecto del ruido aditivo.

Podemos representar la secuencia recibida como

$$y(n) = \alpha x(n - D) + w(n)$$

donde α es un factor de atenuación que representa la perdida de la señal que se produce en la transmisión de ida y vuelta que sigue la señal $x(n)$, D es el retardo de ida y vuelta, que se supone que es un múltiplo entero del intervalo de muestreo y $w(n)$ representa el ruido aditivo que capta la antena [...]. Por el contrario, si no hay un blanco en el espacio barrido por el radar o sonar, la señal recibida $y(n)$ es solamente ruido.

Disponiendo de las dos señales, $x(n)$, [señal de referencia], e $y(n)$, [señal recibida], el problema de la detección por el radar o sonar consiste en comparar $y(n)$ y $x(n)$ para determinar si hay presente un blanco y, en caso afirmativo, determinar el tiempo de retardo D y calcular la distancia al blanco. [5]

Al comparar las dos señales por medio del método de correlación es posible obtener el retardo de la señal (D) al considerar que la señal recibida no es la misma excepto cuando se detecte el blanco. En la figura 3.3 se pueden observar dos señales, la primera es un conjunto de senoidales de diferentes frecuencias colocadas de forma consecutiva. La segunda señal muestra un trozo aleatorio de la primera con ruido blanco. La tercera señal presenta la correlación de las dos anteriores, en donde se puede observar que hay

un comportamiento peculiar en la posición de la señal original de donde se extrajo la segunda. Esta región puede ser empleada para determinar el retardo de la señal y con ello la distancia del objeto que se presenta en el radar.

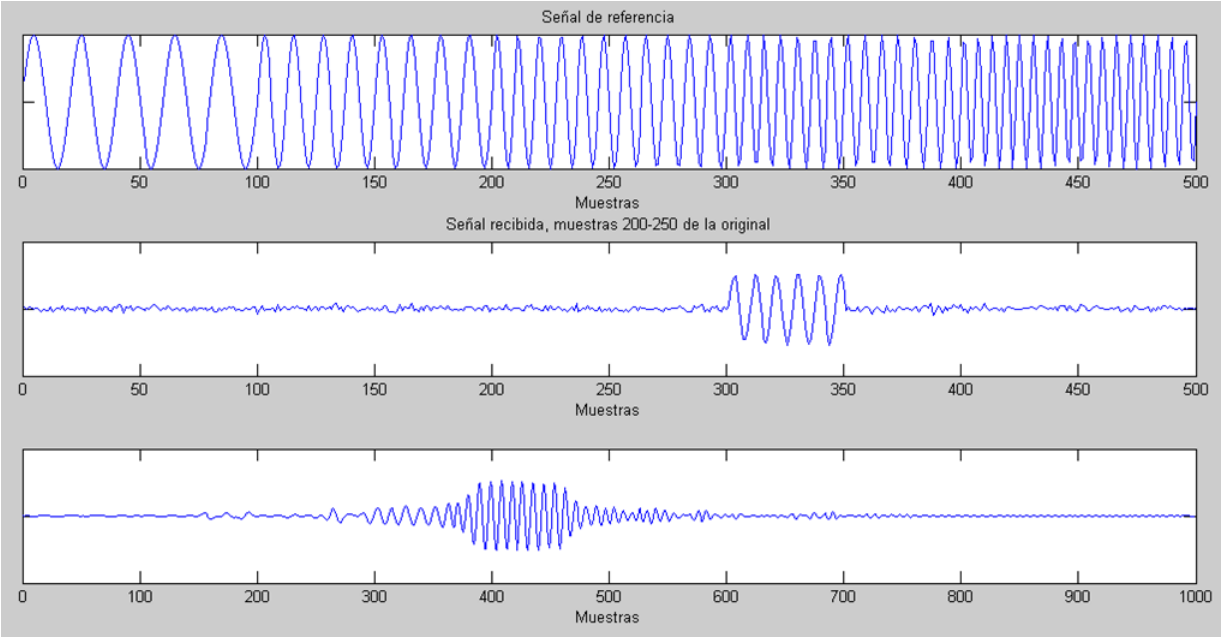


Figura 3.3: Ejemplo de Autocorrelación de una señal de referencia con un trozo de señal extraída de la misma con ruido

3.3.1. Correlación cruzada y autocorrelación

Para calcular la correlación de dos señales, $x(n)$ e $y(n)$ de tamaño finito se emplea la correlación cruzada, definida como a continuación [5]:

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) \tag{3.18}$$

Asimismo podemos reescribir la ecuación de la siguiente forma

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n) \tag{3.19}$$

La variable l representa el desfasamiento entre las señales, e indica que señal será la que se desplaza. En la primera ecuación la señal y se desplaza sobre la señal x, mientras que en la segunda y permanece estática.

Al caso particular en el que $x = y$ se le conoce como Autocorrelación. Esta indica la relación que posee una señal consigo misma a lo largo del tiempo, por ejemplo en una señal periódica se observarían oscilaciones periodicas, a diferencia de una señal de ruido blanco cuya Autocorrelación asemejaría una señal impulso. En la imagen 3.4 se ejemplifican las autocorrelaciones de una señal periódica y una de ruido blanco. La función de Autocorrelación puede ser definida por la ecuación 3.20 o 3.21.

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) \tag{3.20}$$

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n+l)x(n) \tag{3.21}$$

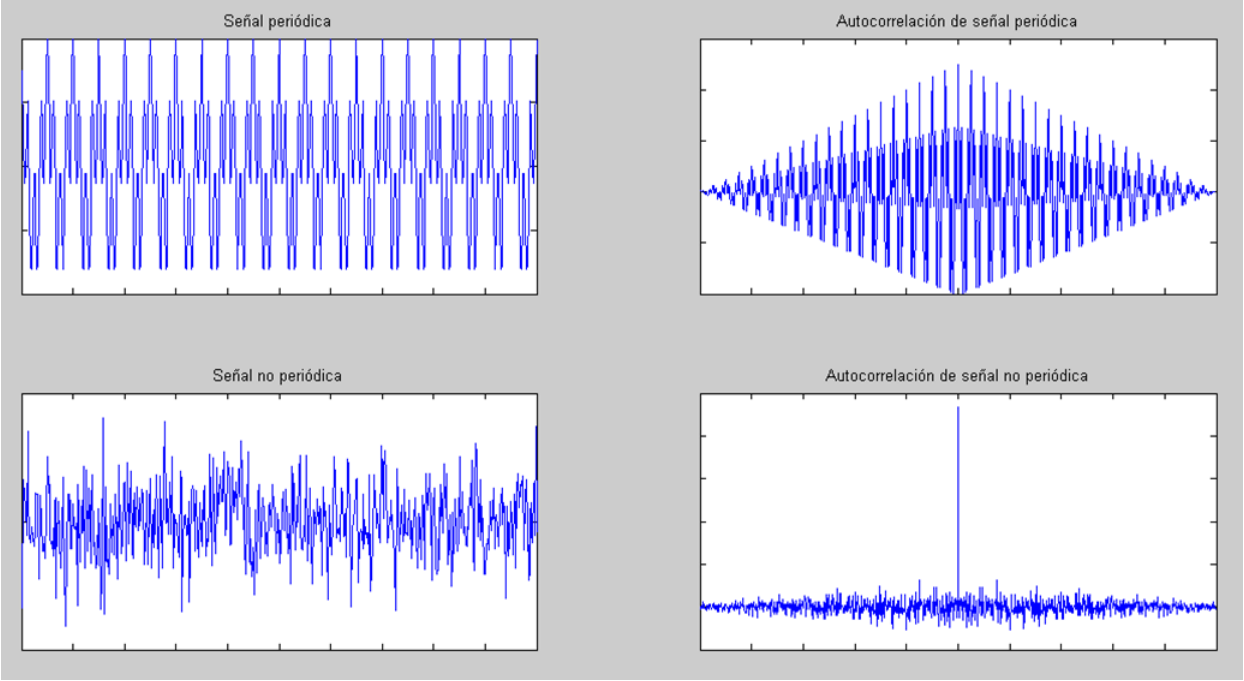


Figura 3.4: Ejemplo de señal periódica y señal no periódica, junto con sus respectivas autocorrelaciones

Considerando señales de duración finita, la función de correlación se redefine de la siguiente forma

$$r_{xy}[l] = \sum_{n=i}^{N-|k|-1} x[n]y[n-l] \tag{3.22}$$

Donde

$$i = 0 \;\; y \;\; k = 0 \;\; \forall \;\; l \geq 0$$

e

$$i = l \;\; y \;\; k = l \;\; \forall \;\; l < 0$$

Asimismo podemos definir el tamaño del vector r como $2N - 1$, siendo su rango de valores

$$-(N-1) \geq l \geq N - 1$$

A continuación se ejemplifica la ecuación 3.22 con un par de señales de tamaño $N = 4$

Dadas las señales x e y de tamaño $N = 4$, definidas de la siguiente forma

$$x = [x_0, x_1, x_2, x_3]$$
$$y = [y_0, y_1, y_2, y_3]$$

Se sabe que el tamaño del vector r será igual a $2N - 1$, es decir, tendrá 7 datos en total. Cabe recordar que la Correlación establece que los valores antes o después de los contenidos en x e y toman el valor de cero. Por lo tanto, si se desarrolla la ecuación 3.22 para los 7 valores del vector r, se tendrían los siguientes comportamientos

$$r[-3] = \sum_{n=-3}^{0} x_n y_{n+3} = x_{-3}y_0 + x_{-2}y_1 + x_{-1}y_2 + x_0 y_3 = x_0 y_3$$

$$r[-2] = \sum_{n=-2}^{1} x_n y_{n+2} = x_{-2}y_0 + x_{-1}y_1 + x_0 y_2 + x_1 y_3 = x_0 y_2 + x_1 y_3$$

$$r[-1] = \sum_{n=-1}^{2} x_n y_{n+1} = x_{-1}y_0 + x_0 y_1 + x_1 y_2 + x_2 y_3 = x_0 y_1 + x_1 y_2 + x_2 y_3$$

$$r[0] = \sum_{n=0}^{3} x_n y_n = x_0 y_0 + x_1 y_1 + x_2 y_2 + x_3 y_3$$

$$r[1] = \sum_{n=0}^{3} x_n y_{n-1} = x_0 y_{-1} + x_1 y_0 + x_2 y_1 + x_3 y_2 = x_1 y_0 + x_2 y_1 + x_3 y_2$$

$$r[2] = \sum_{n=0}^{3} x_n y_{n-2} = x_0 y_{-2} + x_1 y_{-1} + x_2 y_0 + x_3 y_1 = x_2 y_0 + x_3 y_1$$

$$r[3] = \sum_{n=0}^{3} x_n y_{n-3} = x_0 y_{-3} + x_1 y_{-2} + x_2 y_{-1} + x_3 y_0 = x_3 y_0$$

Como se puede observar el número de multiplicaciones totales es la misma para todos los casos. Sin embargo, gracias a la consideración de los valores iguales a cero las multiplicaciones que realmente influyen en los resultados disminuyen en todos los casos excepto para $l = 0$. En total, incluyendo las multiplicaciones que dan como resultado cero, se realizan $N(2N-1)$ multiplicaciones y $(N-1)(2N-1)$ sumas. Esto último implica un gran coste de cálculo en dado caso de que N sea muy grande.

Se puede destacar, del ejemplo anterior, que en ningún caso las multiplicaciones se repiten, lo cual permite que el método de la correlación proporcione información distinta para cada caso de r. Asimismo, para el caso de la Auto-correlación, se podría destacar que el valor con mayor magnitud correspondería a $l = 0$, ya que contiene toda la información del vector. En otras palabras, para $l = 0$ en la Auto-correlación se tiene expresada la energía de la señal. También, se puede considerar el evitar calcular los valores cuando $l < 0$ dado que serán equivalentes a los proporcionados cuando $l > 0$.

3.4. Densidad espectral

Esta es una característica de las señales que especifica la distribución de la energía o la potencia de las mismas en el dominio de la frecuencia.

3.4.1. Densidad espectral de energía

Es posible calcular la energía de una señal en el tiempo mediante la ecuación 2.1. Analogamente se puede calcular la energía de una señal en el dominio de la frecuencia mediante la ecuación 3.23. Para ello se emplea la correspondiente representación en frecuencia de la señal ($x(t) \rightarrow X(f)$), obtenida mediante Fourier. Este valor representa la energía de la señal por unidad de ancho de banda (joules/Hertz)

$$E_X = \int_{-\infty}^{\infty} |X(f)|^2 df \tag{3.23}$$

3.4.2. Densidad espectral de potencia

La densidad espectral de potencia describe la distribución de potencia de una señal en el rango de frecuencias que esta comprende. La forma en que se calcula esta se describe a continuación [9]:

Primero se define la versión truncada de la forma de onda a través de

$$w_T(t) = \left\{ \begin{array}{ccc} w(t) & \forall & -T/2 < t < T/2 \\ 0 & , & \text{en cualquier otro caso} \end{array} \right\} = w(t) \prod \frac{t}{T}$$

Dada la definición de potencia que se puede escribir como a continuación

$$P = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} w^2(t)dt$$

Se obtiene la potencia promedio normalizada

$$P = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} w^2(t)dt = \lim_{T\to\infty} \frac{1}{T} \int_{-\infty}^{\infty} w_T^2(t)dt$$

Empleando el teorema de Perseval para una sola señal, definido en la ecuación 3.24

$$\int_{-\infty}^{\infty} |w_1(t)|^2 dt = \int_{-\infty}^{\infty} |W_1(f)|^2 dt \tag{3.24}$$

La potencia promedio normalizada se convierte en

$$P = \lim_{T\to\infty} \frac{1}{T} \int_{-\infty}^{\infty} |W_T(f)|^2 df = \int_{-\infty}^{\infty} \lim_{T\to\infty} \frac{|W_T(f)|^2}{T} df$$

Donde W_T (f) es la transformada de Fourier de $w_T(t)$

La densidad espectral de potencia (PSD) es

$$P_w(f) \equiv \lim_{T\to\infty} \frac{|W_T(f)|^2}{T}$$

Donde $w_T(t) \leftrightarrow W_T(f)$ y $P_w(f)$ tiene unidades de watts por herz

La densidad espectral de potencia para un proceso aleatorio x(t) se obtiene de

$$P_X(f) = \lim_{T\to\infty} \frac{\overline{|X_T(f)|^2}}{T}$$

Donde X_T es la transformada de Fourier de $x_T(t)$ y

$$x_T(t) = \left\{ \begin{array}{ccc} x(t) & \forall & |t| < T/2 \\ 0, & , & \text{en cualquier otro caso} \end{array} \right\}$$

3.5. Segmentación de una señal

Muchos de los algoritmos que se emplean para el análisis de señales requieren que estas estén compuestas por una cantidad finita y reducida de datos. Esta condicionante es con el fin de reducir el coste de tiempo de ejecución de los programas computacionales. Para dar solución a este problema se suele fragmentar el vector de la señal a analizar en una matriz de dimensiones $P \times Q$. Donde P son la cantidad de segmentos en que se divide el vector y Q es la cantidad de datos por cada segmento, asumiendo que el tamaño del vector es igual a la multiplicación de las constantes P, Q.

Dada una señal en tiempo discreto de tamaño N, definida por el siguiente vector

$$x[k] = [x[1], x[2], ..., x[N]]$$

Se puede generar una matriz al segmentar en tamaños iguales de la forma en que se muestra en la siguiente cuadro

		Datos por segmento				
		1	**2**	**3**	...	***Q***
Segmentos	**1**	$x[1]$	$x[2]$	$x[3]$	...	$x[Q]$
	2	$x[Q+1]$	$x[Q+2]$	$x[Q+3]$	...	$x[Q+Q]$
	3	$x[2Q+1]$	$x[2Q+2]$	$x[2Q+3]$	...	$x[2Q+Q]$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
	P	$x[(P-1)Q+1]$	$x[(P-1)Q+2]$	$x[(P-1)Q+3]$	...	$x[(P-1)Q+Q]$

Cuadro 3.1: Segmentación del vector x

Si llamamos a esta matriz m y a los índices que cuentan los segmentos y los datos como i y j respectivamente. Podemos definir la siguiente ecuación que relaciona los valores del vector x con la matriz.

$$m[i,j] = x[(i-1)Q + j], \text{ donde } \left\{ \begin{array}{c} i = 1, 2, 3, \ldots, P \\ j = 1, 2, 3, ..., Q \end{array} \right\} \tag{3.25}$$

El tamaño de los segmentos dependerá de los requisitos que se precisen para la aplicación. La forma más sencilla de realizar la división es partiendo el vector de la

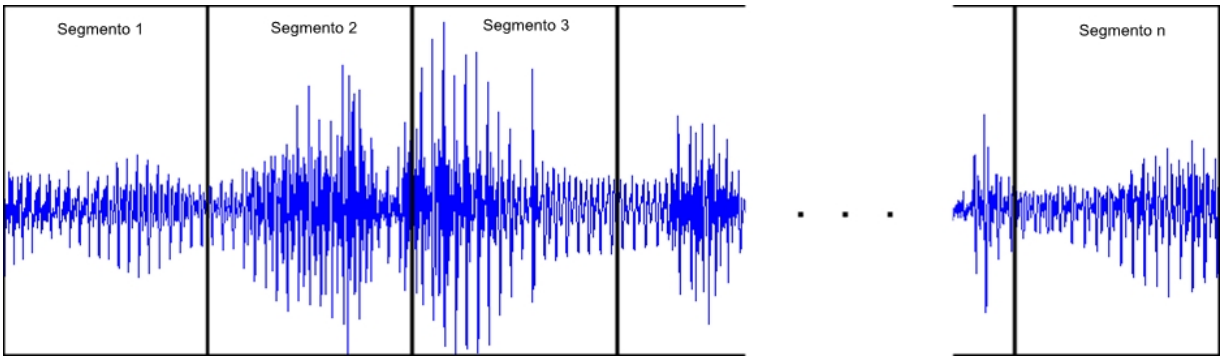


Figura 3.5: Segmentación de una señal en partes iguales

señal en fragmentos del mismo tamaño como se ejemplifica en la figura 3.5. Aquí se puede observar una señal cualquiera segmentada en P partes.

Con el objetivo de generar un algoritmo computacional que sea capaz de segmentar una señal cualquiera en partes iguales, se emplea la ecuación 3.25 para obtener los valores iniciales y finales de cada fracción. Es decir, los casos para cada i cuando $j = 1$ y $j = Q$. Para este fin se consideran los casos que a continuación se muestran.

$$(P-1)Q + Q = PQ = N \tag{3.26}$$

El desarrollo antes mencionado se presenta en el siguiente cuadro.

Segmento	Inicio	Fin
1	1	Q
2	Q+1	2Q
3	2Q+1	3Q
4	3Q+1	4Q
P	(P-1)Q+1	PQ

Cuadro 3.2: Segmentación de una señal

Los índices mostrados en la cuadro 3.2 nos ayudan a comprender las partes exactas en las que la señal original será cortada para generar la matriz. Dependiendo de las formas y lenguajes de programación, se puede implementar de distintas formas esta segmentación. La primera forma consiste en realizar dos ciclos, ligados a los dos índices mostrados (i, j), lo cuales se encargarán de copiar cada uno de los datos desde el vector hasta la matriz. Para llevar a cabo este se requiere de la implementación directa de la Ecuación 3.25. El segundo método plantea que se pueden copiar fracciones enteras de un vector dado el primer y último valor, para lo cual podemos implementar únicamente los índices de la cuadro 3.2. En el diagrama de la figura 3.6 se pueden apreciar los dos métodos sugeridos para realizar la copia del vector a la matriz.

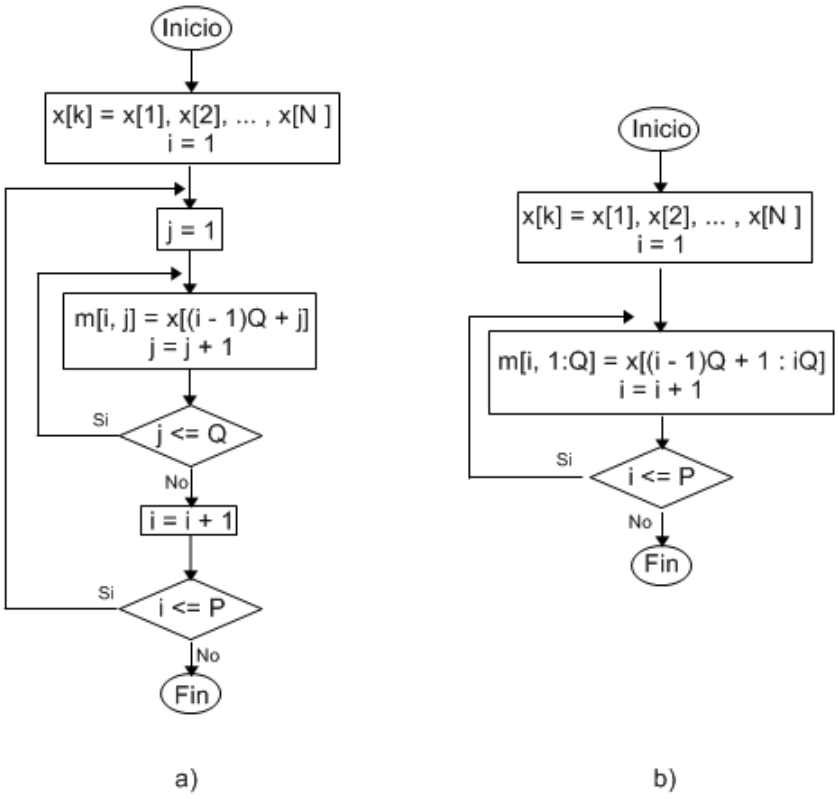


Figura 3.6: Algoritmos para segmentar un vector

En el caso a) presentado en el diagrama de la figura 3.6 se debe de realizar una resta, una suma y una multiplicación cada vez que se copia un dato. Además del incremento de los contadores i y j, lo cuales se pueden considerar como sumas. En total se tienen que realizar:

$$SumasTotales : 2PQ + P$$
$$RestasTotales : PQ$$
$$MultiplicacionesTotales : PQ$$

Para el segundo caso, se requiere realizar una resta, dos multiplicaciones y una suma cada vez que se realiza una copia. Además del incremento del contador i. En total se tendrían:

$$SumasTotales : 2P$$
$$RestasTotales : P$$
$$MultiplicacionesTotales : 2P$$

Dependiendo de las facilidades que se posean, podrían implementarse cualquiera de los algoritmos.

3.5.1. Solapamiento en los segmentos

Como consecuencia de realizar cortes consecutivos en la señal se tiene el riesgo de discriminar información relevante para cada segmento. Esto podría provocar, en algunos casos, que los análisis sean erróneos o poco fiables por falta de información. Uno de los métodos más usados para solventar este problema consiste en agregar información extra a cada sección. Esta información es obtenida de los segmentos consecutivos, por lo que se estaría repitiendo en dos de las secciones. En la figura 3.7 se puede observar un ejemplo de la división de la señal x en distintos segmentos junto con la información adjuntada (traslape o solapamiento).

Podemos observar en el ejemplo de la figura 3.7 que cada segmento posee en cada extremo un conjunto de datos, el solapamiento. Tanto para la información principal como para la información de traslape podemos definir índices que nos ayudaran a comprender los patrones a seguir para el desarrollo de un algoritmo que realice esta segmentación.

Primeramente debemos de considerar las siguientes constantes:

$$P' \rightarrow \text{Número de segmentos}$$
$$Q' \rightarrow \text{Datos por segmento sin repetir}$$
$$M' \rightarrow \text{Datos de cada traslape}$$

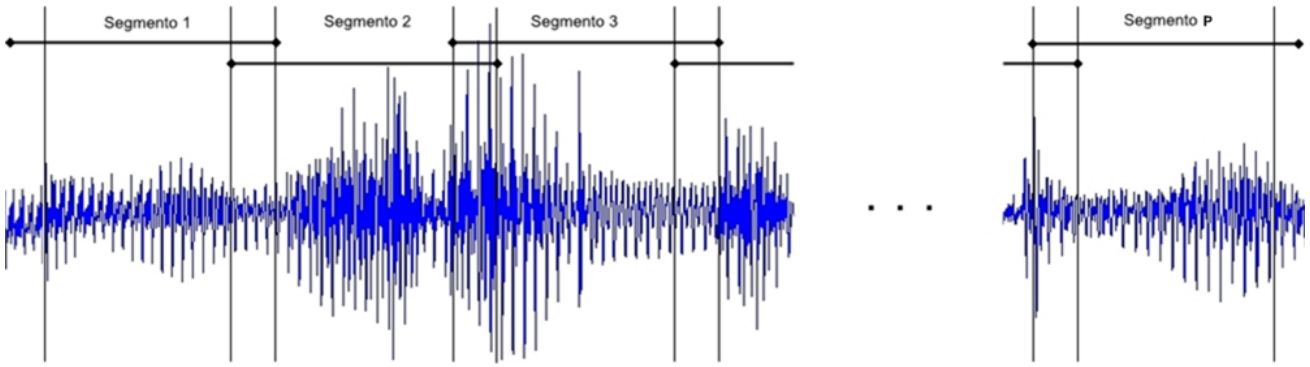


Figura 3.7: Segmentación de la señal con solapamiento en los extremos

Por lo tanto, podemos afirmar que cada segmento está constituido por dos traslapes además de la información Q'. Es decir, el número de datos por cada segmento se obtiene como a continuación se muestra

$$\text{Datos por segmetno} = 2M' + Q'$$

Asimismo, se puede relacionar el tamaño total de la señal x con los parámetros antes mencionados por medio de la siguiente igualdad

$$N = P'Q' + P'M' + M'$$

Podemos deducir esta igualdad al plantear que se tiene Q' datos por cada sección, es decir $P'Q'$ datos. Asimismo se sabe que se tienen dos traslapes por segmento, sin embargo al repetirse en las secciones consecutivas se puede realizar la consideración de que por cada división se tiene un solo solapamiento, concluyendo en que habría $P'M'$ datos de solapamiento. Pero esta última consideración no aplica al último segmento debido a que no tiene otro con el cual realizarla, lo que nos deja con el último valor de la igualdad, M'.

Asimilando el desarrollo realizado en la cuadro 5.1, podemos generalizar la división como se presenta en la cuadro 3.3.

Podemos generalizar para cualquier valor de la matriz (m'), empleando los índices i y j como se muestra en la Ecuación 3.27.

$$m'[i,j] = x[(i-1)(M'+Q')+j] \text{ donde } \left\{ \begin{array}{c} i = 1,2,3,\ldots,P' \\ j = 1,2,3,...,(2M'+Q') \end{array} \right\} \qquad (3.27)$$

Con el fin de simplificar la ecuación 3.27 podemos realizar un cambio de variable en el que se consideraría que la información de sobrepaso de cada segmento está contenida

		Segmento			
		1	**2**	...	***P'***
Datos	**1**	$x[1]$	$x[M'+Q'+1]$	...	$x[(P'-1)(M'+Q')+1]$
	2	$x[2]$	$x[M'+Q'+2]$	...	$x[(P'-1)(M'+Q')+2]$
	...	...	...	...	...
	M'	$x[M']$	$x[M'+Q'+M']$	...	$x[(P'-1)(M'+Q')+M']$
	...	...	...	...	...
	M' + Q'	$x[M'+Q']$	$x[2M'+2Q']$	...	$x[P'M'+P'Q']$
	...	...	...	...	...
	2M' + Q'	$x[2M'+Q']$	$x[3M'+2Q']$	...	$x[(P'+1)M'+P'Q']$

Cuadro 3.3: Segmentación de una señal x con solapamiento

en un solo conjunto en uno de los extremos. En la figura 3.8 podemos ver un ejemplo de esta consideración. Las constantes e igualdades resultantes para este caso se expresarían de la siguiente forma.

$$\text{Número de segmentos} \rightarrow P = P'$$
$$\text{Datos principales por segmento} \rightarrow Q = Q' + M'$$
$$\text{Datos de cada traslape} \rightarrow M = M'$$
$$\text{Datos por segmetno} = M + Q$$
$$N = PQ + M$$

Al aplicar el cambio de variable a la ecuación 3.27 podemos obtener la siguiente ecuación

$$m[i,j] = x[(i-1)Q + j] \text{ donde } \left\{ \begin{array}{c} i = 1,2,3,\ldots,P \\ j = 1,2,3,...,Q+M \end{array} \right\} \tag{3.28}$$

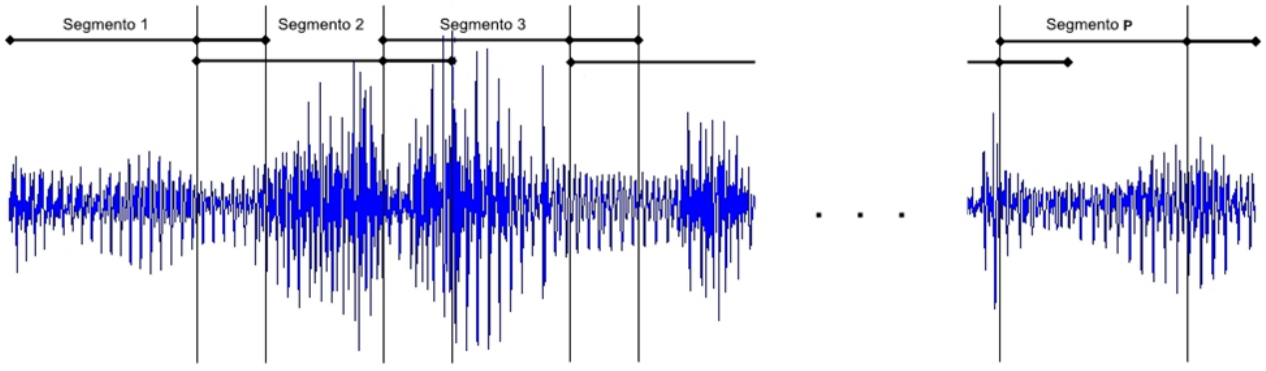


Figura 3.8: Segmentación de la señal con un solo solapamiento

Implementando la ecuación 3.28 podemos obtener los índices de inicio y fin de cada segmento como se muestra en la cuadro 3.4.

Segmento	Inicio del segmento	Fin del segmento
1	1	Q+M
2	Q+1	2Q+M
3	2Q+1	3Q+M
P	(P-1)Q+1	PQ+M

Cuadro 3.4: Índices de inicio y fin de los segmentos con solapamiento

La ventaja del cambio de variable se puede observar en la cuadro 3.3, debido a que se puede interpretar de forma más fácil que si se hubiera realizado con la ecuación 3.27. Por ejemplo, para el segundo segmento, empleando la ecuación 2, se tendría el índice final

$$(2-1)(M'+Q')+2M'+Q = M'+Q'+2M'+Q = 3M'+2Q'$$

Mientras que para la ecuación 3 se tendría

$$(2-1)Q+M+Q = Q+M+Q = 2Q+M$$

Siendo más fácil de calcular el segundo.

Al igual que en el caso de segmentación sin solapamiento, este puede ser implementado mediante dos algoritmos. Considerando la copia dato por dato y la copia de conjunto de datos según la cuadro 3.4.

Sin embargo, los casos mencionados solo aplican cuando la segmentación realizada fracciona al vector x en partes iguales. En dado caso de que al realizar la segmentación se requieran más valores de los que se poseen se considera que estos serán cero. Es decir, para el caso en que $(i-1)Q+j > N$ se designa que $m[i,j] = 0$.

3.5.2. Tipos de ventanas

Al realizar el análisis de la señal segmentada, se considera que cada fracción es una señal independiente del resto. Para todos los valores de tiempo antes y después de la sección las magnitudes son consideradas como cero, como se ejemplifica en la figura 3.10. A consecuencia de esta última consideración, se puede presentar el caso en el que haya un cambio brusco de magnitudes, como se observa en la figura 3.11. Este comportamiento podría provocar errores en los análisis. Para solventar este tipo de fenómenos se suelen aplicar algoritmos que suavizarían la transición. A estos algoritmos se les conoce como ventanas y dependiendo de la aplicación pueden variar su forma.

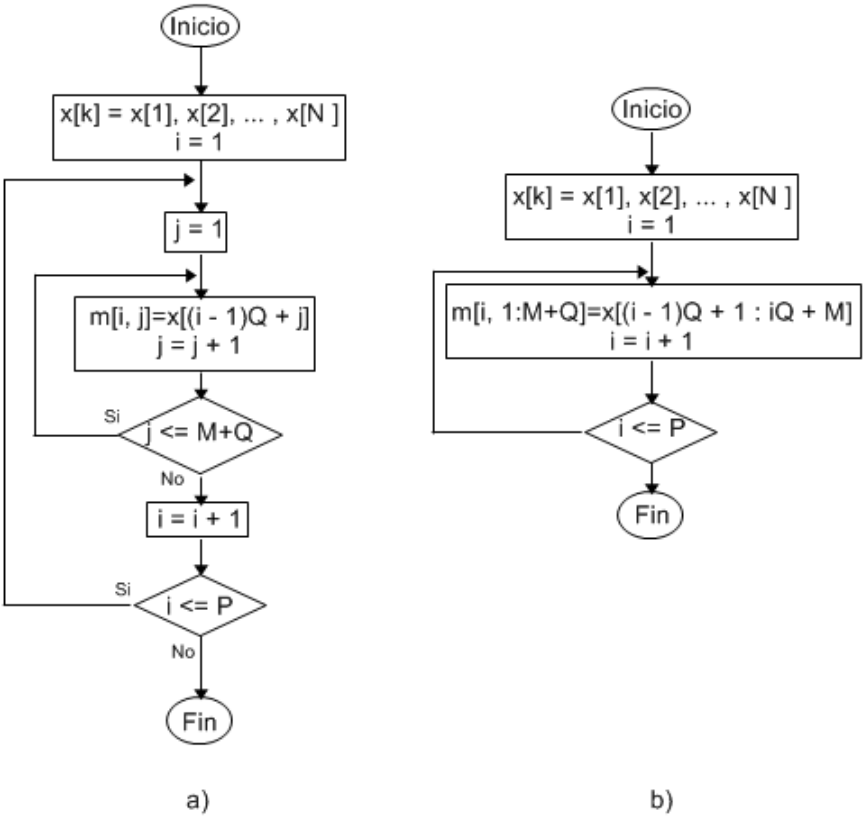


Figura 3.9: Algoritmos para segmentar un vector con solapamiento

Existen distintos tipos de ventana que pueden ser aplicadas a los segmentos de la señal. Cada una de ellas tiene un comportamiento en el tiempo y en la frecuencia que podrían dañar o favorecer nuestro análisis dependiendo de los algoritmos que se usen. De entre las ventanas que se pueden usar se encuentra la rectangular, la triangular, de Hanning y la de Hamming. En la figura 3.12 podemos observar la forma de las cuatro ventanas, mientras que en la figura 3.13 podemos observar su comportamiento en el dominio de la frecuencia y en la figura 3.14 se observa su comportamiento referido a la potencia en decibeles.

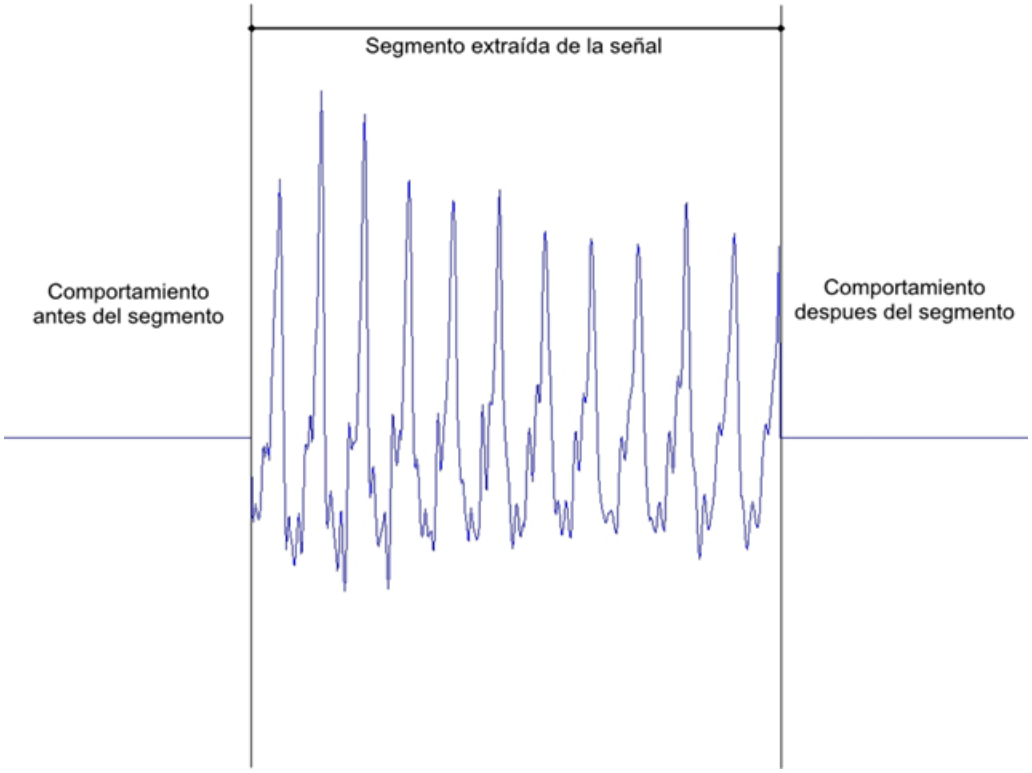


Figura 3.10: Comportamiento supuesto del segmento antes y después de los valores extraídos de la señal

La ventana ideal para el procesamiento de señales estaría descrita por una constante, es decir, que la ventana estaría compuesta por un número infinito de datos con la misma magnitud. En el dominio de la frecuencia esta ventana equivaldría a un impulso. Sin embargo, al no poder disponer de una cantidad infinita de datos se suele optar por otra ventana que asemeje el comportamiento de la ideal.

El segmentar una señal es equivalente a aplicar una ventana rectangular sobre la señal original. La ventana rectangular no modifica la señal en el dominio del tiempo de los datos considerados, sin embargo tampoco resuelve el problema de transición en los extremos. Al observar su comportamiento en frecuencia podremos observar que esta ventana es la más estrecha de las cuatro presentadas, asemejando a la ideal. Como consecuencia de que su lóbulo central sea estrecho, sus lóbulos secundarios tienen amplitudes grandes, además de ser la ventana con la caída en potencia más baja de todas

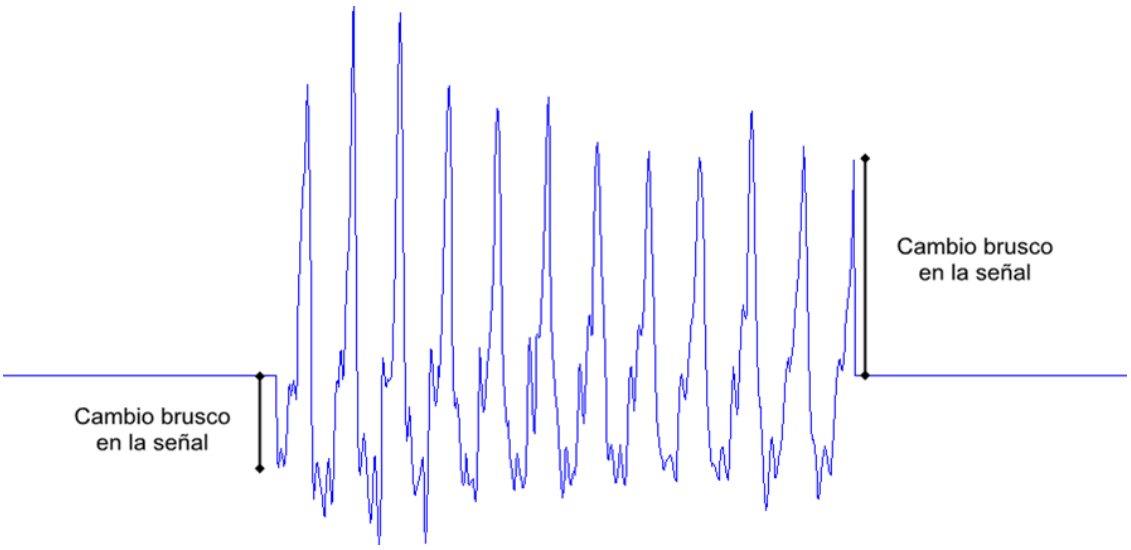


Figura 3.11: Cambios bruscos en la señal como consecuencia de la segmentación de la señal

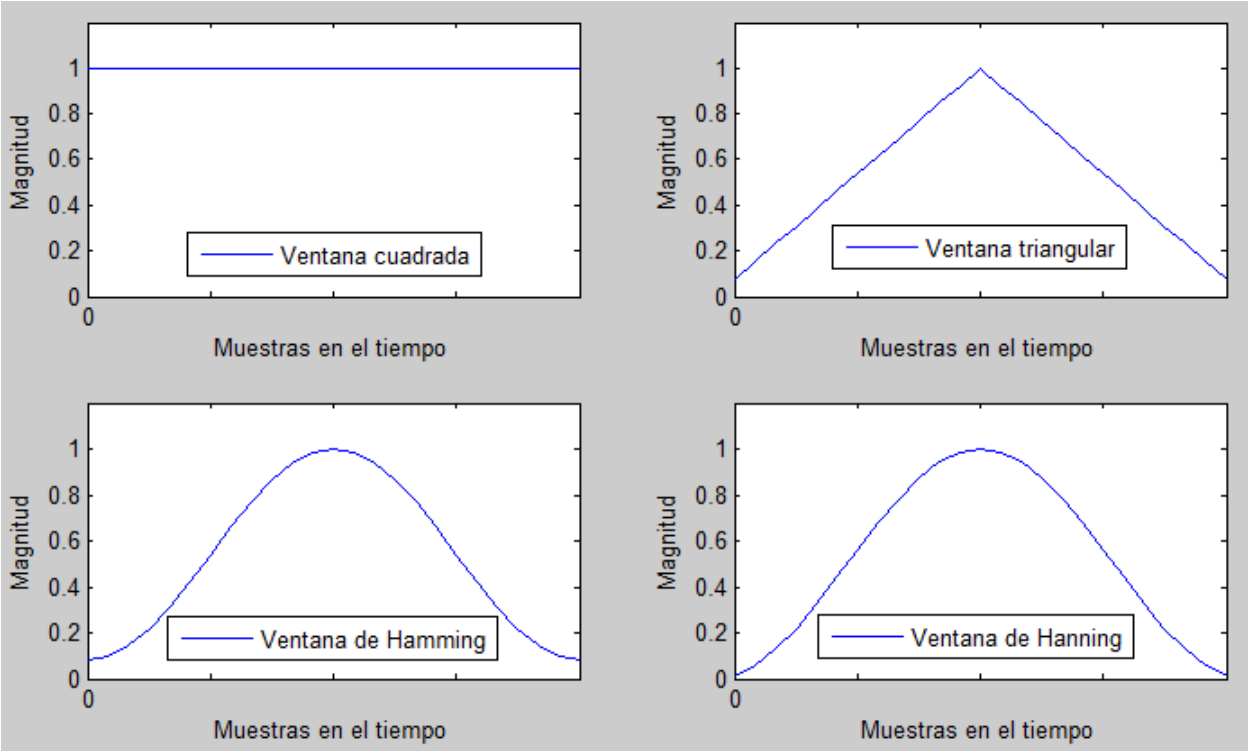


Figura 3.12: Representación en el tiempo de las ventanas cuadrada, triangular, de Hamming y de Hanning

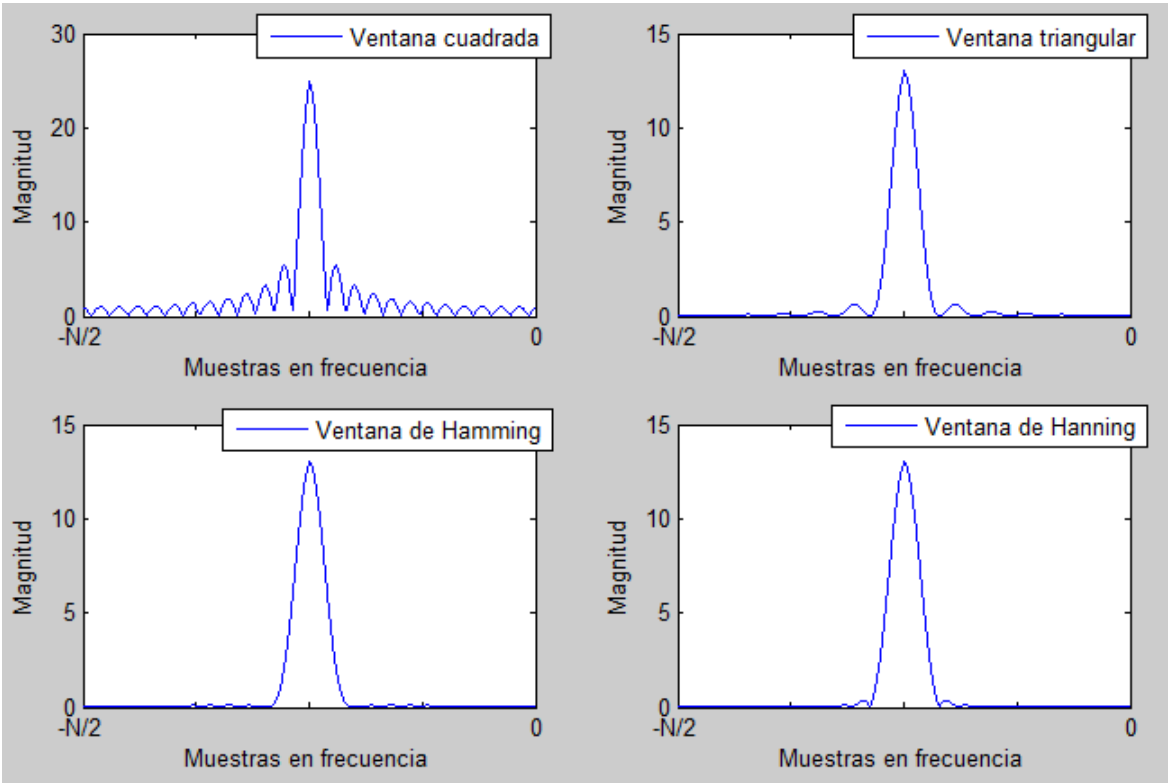


Figura 3.13: Representación en frecuencia de las ventanas cuadrada, triangular, de Hamming y de Hanning

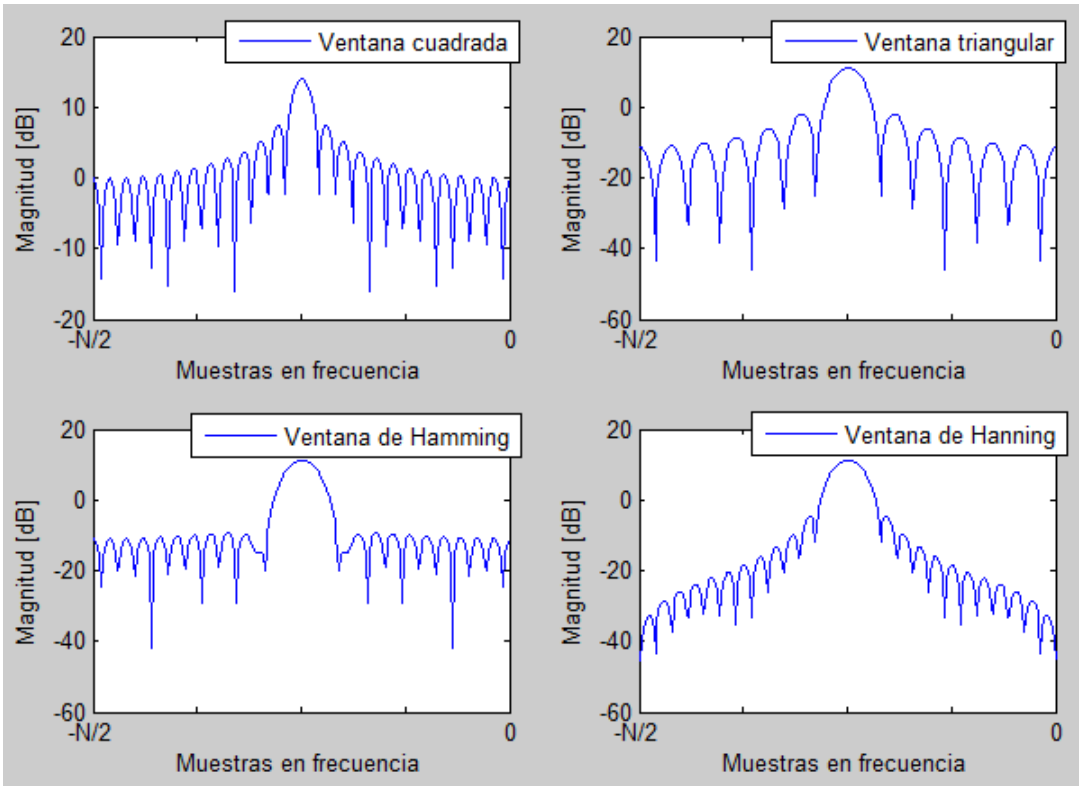


Figura 3.14: Representación en potencia (dB) de las ventanas cuadrada, triangular, de Hamming y de Hanning

entre el lóbulo central y su primer lóbulo segundario, al solo tener una caída -6dB.

La ventana triangular, la de Hanning y la de Hamming solventan el problema de transición en los extremos del segmento. Cada una de las ventanas se genera a partir de una ecuación y tienen comportamientos diferentes tanto en el dominio del tiempo como en el dominio de la frecuencia.

La Ventana triangular se puede generar de distintas formas, algunas con un mayor o menor coste computacional. La forma directa de calcularlo es mediante la ecuación 3.29.

$$x(n) = \left\{ \begin{array}{ccc} \frac{N-1}{2} - \left| n - \frac{N-1}{2} \right| & \forall & 0 \leq n \geq N-1 \\ 0 & , & \text{cualquier otro caso} \end{array} \right\} \tag{3.29}$$

A diferencia de la ventana rectangular, está ventana tiene su lóbulo central más ancho, lo que causa que sus lóbulos consecutivos tengan menor amplitud. Mientras tanto, la caída en potencia del lóbulo central al siguiente consecutivo es de aproximadamente -13 dB. Esto último confirma que las perturbaciones son menores.

La ventana de Hanning está definida por la ecuación 3.30.

$$x(n) = \left\{ \begin{array}{ccc} \frac{1}{2}\left(1 - cos\frac{2\pi n}{N-1}\right) & \forall & 0 \leq n \geq N-1 \\ 0 & , & \text{cualquier otro caso} \end{array} \right\} \tag{3.30}$$

Esta Ventana actúa de forma similar que la triangular. Su lóbulo central es casi del mismo tamaño que el de la triangular, con la diferencia de que los lóbulos consecutivos son más pequeños. La decaída en potencia del primer lóbulo consecutivo al central es de aproximadamente -15 dB.

La ventana de Hamming modifica los coeficientes de la ecuación de Hanning para reducir la amplitud de los lóbulos consecutivos al central. La modificación se presenta en la ecuación 3.31

$$x(n) = \left\{ \begin{array}{ccc} \frac{1}{2}\left(0{,}54 - 0{,}46cos\frac{2\pi n}{N-1}\right) & \forall & 0 \leq n \geq N-1 \\ 0 & , & \text{cualquier otro caso} \end{array} \right\} \tag{3.31}$$

El tamaño del lóbulo central es casi del mismo tamaño que la de la ventana de Hanning, pero los lóbulos consecutivos son mucho más pequeños. La decaída del primer lóbulo consecutivo es de aproximadamente -19 dB.

3.5.3. Análisis de la implementación de las ventanas

Cada una de las ventanas tiene efectos diferentes en la señal que se implementa. El caso que se analizará a continuación se procura que el único lóbulo presente, al usar ventanas, sea el central. De esta forma se pretende minimizar las interferencias de baja potencia y tratar de priorizar la frecuencia deseada.

Al realizar la implementación de la ventana sobre el segmento de datos de la señal, en el tiempo corresponde a realizar la multiplicación de ambas señales. Mientras que en el dominio de la frecuencia la operación que se realiza corresponde a una convolución. A continuación se plantea las ecuaciones que representan estás operaciones.

Sea $x(t)$ la señal y $v(t)$ la ventana en el tiempo. Donde:

$$X = X(w) = F[x(t)]$$
$$V = V(w) = F[v(t)]$$

Donde $F[]$ corresponde a la transformada de Fourier de la función $x()$

La implementación de la ventana en el tiempo esta dada por

$$y(t) = x(t)v(t)$$

De forma similar, la implementación de la ventana en frecuencia se da por

$$Y = X * V$$

Esta operación puede ser ejemplificada de forma gráfica al aplicar una ventana cuadrada sobre una señal senoidal de frecuencia arbitraria, como se muestra a continuación.

Como se puede observar en la figura 3.15 al aplicar la ventana cuadrada sobre la señal, en el dominio de la frecuencia, la función de la ventana se centra sobre cada impulso de la señal original; esto como consecuencia de la convolución. En este ejemplo, el lóbulo central se aprecia claramente, permitiéndonos ignorar el resto de lóbulos. Sin embargo, en dado caso de que aparecieran más frecuencias en nuestra señal, los lóbulos de menor tamaño se podrían sumar junto con otros llegando a generar lóbulos lo suficientemente grandes para crear información errónea.

En la figura 3.16 se pude observar una señal con tres frecuencias diferentes, donde dos de ellas se encuentran próximas. En la señal resultante de la convolución se pueden observar lóbulos de mayor tamaño que los presentados en el mismo caso de la figura 3.16, los cuales representarían errores en nuestra señal generados por el tipo de ventana implementada. Adicionalmente, a pesar de que todos los lóbulos centrales deberían de

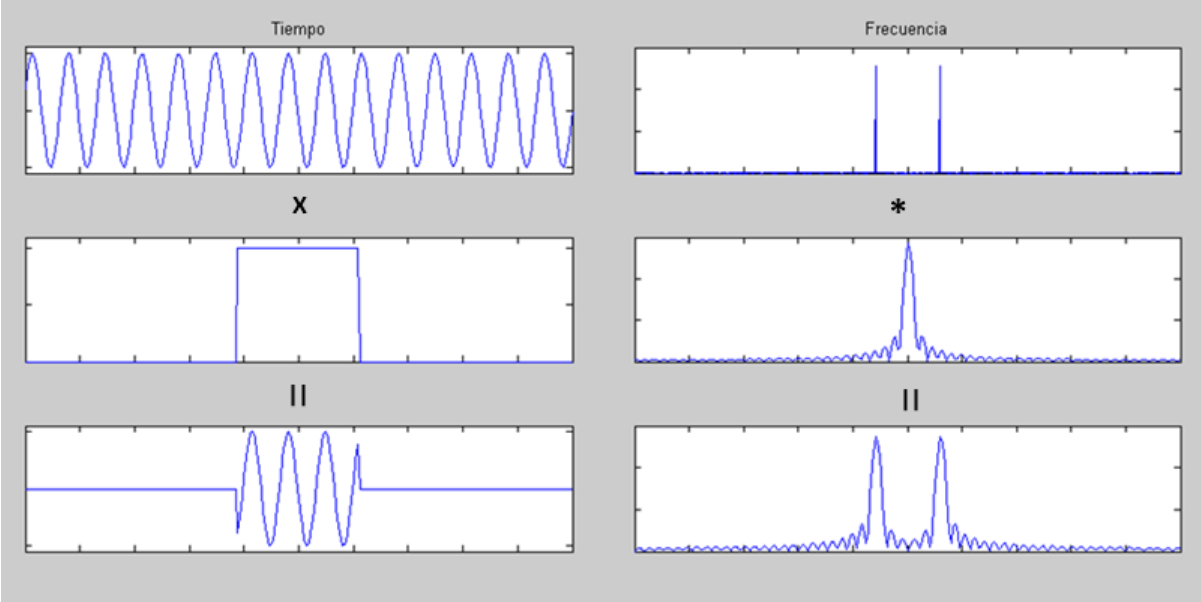


Figura 3.15: Ejemplo de aplicación de una ventana rectangular sobre una señal con una única frecuencia. Donde (x) significa multiplicación y (*) convolución

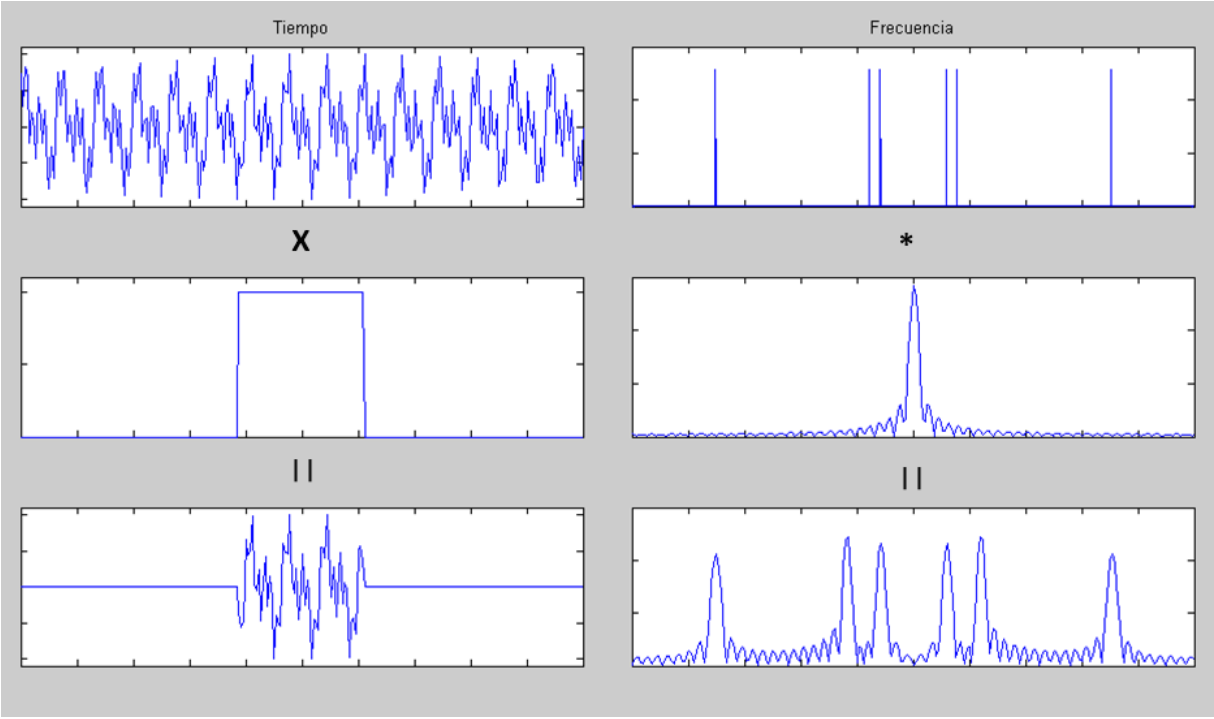


Figura 3.16: Ejemplo de aplicación de ventana rectangular sobre una señal con tres frecuencias. Donde (x) significa multiplicación y (*) convolución

ser del mismo tamaño se puede distinguir un ligero cambio de amplitud en las dos frecuencias más próximas. Esto último a causa de que se les suman lóbulos secundarios próximos, causando modificaciones de la información de relevancia de la señal.

Para estudiar la forma en que actúan sobre una señal la ventana rectangular, triangular, de Hanning y de Hamming, se propone el uso de la función de diente de sierra como la señal a analizar. Esta al ser llevada al dominio de la frecuencia se convierte en un conjunto de impulsos decrecientes, lo cual facilitará observar el comportamiento de las ventanas en el dominio de la frecuencia.

Como se ha mencionado con antelación, la ventana ideal correspondería a una señal constante para cualquier valor de tiempo. Sin embargo, para poder implementar esta ventana se requeriría un número infinito de valores o en el caso computacional un número considerablemente alto de los mismos. Una aproximación de este caso de ventana se aprecia en la figura 3.17. Se puede observar que cada una de las frecuencias presentadas está representada por una señal que casi aproxima a un impulso. Para obtener una mejor representación, bastaría con aumentar el número de valores de la señal de diente de sierra, lo que conllevaría un mayor coste de cálculo.

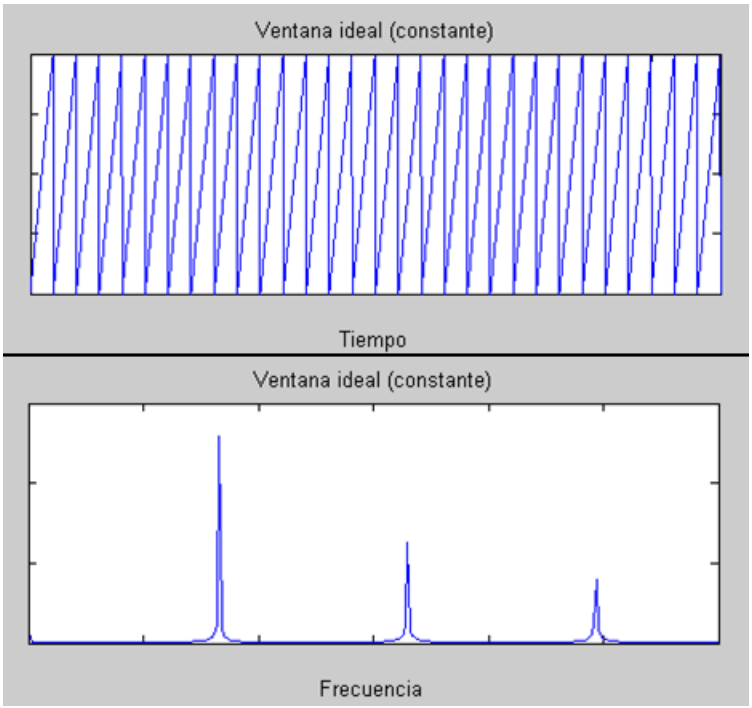


Figura 3.17: Aproximación de una ventana ideal aplicada a un diente de sierra

Se ha de considerar la señal presentada en la figura 3.17 como la señal a analizar. De ésta, se extraerá una muestra a la que se le aplicará la ventana y se observará su comportamiento en frecuencia.

Al aplicar la ventana rectangular, se puede observar que en el dominio de la frecuencia existen junto a los lóbulos centrales un conjunto de lóbulos secundarios, como

se muestra en la figura 3.18. Estos lóbulos secundarios, a pesar de ser de menor tamaño que el central, causan errores en la señal. Un ejemplo de estos errores se presenta al comparar los comportamientos en frecuencia de las figuras 3.17 y 3.18, en las cuales se observa que para el caso de la ventana rectangular el transitorio entre cada frecuencia se mantiene separado de cero. Esta separación es interpretada como la presencia de todas las frecuencias de este intervalo en la señal, que no deberían de existir.

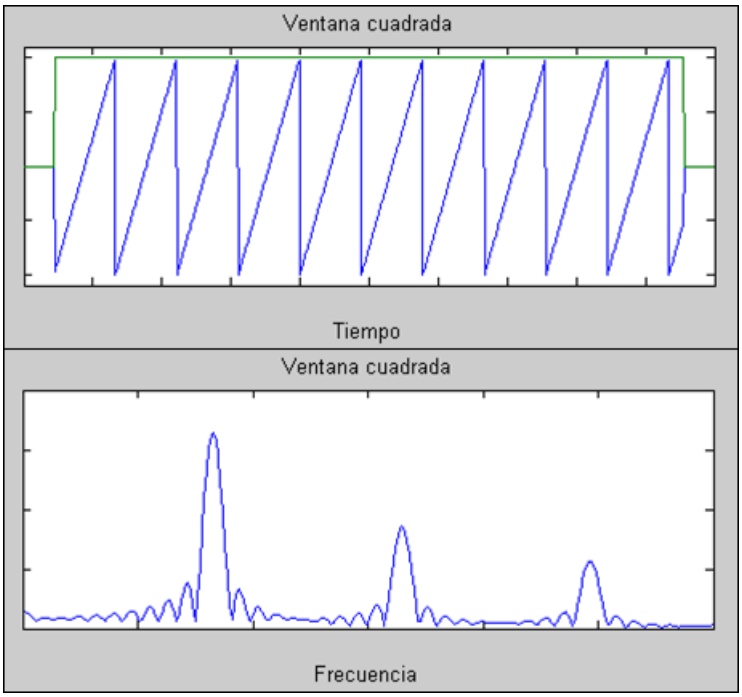


Figura 3.18: Comportamiento de una ventana rectangular aplicada sobre una función diente de sierra

Al aplicar la ventana triangular sobre la muestra de la señal de diente de sierra se solventa en el problema de transición en los extremos del conjunto de datos en el dominio del tiempo, como se observa en la figura 3.19. Asimismo, en el dominio de la frecuencia se puede observar la disminución de la amplitud y cantidad de lóbulos secundarios y el aumento de la anchura de los lóbulos centrales, en comparación con la ventana cuadrada. Idealmente deberían de ser impulsos los lóbulos centrales, por lo que es poco conveniente que su grosor incremente.

La ventana de Hanning, al igual que la triangular, resuelve el problema de transición, como se observa en la figura 3.20. Asimismo, en la representación en frecuencia, los lóbulos secundarios son pocos y con una amplitud considerablemente baja. Sin embargo, la anchura de los lóbulos centrales sigue siendo mayor a comparación de la ventana cuadrada.

De las ventanas analizadas, la ventana de Hamming tiene el mejor comportamiento respecto a los lóbulos secundarios. Desafortunadamente, esta ventana no resuelve el problema de la anchura de los lóbulos centrales.

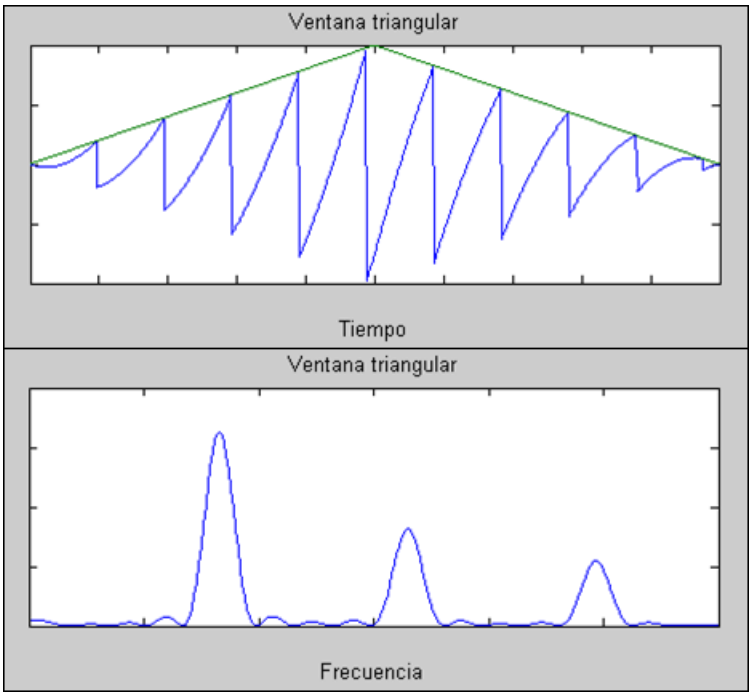


Figura 3.19: Comportamiento de una ventana triangular aplicada sobre una función diente de sierra

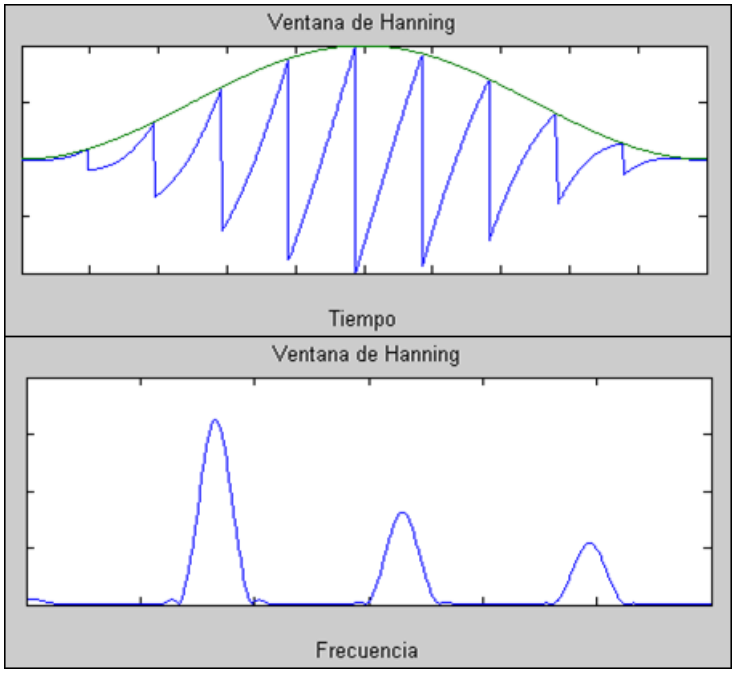


Figura 3.20: Comportamiento de una ventana de Hanning aplicada sobre una función de diente de sierra

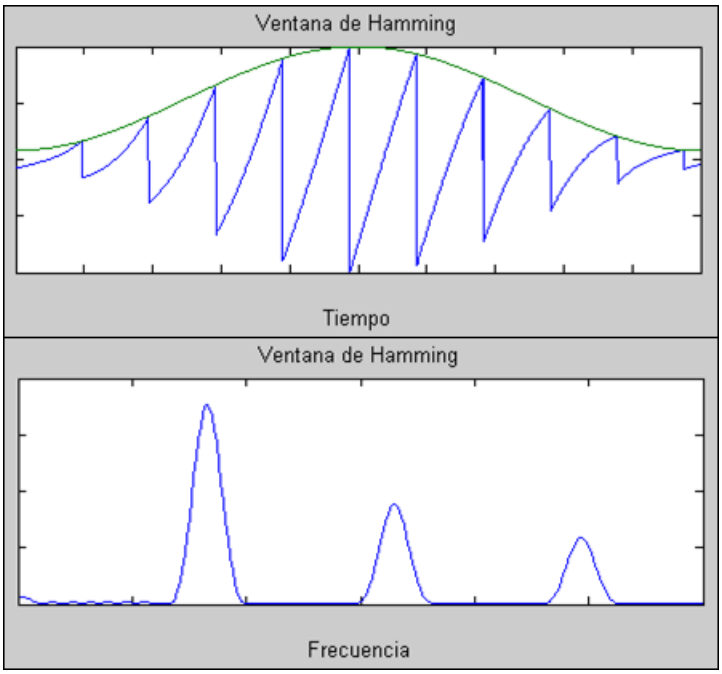


Figura 3.21: Comportamiento de una ventana aplicada sobre una función de diente de sierra

3.6. Estimación de parámetros LPC

Los coeficientes de predicción lineal (LPC por sus siglas en inglés Linear Prediction Coefficients) corresponden a un conjunto de parámetros extraídos de una señal y que representan el comportamiento de la misma. El método se basa en la extracción de los coeficientes que componen a la función de transferencia de la señal. Mediante la aplicación de un filtro caracterizado por estos valores, es posible reconstruir dicha información, siempre que la señal de excitación y la ganancia asociada sean las adecuadas.

Para señales de voz, el cálculo de los parámetros LPC parte del sistema que modela el tracto vocal, en el dominio de la frecuencia compleja z, representado en la ecuación 3.32. Este consiste en un sistema todo polos con N coeficientes y una ganancia dada.

$$H[z] = \frac{s[z]}{u[z]} = \frac{G}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{3.32}$$

Donde $s[z]$ corresponde a la señal de salida o deseada, $u[z]$ representa la señal de entrada y a_k son los parámetros representativos del sistema a modelar. Empleando el modelo de ecuaciones en diferencias, definido en la ecuación 2.12, se puede representar, en el dominio del timepo, el sistema del tracto vocal de la siguiente forma

$$s_n = Gu_n - \sum_{k=1}^{P} a_k s_{fn-k} \tag{3.33}$$

En esta ecuación la señal a generarse depende tanto del valor presente de la entrada como de los valores pasados de la salida. A partir de la ecuación 2.12 se puede extraer la señal de estimación, la cual queda representada como a continuación

$$\hat{s}_n = -\sum_{k=1}^{P} a_k s_{n-k} \tag{3.34}$$

La estimación de los parámetros se basa en provocar que el error de predicción sea el más bajo. La función del error queda expresada mediante la siguiente expreción

$$e[n] = s[n] - \hat{s}[n] \tag{3.35}$$

Sustituyendo la ecuación 3.34 en la ecuación 3.35 se obtiene que

$$e[n] = s[n] + \sum_{k=1}^{P} a_k s_{n-k} \tag{3.36}$$

Donde P corresponde al total de valores de a_k.

La suma que se puede observar en la ecuación 3.36 puede ser representada mediante una operación matricial en la que se involucren todos lo parámetros de las dos variables (a y s)

$$A_p^T = [a_1, a_2, a_3, ..., a_p] \tag{3.37}$$

$$S_p^T = [s_{n-1}, s_{n-2}, s_{n-3}, ..., s_{n-p}] \tag{3.38}$$

A partir de estos vectores se puede representar la ecuación 3.36 como se observa en la ecuación 3.39

$$e[n] = s[n] + A_P^T S_P \tag{3.39}$$

A partir de la ecuación 3.39 se calcula el EMC

$$E\{e^2(n)\} = E((s[n] + A_P^T S_P)(s[n] + A_P^T S_P)) \tag{3.40}$$

$$E\{e^2(n)\} = E(s^2[n] + 2s[n]A_P^T S_P + A_P^T S_P S_P^T A_P) \tag{3.41}$$

A partir de la función de auto-correlación 3.20 se puede reescribir la ecuación de la siguiente forma

$$E\{e^2(n)\} = E(r_P(0) + 2A_P^T r_P + A_P^T R_P A_P) \tag{3.42}$$

Donde r_P es un vector de auto-correlación y R_P una matriz de auto-correlación simétrica de Toeplitz. Ambos son obtenidos de la señal $s(n)$.

"Aplicando el criterio de optimización sobre $E\{e^2(n)\}$, es decir, derivando la esperanza matemática del error cuadrático respecto a A_P" [16]

$$\begin{aligned} \frac{\partial E\{e^2(n)\}}{\partial A_P} &= 0 + 2r_P + 2R_P A_P = 0 \\ R_P A_P &= -r_P \end{aligned} \tag{3.43}$$

"la ecuación se denomina ecuación de Wiener-Hopf" [16]. A partir de esta es posible calcular los parámetros del vector A mediante el despeje representado en 3.44.

$$A_P = -R_P^{-1} r_P \tag{3.44}$$

La representación matricial de la ecuación 3.43 queda establecida como se observa en la ecuación 3.45.

$$\begin{bmatrix} r[0] & r[1] & r[2] & ... & r[p-1] \\ r[1] & r[0] & r[1] & ... & r[p-2] \\ r[2] & r[1] & r[0] & ... & r[p-3] \\ ... & ... & ... & \ddots & ... \\ r[p-1] & r[p-2] & r[p-3] & ... & r[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} -r[1] \\ -r[2] \\ -r[3] \\ \vdots \\ -r[P] \end{bmatrix} \tag{3.45}$$

Por lo tanto, el cálculo de los parámetros del vector A_P pueden obtenerse a partir de la ecuación matricial 3.45.

3.6.1. Cálculo de los parámetros LPC

El cálculo de los parámetros LPC se basa en la ecuación 3.43. Con el fin de realizar los cálculos de una forma más dinámica se puede reescribir como se muestra en la ecuación 3.46.

$$RA_3 = -r \tag{3.46}$$

Para este caso se considerará un conjunto de parámetros A de tamaño 4 ($n = 4$). Por lo tanto, los vectores y matrices que se emplearán quedan definidos de la sigueinte forma

$$R = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & 1 & \rho_1 & \rho_2 \\ \rho_2 & \rho_1 & 1 & \rho_1 \\ \rho_3 & \rho_2 & \rho_1 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} a_3 1 \\ a_3 2 \\ a_3 3 \\ a_3 4 \end{bmatrix}$$

$$r = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix}$$

Por lo tanto, la forma matricial de la ecuación 3.46 es la siguiente

$$\begin{bmatrix} 1 & \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & 1 & \rho_1 & \rho_2 \\ \rho_2 & \rho_1 & 1 & \rho_1 \\ \rho_3 & \rho_2 & \rho_1 & 1 \end{bmatrix} \begin{bmatrix} a_{31} \\ a_{32} \\ a_{33} \\ a_{34} \end{bmatrix} = - \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \tag{3.47}$$

Está matriz puede ser reescrita mediante un conjunto de sub-matrices de la siguiente forma

$$\begin{bmatrix} R' & \tilde{r}' \\ \tilde{r}'^T & 1 \end{bmatrix} \begin{bmatrix} A'_3 \\ a_{34} \end{bmatrix} = - \begin{bmatrix} r' \\ \rho_4 \end{bmatrix} \tag{3.48}$$

Donde

$$R' = \begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix}$$

$$A'_3 = \begin{bmatrix} a_{31} \\ a_{32} \\ a_{33} \end{bmatrix}$$

$$r' = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix}$$

A partir de la expresión matricial 3.48 se pueden obtener las siguientes ecuaciones

$$R'A'_3 + \tilde{r}'a_{34} = -r' \tag{3.49}$$

$$r'^T A'_3 + a_{34} = -\rho_4 \tag{3.50}$$

De la ecuación 3.49 se depeja la matriz A'_3

$$A'_3 = -R'^{-1}r' - R'^{-1}\tilde{r}'a_{34} \tag{3.51}$$

Con base en ésta se realiza un cambio de variable que asemeja a la ecuación 3.46

$$A_2 = -R'^{-1}r' \tag{3.52}$$

Por lo tanto la ecuación 3.51 queda expresada de la siguiente forma

$$A'_3 = A_2 + \tilde{A}_2 a_{34} \tag{3.53}$$

A partir de la ecuación 3.50 se despeja el parámetro a_{34}, que permanecerá en función de A_2

$$a_{34} = \frac{-\rho_4 - r'^T A_2}{\tilde{r}'^T \tilde{A}_2 + 1} \tag{3.54}$$

Con base en la ecuación 3.52 se define la siguiente función cuya estructura es la misma que 3.46

$$R'A_2 = -r' \tag{3.55}$$

Con

$$A_2 = \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \end{bmatrix}$$

De forma similar que con la ecuación 3.48 se reestructura la forma matricial de 3.55 como a continuación se muestra

$$\begin{bmatrix} R'' & \tilde{r}'' \\ \tilde{r}''^T & 1 \end{bmatrix} \begin{bmatrix} A'_2 \\ a_2 3 \end{bmatrix} = - \begin{bmatrix} r'' \\ \rho_3 \end{bmatrix} \tag{3.56}$$

Donde

$$R'' = \begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix}$$

$$A_2' = \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix}$$

$$r'' = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}$$

Con esta nueva ecuación matricial se realiza el mismo procedimiento que con 3.46. Las expreciones obtenidas de la ecuación matricial son las siguientes

$$R''A_2' + \tilde{r}''a_{23} = -r'' \tag{3.57}$$

$$\tilde{r}''^T A_2' + a_{23} = -\rho_3 \tag{3.58}$$

Al despejar A_2' de 3.57 se tiene que

$$A_2' = -R''^{-1}r'' - R''^{-1}\tilde{r}''a_{23} \tag{3.59}$$

Realizando el cambio de variable

$$A_1 = -R''^{-1}r'' \tag{3.60}$$

Por lo tanto se puede representar la ecuación 3.59 como

$$A_2' = A_1 + \tilde{A}_1 a_{23} \tag{3.61}$$

También se realiza el despeje del parámetro a_{23} a patir de 3.58

$$a_{23} = \frac{-\rho_3 - \tilde{r}''^T A_1}{\tilde{r}''^T \tilde{A}_1 + 1} \tag{3.62}$$

Finalmente se define una nueva función con base en 3.60, como se muestra a continuación

$$R''A_1 = -r'' \tag{3.63}$$

Con

$$A_1 = \begin{bmatrix} a_{11} \\ a_1 2 \end{bmatrix}$$

La última forma matricial que se generará es el siguiente

$$\begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix} = - \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \tag{3.64}$$

De esta se extraen las siguientes ecuaciones, las cuales están compuestas únicamente por valores y no por vectores y matrices como sus similares anteriores.

$$a_{11} + \rho_1 a_1 2 = -\rho_1 \tag{3.65}$$

$$\rho_1 a_1 1 + a_1 2 = -\rho_2 \tag{3.66}$$

Por lo tanto, para calcular los parámetros a_{12} y a_{11} se emplean las siguientes ecuaciones

$$a_{12} = \frac{-\rho_2 + \rho_1^2}{-\rho_1^2 + 1} \tag{3.67}$$

$$a_{11} = -\rho_1(1 + a_{12}) \tag{3.68}$$

De esta forma, el cálculo de los parámetros de la matriz A_3 permanece en función de los vectores A_2 y A_1. Estos vectores solo pueden ser determinados si se calcula su anterior inmediato. Este cálculo queda expresado en las siguientes ecuaciones.

$$A_1 = \begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix} = \begin{bmatrix} -\rho_1(1 + a_{12}) \\ \frac{-\rho_2 + \rho_1^2}{-\rho_1^2 + 1} \end{bmatrix} \tag{3.69}$$

$$A_2 = \begin{bmatrix} A_2' \\ a_{23} \end{bmatrix} = \begin{bmatrix} A_1 + \tilde{A}_1 a_{23} \\ \frac{-\rho_3 - \tilde{r}''^T A_1}{\tilde{r}''^T \tilde{A}_1 + 1} \end{bmatrix} \tag{3.70}$$

$$A_3 = \begin{bmatrix} A_3' \\ a_{34} \end{bmatrix} = \begin{bmatrix} A_2 + \tilde{A}_2 a_{34} \\ \frac{-\rho_4 - r'^T A_2}{\tilde{r}'^T \tilde{A}_2 + 1} \end{bmatrix} \tag{3.71}$$

Donde

$$r' = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix} \text{ y } r'' = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \tag{3.72}$$

A partir de este ejemplo se puede generalizar la obtención de cualquier matriz después de A_1. Se considera que el vector A_1 siempre será obtenido usando la misma función 3.69, mientras que los siguientes serán una combinación de sus predecesores

inmediatos. Por lo tanto, asemejando el comportamiento de las ecuaciones 3.70 y 3.71, se puede desarrollar la ecuación a partir de la cual es posible calcular los parámetros A_k

$$A_k = \begin{bmatrix} A'_k \\ a_{kp} \end{bmatrix} \tag{3.73}$$

Donde

$$A'_k = A_{k-1} + \tilde{A}_{k-1} a_{kp} \tag{3.74}$$

$$a_{kp} = \frac{-\rho_{k+1} \tilde{r}^T(1:k) A_{k-1}}{\tilde{r}^T(1:k) \tilde{A}_{k-1} + 1} \tag{3.75}$$

$$k = 2...n-1$$

$$p = k+1$$

Donde n es la cantidad de parámetros que se están calculando

Capítulo 4

Implementación del sintetizador por el método de auto-correlación

Un sintetizador es un sistema que trata de reproducir el comportamiento del sistema vocal para generar voz. En la figura 4.1 se observan los distintos componentes que lo integran, éstos están basados en la estructura biológica del tracto vocal. Cada componente se encarga de caracterizar la señal de sonido de diferentes formas. El funcionamiento de este sistema consiste en hacer pasar una señal que ha sido previamente adecuada con su correspondiente ganancia o energía, por un filtro que le proporcionará las características que definan los parámetros LPC asignados. Finalmente se aplicará un filtro que atenuará las frecuencias altas, resaltando las frecuencias bajas donde se encuentran las diferentes frecuencias fundamentales de la voz [10].

Como se puede observar en la figura 4.1, se requiere de una señal base para realizar la sintesis. Está puede ser tanto un ruido blanco como un tren de impulsos. La primera opción se emplea para aquellos sonidos que son considerados como no sonoros (aquellos que no tienen pitch), mientras que la segunda para los sonidos sonoros (aquellos que presentan pitch). Al generar la señal de tren de impulsos, esta deberá de tener una frecuencia equivalente al pitch de la señal original.

La señal base se adecuada con la ganancia de la señal original, como se muestra en la figura 4.1. Esto permitirá proporcionar la intensidad que le corresponde a cada una. De esta forma aquellas partes de la señales con energía baja no podrán ser escuchadas.

Para darle forma a la señal base, se le hace pasar por un filtro que le proporcionará las características y la forma adecuada. Este filtro se trata de un sistema de ecuaciones en diferencias que emplea los parámetros LPC extraídos de la señal original. Finalmente se aplica un filtro de des-énfasis para compensar el efecto generado por el filtro de pre-énfasis.

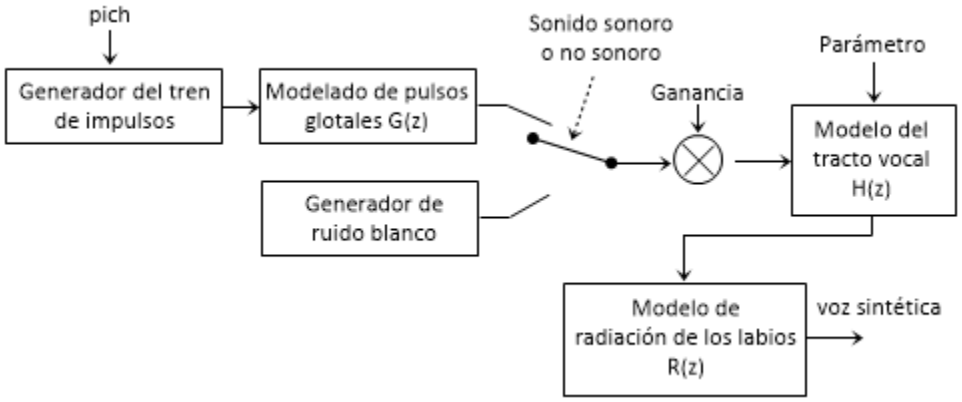


Figura 4.1: Modelo del aparato fonador

4.1. Diseño del sintetizador

Dado que el sintetizador de voz trata de imitar el comportamiento del aparato fonador, es conveniente que su diseño parta de un sistema que lo asemeje. El sintetizador que se diseñará se basa en la estructura que se presenta en el diagrama de la figura 4.1. Sin embargo, para que este funcione se requiere de un conjunto de parámetros que deben de ser extraídos de la señal original. El procedimiento general que se debe de seguir para poder calcular dichos valores se puede apreciar en la figura 4.2.

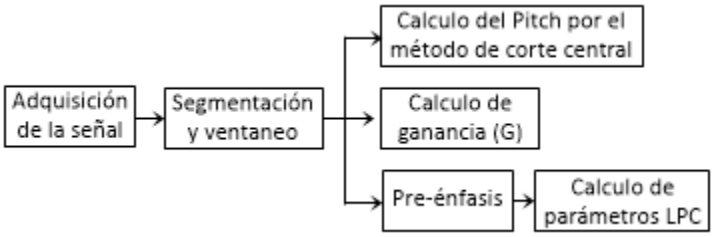


Figura 4.2: Modelo para la extracción de parámetros

4.1.1. Acondicionamiento de la señal

Cuando una señal es muestreada algunas de sus características pueden variar respecto a la original. Estas alteraciones, en algunas ocasiones, modifican de forma sustancial los parámetros requeridos para poder llevar a cabo un análisis satisfactorio. Sin embargo, es posible modificar algunas de estas características sin impactar en la información relevante contenida. Algunas de las características que pueden ser alteradas son la energía de la señal en el tiempo, las frecuencias contenidas por la señal, las características bajo las cuales la señal ha sido capturada (frecuencia de muestreo y tamaño de la palabra) y la intensidad de las frecuencias.

La energía de la señal en el tiempo, es una propiedad que puede modificarse con facilidad al multiplicar por un mismo escalar todas las muestras del conjunto, permitiéndonos incrementar o disminuir la intensidad de la señal. En voz, la amplitud de la señal representa el volumen, por lo que a menor magnitud será más difícil escuchar los sonidos. En muchos de los algoritmos empleados en computación se suele usar una magnitud que varía entre -1 y 1, aunque no siempre las señales llegan a tocar estos extremos. En algunas ocasiones la intensidad de la señal es muy baja por lo que se debe amplificar para poder ser analizada. Para lograr que una señal varíe entre -1 y 1 se dividen todos los valores entre aquel del conjunto con el valor absoluto de mayor magnitud, como se muestra en la siguiente función.

$$x_2 = \frac{x_1}{max(abs(x_1))}$$

Donde $max()$ es una función que extrae el valor con mayor magnitud positiva y $abs()$ es una función que obtiene el valor absoluto de todo un conjunto que recibe como entrada. El vector x_1 contiene la señal a normalizar, mientras que el vector x_2 almacenará la señal normalizada.

Las frecuencias contenidas pueden ser alteradas mediante filtros, con lo que es posible eliminar todas aquellas que permanecen de forma parasita y que son fáciles de caracterizar. Un ejemplo es la frecuencia de 60 [Hz] que se mantiene presente en cualquier medio en el que existan aparatos o cableado que transporte corriente eléctrica alterna. En voz se pueden suprimir frecuencias bajas con un límite de 50 [Hz], dado que la frecuencia fundamental o pitch de la voz humana masculina puede llegar a tomar este valor o mayores.

También es posible modificar la frecuencia de muestreo o el tamaño de las muestras. Sin embargo, solo es posible el disminuir la magnitud de éstas. Por ejemplo, si se posee una señal muestreada a 44100 [Hz] con tamaño de palabra de 16 [bits], es posible el disminuir la frecuencia de muestrea a 8000 [Hz] y con un tamaño de palabra de 8 [bits]. Pero no es posible el poder realizar el proceso contrario sin tener que generar valores que podrían o no corresponder a los reales. En voz se puede comprobar experimentalmente que la frecuencia de muestreo mínima tras la que se conserva plenamente la legibilidad de las palabras son los 8000 Hz [10].

Otra característica que puede ser modificada es la amplitud de las frecuencias. Esto se puede lograr con filtros que solo intensifiquen determinados valores pero no elimina ninguno. En voz se suele emplear el filtro de pre-énfasis para dar mayor peso a las frecuencias altas.

4.1.2. Cálculo de la ganancia

En señales de voz la ganancia permitirá asignar la energía correspondiente a cada segmento de la señal que equivale al volumen. El cálculo de este valor puede ser obtenido por varios métodos. El procedimiento que se empleará en este caso está basado en los valores atípicos. Para este método se considera que los valores con mayor relevancia son aquellos que se encuentran entre el 25 % y el 75 % del total de datos, acomodados de menor a mayor, considerando el valor absoluto de cada uno. A toda la información que permanece fuera de este rango se le está considerando como información poco relevante o con valores que podrían dañar el resultado conjunto. La magnitud de la ganancia se obtiene tras promediar los valores dentro del rango ya mencionado.

Por lo tanto, el cálculo de la ganancia queda definido por el siguiente procedimiento

- Sea $x[k]$ un vector que contiene n datos, se debe de obtener el valor absoluto de cada uno de ellos.
$$y[k] = |x[k]|,\ k = 1, 2, 3..., n$$
- Se ordenan los datos de menor a mayor
- Cálculo de las posiciones equivalentes al 25 % y al 75 % del total de datos
$$k_1 = 0{,}25n$$
$$k_2 = 0{,}75n$$
- Se calcula el promedio con todos los valores que están dentro del rango del paso anterior.
$$G = \frac{1}{k_2 - k_1} \sum_{i=k_1}^{k_2} y[i]$$

4.1.3. Cálculo del Pitch por el método de corte central

El pitch o frecuencia fundamental es un valor característico de las señales de voz sonoras. Este representa la frecuencia base bajo la que están construidas dichos sonidos. Su cálculo se basa en la idea de que toda señal sonora es periódica y ese periodo puede ser estimado por medio del método de auto-correlación [10].

El método de la auto-correlación permite resaltar la periodicidad de una señal. En la figura 4.3 se pueden observar algunos ejemplos de su aplicación. En la columna izquierda se presentan las señales base que se emplearon para generar su equivalente de auto-correlación presentada en la columna de la derecha.

Como se puede observar, en el caso 'a' de la figura 4.3, el algoritmo de auto-correlación genera una señal oscilante-periódica con magnitud decreciente a partir de

una señal periódica con una única frecuencia. Cuando a la señal base se le agrega una nueva frecuencia el comportamiento de la señal de auto-correlación también se ve modificado, como se presenta en el caso 'b'. Se observan nuevas oscilaciones, donde una de ellas sobresale a las demás. Finalmente, en el caso 'c' se puede observar una señal con más de una frecuencia y afectada por ruido y sin embargo su función de auto-correlación es muy parecida al caso 'b2'. El valor del pitch es considerado como la distancia entre los picos con mayor magnitud de la señal de auto-correlación **??**.

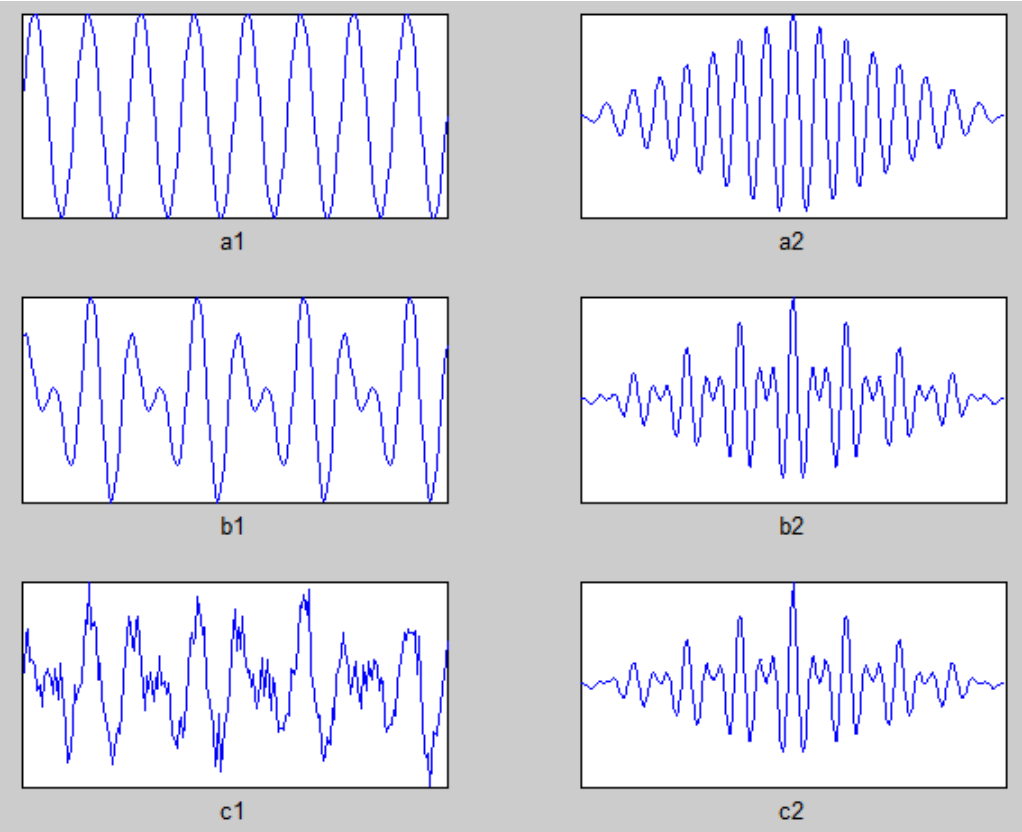


Figura 4.3: Ejemplos de señales y su auto-correlación

La señal de auto-correlación tiene $2N - 1$ datos, donde la magnitud localizada en la posición N es la de mayor intensidad. El algoritmo que se emplea para calcular el pitch busca el valor más alto después del localizado en la posición N. La distancia que existe entre estos dos valores es equivalente al periodo fundamental de la señal **??**.

Un método que se suele emplear para optimizar el algoritmo de auto-correlación es el método de recorte central, el cual trata de suprimir la información no relevante de la señal. Este consiste en suprimir todos los datos que permanecen dentro de un rango específico. Para obtener el umbral de discriminación se debe de dividir la señal en tres partes y obtener de la primera y tercera la amplitud máxima. Posteriormente se aplica la ecuación 4.1 para determinar el parámetro d. Esta ecuación establece que hay que obtener el valor mínimo de las opciones de entrada (A_1 y A_2) y multiplicarla por una constante (K), la cual suele variar entre 0.6 y 0.8 [10].

$$d = Kmin(A_1, A_2) \tag{4.1}$$

La nueva función que se generará a partir del uso del umbral es la siguiente

$$y[k] = \left\{ \begin{array}{ccc} x[k] - d & si & x[k] \geq d \\ x[k] + d & si & x[k] \geq -d \\ 0 & , & \text{cualquier otro caso} \end{array} \right\} \tag{4.2}$$

Tras calcular la señal con el recorte central se puede implementar nuevamente la función de auto-correlación La cual contendrá una menor cantidad de ondulaciones. Sin embargo, dado que el método solo elimina información no relevante, puede presentarse el caso en el que señales no periódicas presenten picos aleatorios. Por esta razón, se suele evaluar que las magnitudes de los picos sean superiores al 30 % de la magnitud del pico central, de la posición en N **??**.

En el segmento (a1) de la figura 4.4 se puede observar la representación de la vocal 'a', que es una señal sonora. Se puede apreciar en la sección (b1) la misma señal a la cual ya se la ha aplicado el recorte central, y en la columna derecha se representan las auto-correlaciones correspondientes a cada una de ellas. Es apreciable que la auto-correlación de la segunda señal posee menos información pero conservan los picos con mayor relevancia en las mismas posiciones. Esto facilita la estimación de dicha información.

La distancia que hay entre los picos de la señal puede ser convertida a frecuencia tras dividir la frecuencia de muestreo entre el valor medido. Es conveniente el almacenar el valor medido para poder ser empleado posteriormente, ya que suele ser de mayor utilidad que la frecuencia asociada.

Este valor de pitch nos ayudará a clasificar las señales que estamos analizando como sonoras o no sonoras. Asimismo, nos dará la base para generar la señal de tren de impulsos para aquellas opciones donde no tome valor de cero.

4.1.4. Uso de parámetros LPC como coeficientes del filtro para la síntesis

Los coeficientes LPC representan el comportamiento de la señal que los han generado. Estos valores pueden ser aplicados como coeficientes de un filtro que darán forma a la señal que reciba de entrada. El resultado que proporcionará este filtro será un aproximado de la señal que dio origen a los coeficientes.

El inverso de la representación en frecuencia de los coeficientes LPC proporciona un aproximado de la envolvente de la representación en frecuencia de la señal original, como

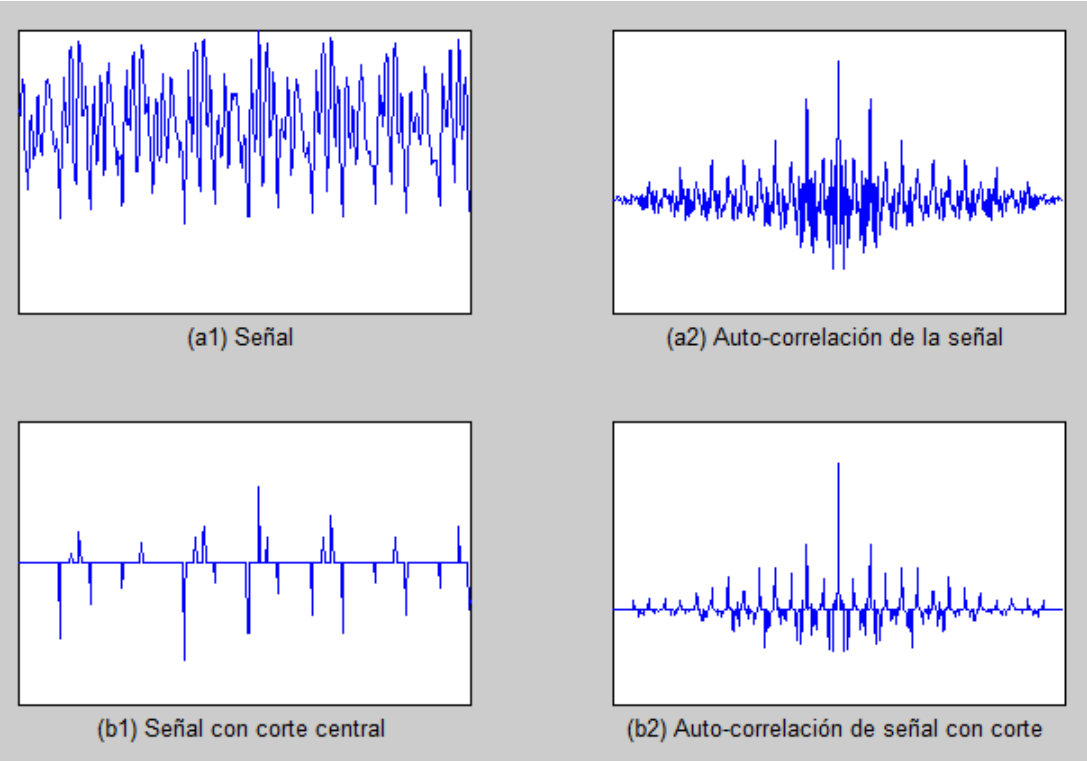


Figura 4.4: Representación de la vocal 'a' y su auto-correlación con y sin recorte central

se ve ejemplificado en la figura 4.5. Este ejemplo se basa en la extracción de solo diez parámetros LPC de la señal, sin embargo si se llegaran a calcular una mayor cantidad de estos la envolvente sería más fiel a la forma de la señal original en frecuencia.

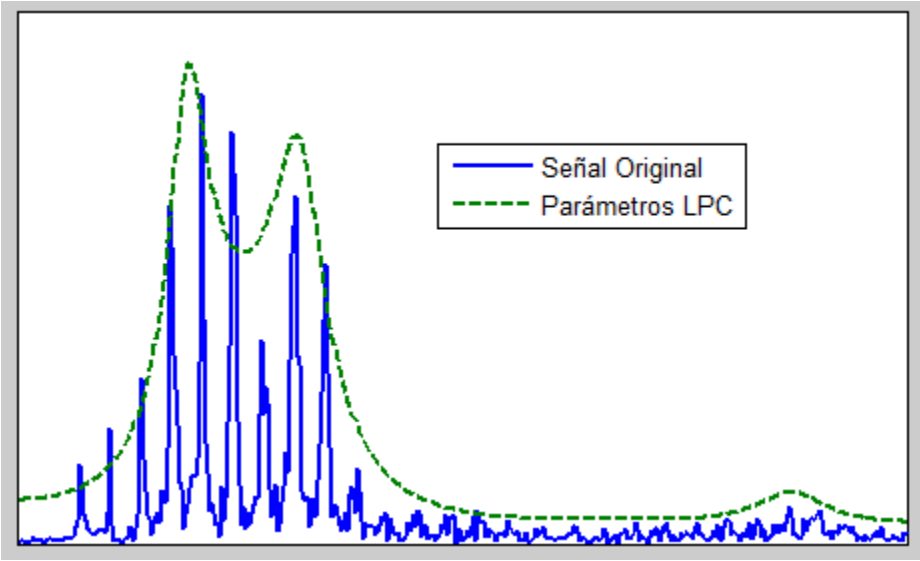


Figura 4.5: Comparación de la representación en frecuencia de la señal y sus parámetros LPC

Cuando se emplea un filtro basado en ecuaciones en diferencias del tipo IIR, definido por la ecuación 2.15. Se pueden emplear los coeficientes LPC para formar un filtro todo polos, como se plantea en 4.3. Este filtro combina los valores pasados de la señal filtrada con el valor de entrada actual para predecir el valor actual de la señal filtrada.

$$y_k = x_k + \sum_{m=1}^{M} a_m y_{k-m} \tag{4.3}$$

Por lo tanto, el filtro se encargará de que solo prevalezcan las frecuencias y sus magnitudes denotadas por la envolvente que genera los parámetros.

4.2. Implementación del sintetizador

Para poder implementar el sintetizador de palabras que aquí se ha propuesto, es necesario el poder manipular archivos de sonido. Hay que tener en cuenta que si el entono de desarrollo no cuenta con las funciones que se requieren, como son los filtros o las operaciones con matrices, es necesario que se desarrollen dichas funciones. MATLAB es un entorno de desarrollo en el que se pueden realizar pruebas de algoritmos de forma fácil, además de visualizar la información cómodamente gracias a la gama de funciones y herramientas que esta incluye. Esta es una buena opción para evaluar el desempeño del algoritmo propuesto para poder ser llevado a otros entonos más simples (código fuente: [19]).

De acuerdo con el diagrama de la figura 4.1, el sistema de síntesis de voz requiere una serie de parámetros para poder realizar su trabajo. Este necesita una matriz que contenga los parámetros LPC de todos los segmentos de la señal original. Asimismo, se requiere un vector con las ganancias y otro con los valores del pitch asociados a cada fracción. Asimismo se requiere saber la cantidad de datos que tenía cada ventana y las características bajo las que se muestreo la señal.

Con el vector que contiene los valores de pitch se determinarán los segmentos cuya base será una señal de ruido blanco y aquellas con base de tren de impulsos. Estos últimos tendrán una frecuencia equivalente a la que indica su respectivo pitch. Las señales de ruido blanco se generarán cuando el valor de pitch sea cero, es decir, no hay frecuencia fundamental. El tamaño de la ventana de la señal que se ha de generar debe de tener la misma cantidad de muestras de la señal que se analizó originalmente, sin considerar el sobrepaso. Sin embargo el tamaño final de la señal puede variar al de la señal original dado que se pudo haber realizado un redondeo en la última ventana.

El tren de impulsos se obtiene tras generar una señal con magnitudes iguales a cero. Posteriormente se asignan valores unitarios a cada posición con una separación dada por el pitch. Este último debe de estar representado en número de muestras. Por otra

parte, las señales de ruido suelen ser generadas con los algoritmos de valores aleatorios asociados a los lenguajes de programación que se están empleando.

Las señales que se han generado requieren que se les asigne una ganancia equivalente a la que poseía su igual en la señal original. Con esta asignación se ha adecuado la señal con las características necesarias para poder ser procesada por el filtro de parámetros LPC. Este dará como salida una señal con la forma que definen el conjunto de parámetros. Finalmente, la señal debe de ser pasada por un segundo filtro, el filtro de des-énfasis, que atenuará las frecuencias altas y resaltará el rango donde se localizan los diferentes valores de pitch.

Es recomendable que antes de implementar el filtro de des-énfasis, la señal sea pasada por un filtro paso altas que elimine las frecuencias por debajo de los 50 Hz. Esto último dado que se ha observado de forma experimental que al generar una señal sintética estas frecuencias pueden aparecer con magnitudes que influyen en la señal. Cuando es aplicado el filtro de des-énfasis, estas frecuencias adquieren una relevancia mayor de la que deberían de tener.

4.3. Evaluación de resultados

En el anexo **??** se puede apreciar el algoritmo del sintetizador de voz aplicado en Matlab. Este lee un archivo de audio en formato *wav*, el cual será procesado para extraer los parámetros LPC, de los distintos segmentos, las ganancias y el pitch. Posteriormente se presenta el programa propio del sintetizador, el cual emplea los parámetros obtenidos con antelación.

En la figura 4.6 se puede apreciar dos señales de voz, con un tamaño de 8000 muestras, donde la segunda ha sido obtenida tras hacer pasar la primera por el sistema de síntesis de voz. Como se puede apreciar la forma se la señal es parecida aunque no idéntica. La distribución de la señal es similar en ambos casos y los cambios de amplitud también tienen un comportamiento semejante.

Al expresar estas mismas señales en el dominio de la frecuencia, como se observa en las gráficas de la figura 4.7, se aprecia que el comportamiento es muy simular en cuanto a las amplitudes y las frecuencias de relevancia. A pesar de que en la señal sintética hay una mayor cantidad de frecuencias de baja intensidad, se pueden apreciar con claridad aquéllas que tienen mayor presencia en la señal real.

En la figura 4.8 se puede apreciar la representación de la señal original y sintética en forma de espectrograma. Se visualiza en un color rojizo aquellas partes en las que la energía es más intensa. En ambas representaciones se aprecian los periodos en donde existe la señal de voz. A diferencia de la señal original, la señal sintética resalta más las secciones donde hay voz. Respecto a la frecuencia, se hace notar que hay una mayor carga en los valores bajos.

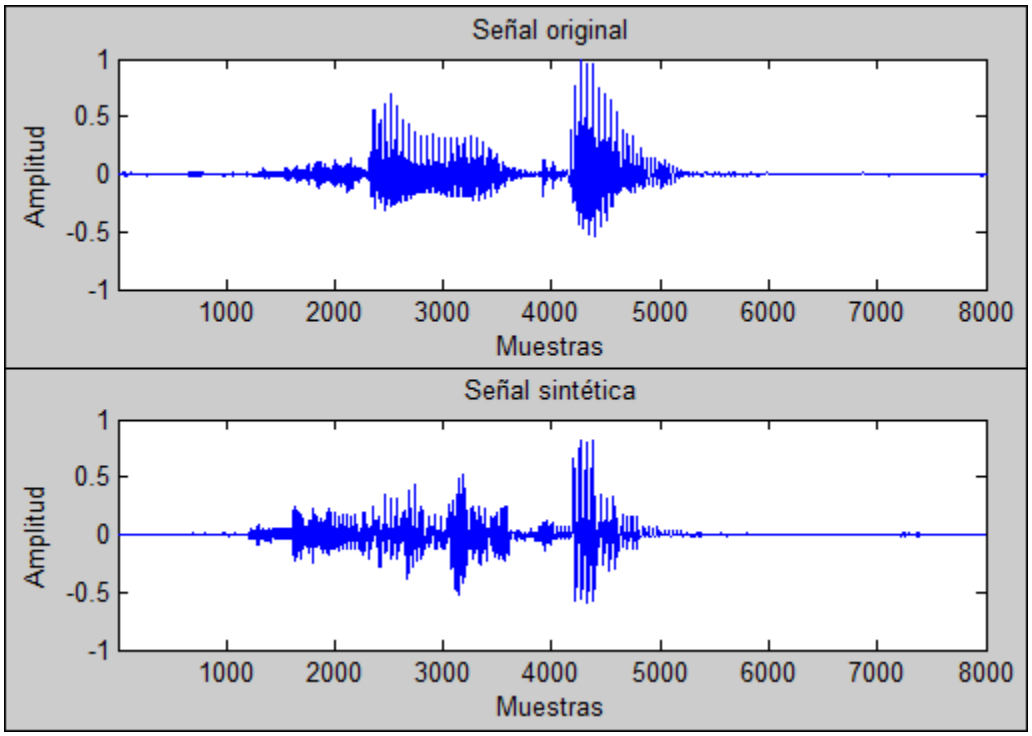


Figura 4.6: Comparación de señal real y sintética de la palabra 'cinco' en el dominio del tiempo

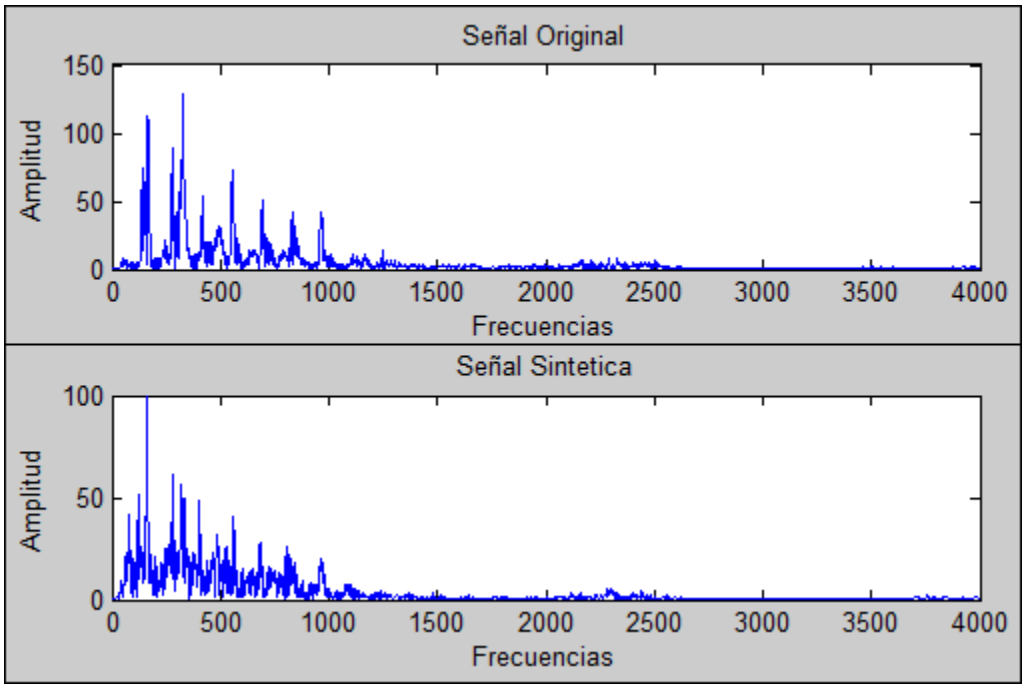


Figura 4.7: Comparación de la señal real y sintética de la palabra 'cinco' en el dominio de la frecuencia

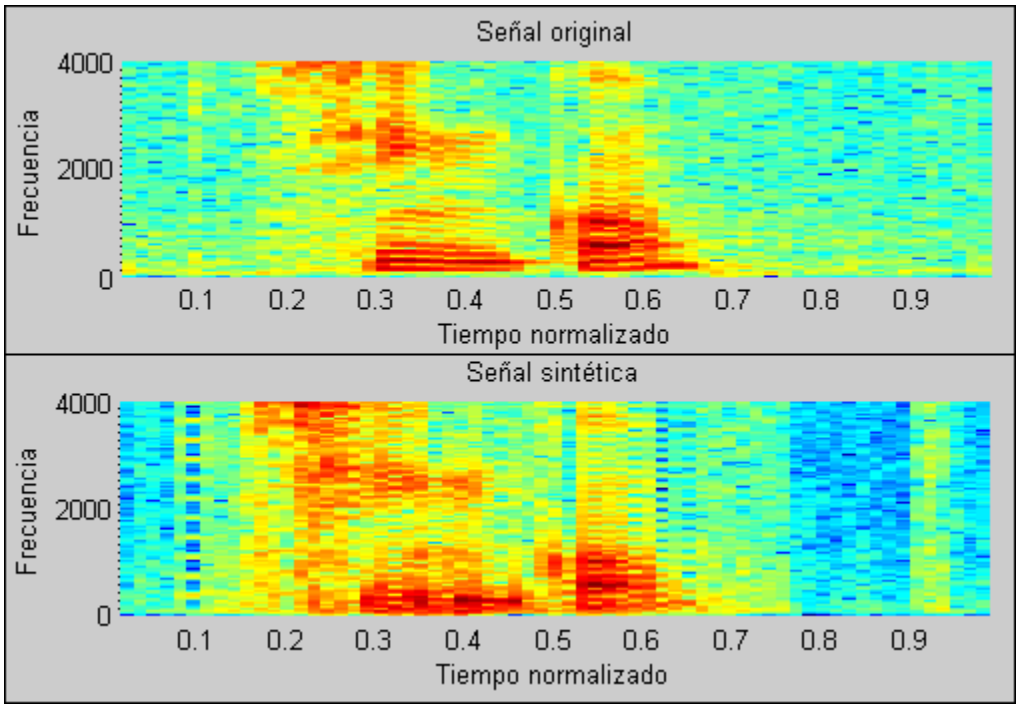


Figura 4.8: Comparación de la señal real y sintética de la palabra 'cinco' en un espectrograma

El segundo ejemplo se basa en una señal compuesta por el abecedario como se observa en la figura 4.9, junto con su correspondiente simulación. Como es apreciable, las dos señales comparten una estructura en el tiempo muy similar. Las principales variaciones se hacen notar en las amplitudes a lo largo de todo el conjunto, pero aún conservando su forma.

Asimismo, al ser representadas estas señales en el dominio de la frecuencia se puede observar una gran similitud, figura 4.10. Al igual que en el ejemplo anterior, la señal sintética retiene una gran cantidad de frecuencias que no aparecen en la primera. Aquéllas con mayor relevancia se aprecian con claridad en ambos casos aún cuando su amplitud varía.

Al realizar un análisis más detallado de un par de fragmentos de las señales, que representen el mismo vocablo, se puede observar un comportamiento más concreto. Como se puede apreciar en la figura 4.11 al realizar el análisis de las vocales 'a' y 'o', las frecuencias que se comparan tienen un mayor parecido que en el ejemplo de todo el abecedario. Sin embargo se siguen apreciando algunas frecuencias parasitas, y amplitudes diferentes.

Como se puede observar en las diferentes gráficas mostradas, el sintetizador logra generar la señal de voz deseada. Aun cuando se observan variaciones en la amplitud de las señales, tanto en el dominio del tiempo como en el dominio de la frecuencia, los resultados pueden ser considerados satisfactorios. Esto último se puede confirmar al

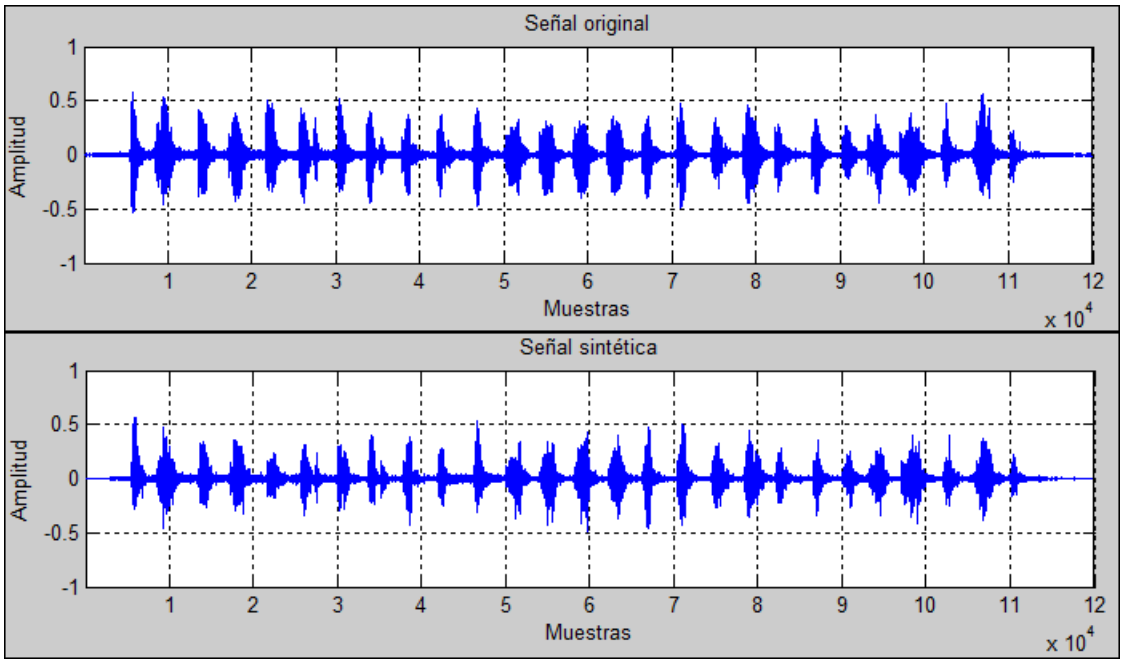


Figura 4.9: Comparación de señal real y sintética de todas las letras del abecedario en el dominio del tiempo

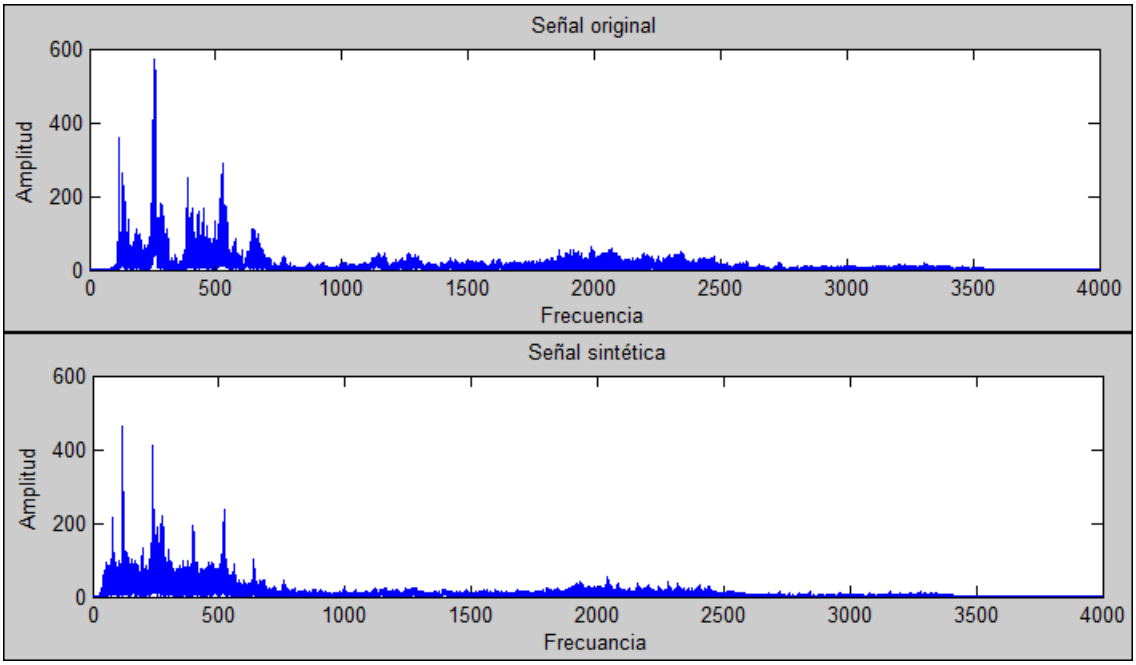


Figura 4.10: Comparación de señal real y sintética de todas las letras del abecedario en el dominio de la frecuencia

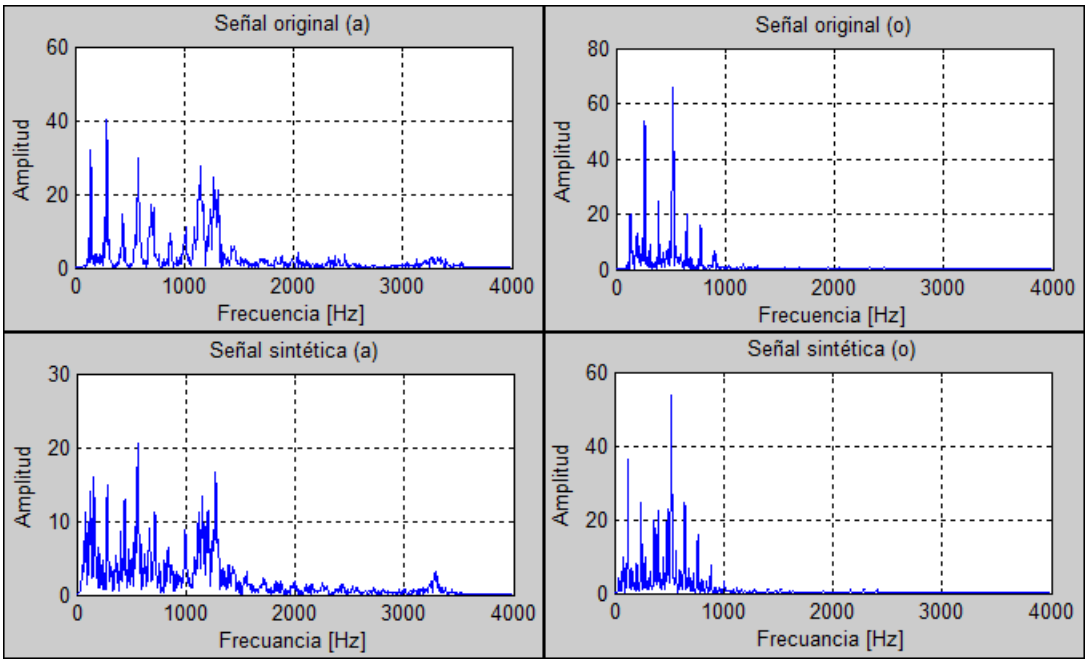


Figura 4.11: Representación en frecuencia de las letras 'a' y 'o' en el dominio de la frecuencia

momento de escuchar y comparar las dos señales

El espectrograma de la figura 4.12 nos permite realizar una comparación más detallada de las señales originales y sintéticas. Como es apreciable, muchas de las frecuencias son parecidas en cualquiera de los casos. Así mismo, la distribución de cada una de ellas en el tiempo es similar.

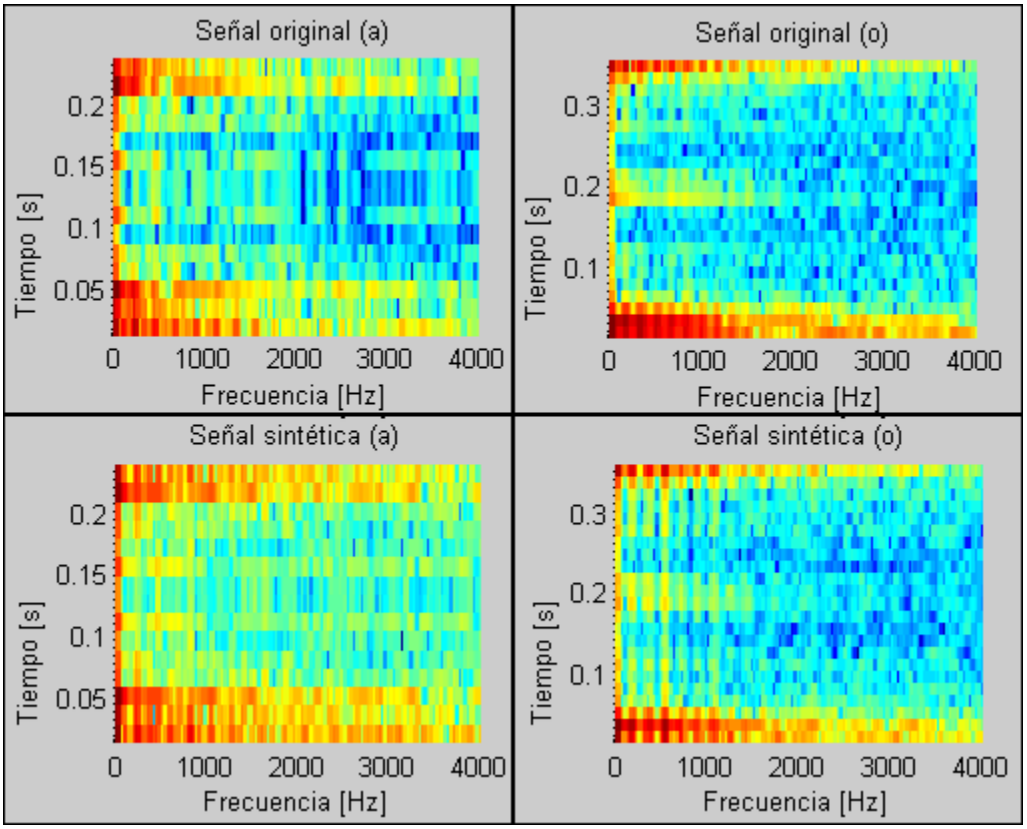


Figura 4.12: Espectrograma de las señales de las letras ‘a’ y ‘o’ originales y sintéticas

Capítulo 5

Implementación del reconocedor de palabras aisladas en tiempo real

El reconocedor de palabras aisladas que se describe a continuación está basado en los mismos principios que se emplearon en la sección anterior para obtener los parámetros característicos de la señal. Se considera que para un conjunto de señales que representan la misma palabra se tendrán características similares. De este grupo se podrá obtener el sub-conjunto con mayor peso a partir del cual sea posible relacionar a los demás y a otros posibles casos que surjan.

Dado que este reconocedor actúa en tiempo real, sus procesos se dividen en dos grandes segmentos, la lectura del audio y su análisis, como se puede observar en la figura 5.1. La primera parte se encarga de capturar el audio proveniente de la tarjeta de sonido, además de adecuar los valores obtenidos al formato que se usará. El segundo segmento se encarga de analizar el audio adquirido según sean las necesidades.

Figura 5.1: Esquema de captura y análisis de audio en tiempo real

Durante la primera iteración únicamente se ejecutará el módulo de la lectura ya que se considera que no hay valores de audio almacenados con antelación. Posteriormente se ejecutan de forma paralela la lectura y el análisis, donde este último trabaja sobre la información capturada en el caso anterior. Al terminar el ciclo se deja de adquirir audio, pero se realiza un último análisis para manipular el último bloque de información.

El segmento de análisis de audio es el encargado de procesar la información para realizar el reconocimiento. Los módulos que lo componen se ven expresados en el diagra-

ma de la figura 5.2. Este sistema considera que la señal no tiene un tamaño finito, que la señal de audio que se desea puede aparecer en cualquier instante de tiempo dentro de la principal y que solo se puede analizar un segmento de está en cada iteración.

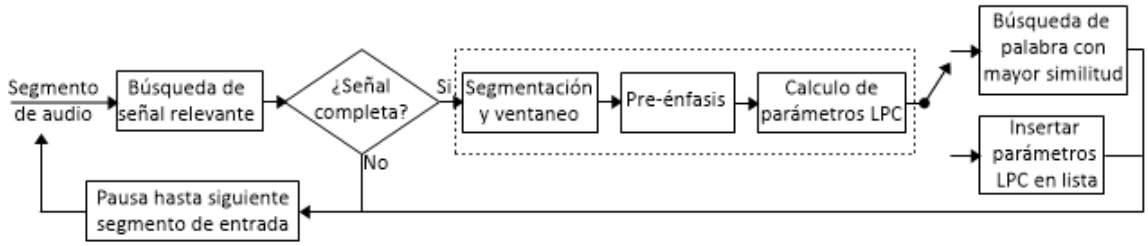


Figura 5.2: Esquema del reconocedor de palabras aisladas en tiempo real

En el diagrama de la figura 5.2 se observan dos opciones en su parte final. El primero se encarga de comparar los parámetros LPC con una base de datos para determinar la de mayor similitud. El segundo almacena los parámetros LPC en una lista para poder analizarlos una vez que se ha culminado la adquisición y generar un nuevo conjunto que formará parte de la base de datos.

5.1. Diseño del reconocedor

5.1.1. Extracción y aislamiento de voz de una señal con base en su energía

Es común que en las grabaciones de palabras aisladas se tengan presentes datos que podrían llegar a ser descartados al no presentarnos información de utilidad. Estos valores son el resultado de los silencios que acompañan a las palabras antes y después de su enunciación. Sin embargo, estas mismas pausas se pueden encontrar dentro de la misma palabra en la separación de silabas. En la figura 5.3 se puede apreciar la gráfica correspondiente a la palabra “cinco”. En ella se destacan los silencios que rodena a la palabra y la pequeña pausa que se tiene entre las silabas “cin” y “co”.

Cabe aclarar que para este caso, así como para todas de las grabaciones empleadas en esta sección, se realizaron las adquisiciones con una frecuencia de muestreo de 8000 [Hz], un tamaño de palabra de 8 [bits] y empleando un único canal.

Para poder realizar el análisis de la información contenida en la señal, es recomendable eliminar los silencios que rodean a la palabra. Esto con el fin de que todas nuestras muestras tengan una forma parecida y distribuida en el tiempo, como se ejemplifica en la figura 5.4. De esta forma aseguramos que los métodos aplicados trabajen bajo un conjunto de muestras teóricamente iguales.

El aislamiento de la información se puede llevar a cabo en dos pasos. Primeramente

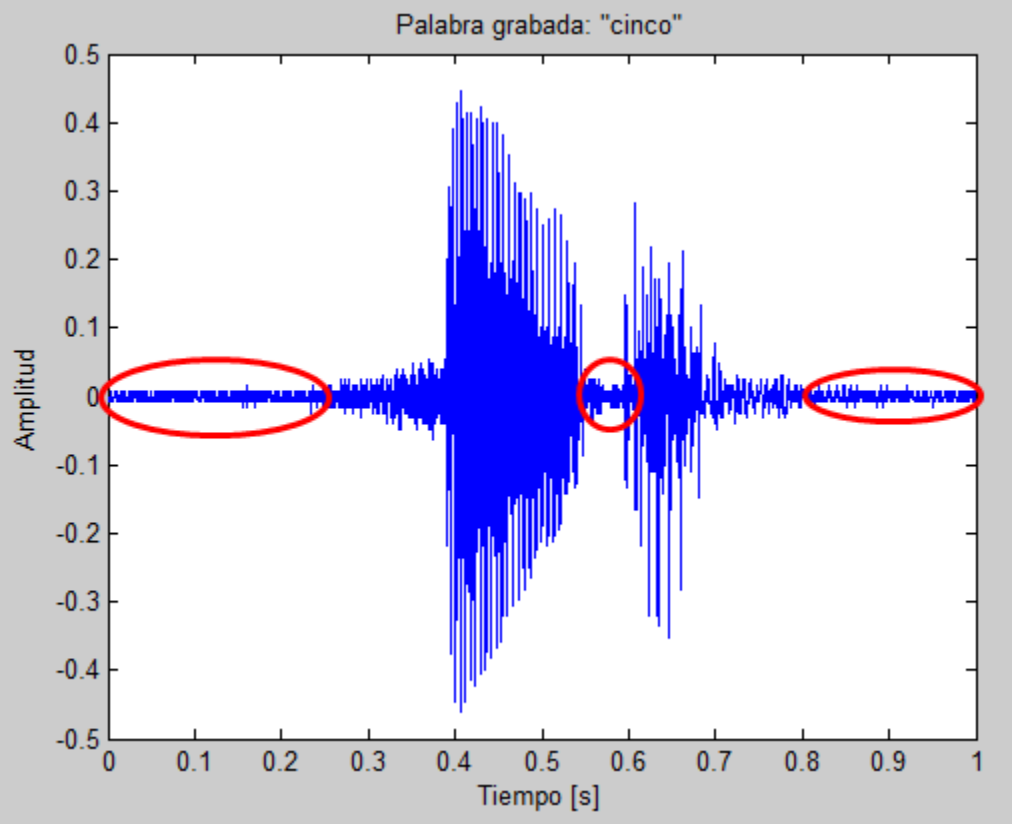


Figura 5.3: Esquema del reconocedor de palabras aisladas en tiempo real

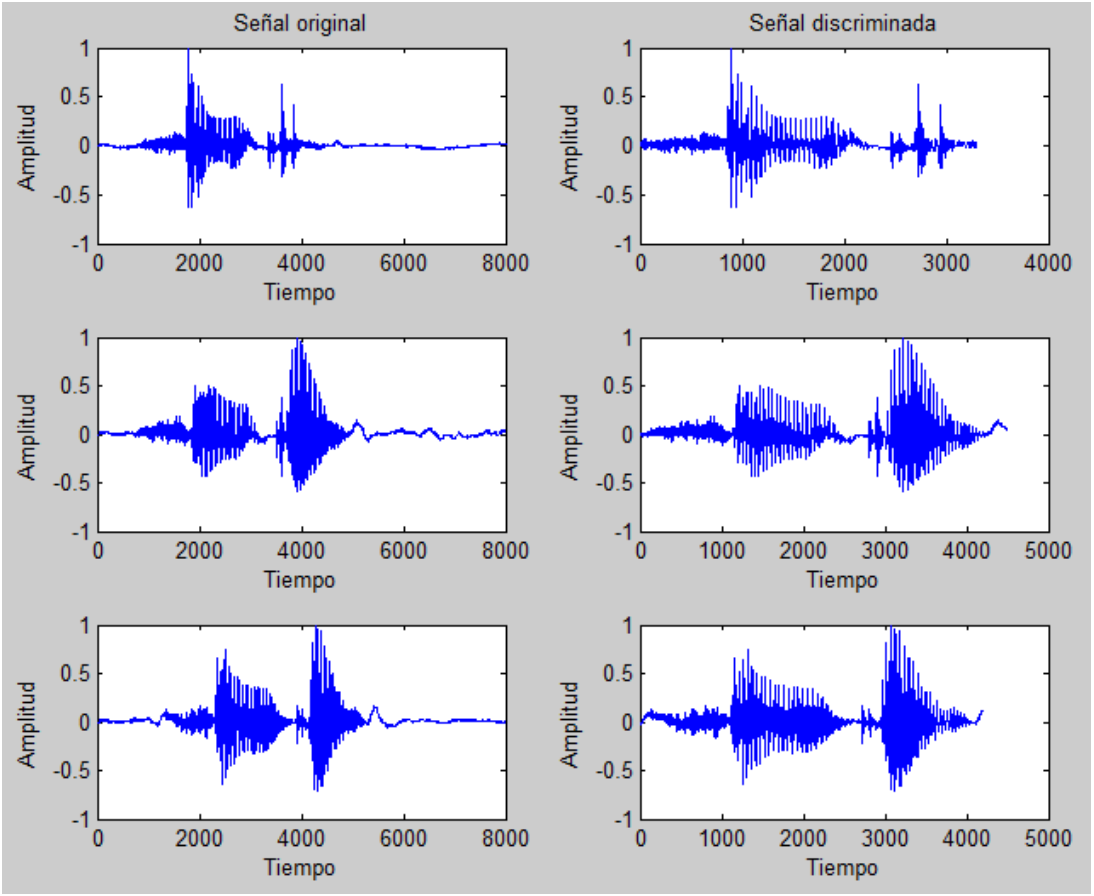


Figura 5.4: Discriminación de señal con información relevante

se deberá de identificar las partes de la señal que sean silencio y aquellas que creamos que sean voz. En segundo lugar habrá que suprimir o eliminar únicamente las partes de silencio que rodean a nuestra señal de voz, como en los ejemplos de la figura 5.4.

Una forma para poder distinguir la parte de la señal correspondiente al silencio, es por medio del uso de la energía. Esta brindará valores con magnitudes grandes para las partes de la voz en comparación con los silencios.

La energía es la suma del cuadrado de los valores. Esto provoca que si se tiene una amplitud alta en la señal, la energía lo será aún más. Mientras que si es baja la amplitud, la energía será mucho más baja. Asimismo nos presenta la ventaja de que todos los valores de energía calculados serán mayores o iguales que cero

Al emplear este método se fraccionará la señal en segmentos del mismo tamaño para obtener un vector que contendrá el conjunto de energías, cuya forma gráficamente asimilará al contorno de la señal original. Dependiendo de la cantidad de muestras que se tomen para obtener la energía, se podrá ver un comportamiento diferente de la envolvente. En la figura 5.5 se ejemplifican cuatro casos en los que se toman distintos tamaños.

Se puede observar que a menor número de muestras por división se tiene un comportamiento más acorde al contorno de la señal. Esto permite precisar que partes tienen menor o mayor energía de una forma más exacta. Sin embargo, ya que el objetivo es solo determinar el punto en que inicia o termina nuestra señal, se puede optar por un número de muestras mayor. Hay que tener en cuenta que si se toma un número demasiado grande, las partes con mayor energía podrían ocultar a aquellas con poca, como es el caso del último ejemplo de la figura 5.5.

Una vez que se ha extraído la energía de la señal a analizar se requiere establecer un límite mínimo aceptable o límite de discriminación. Este nos podrá indicar que conjunto de datos poseen suficiente energía para no ser considerados como silencio. Sin embargo, el índice no puede ser generalizado ya que las condiciones bajo las cuales se realicen las grabaciones no siempre serán las mismas y los silencios podrían tener mayor o menor energía en cada situación.

Para la obtención de este índice se propone el realizar una grabación de silencio, previa a la captura de voz, con una duración de mil muestras (0.125 segundos). A partir de esta se podrá estimar un valor asociado a las condiciones actuales bajo las que se está realizando el procedimiento y que podría considerarse constante para analizar la grabación posterior.

La estimación del índice se puede realizar a partir del conjunto de energías obtenidas de la señal de silencio, que podría ser calculado mediante la distribución normal estándar. Esta nos daría un valor mínimo y máximo bajo el cual poder actuar, sin embargo solo sería necesario mantener el superior.

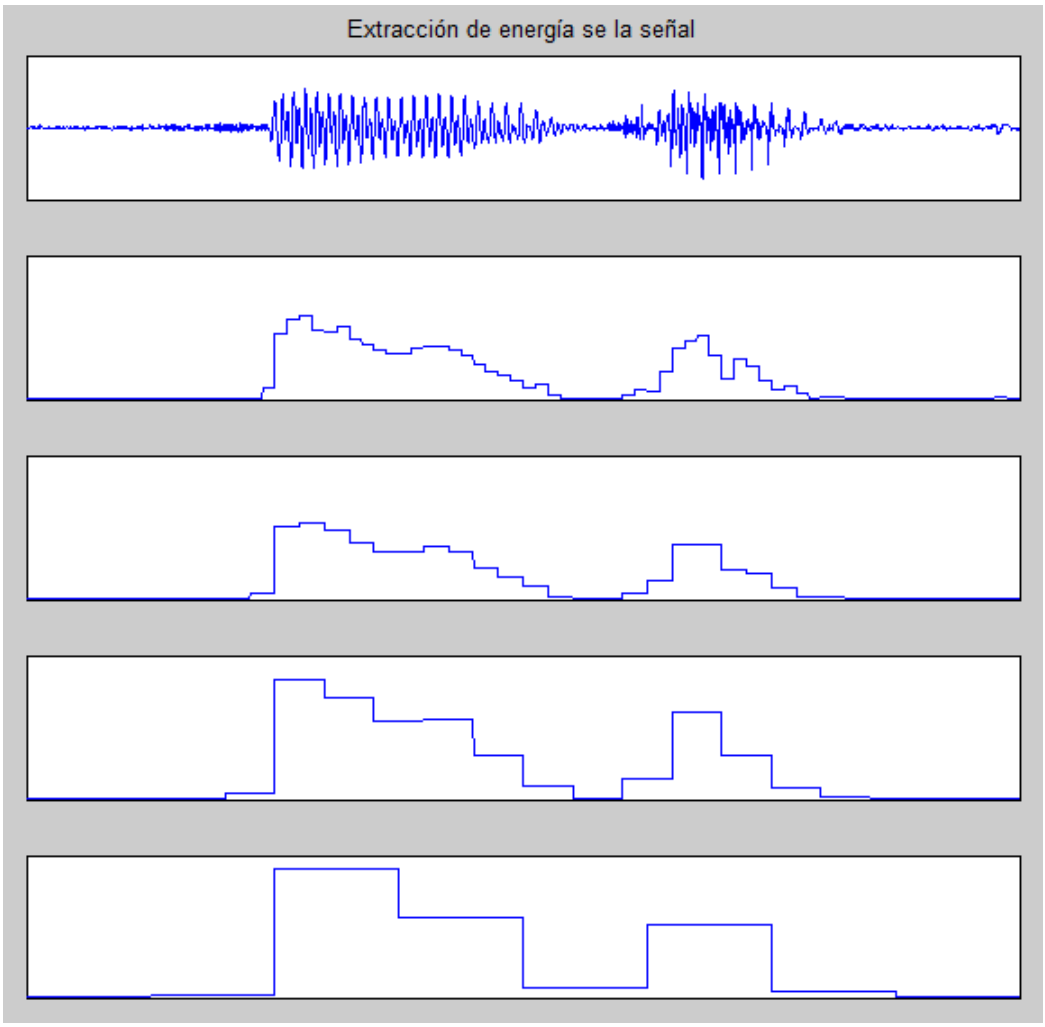


Figura 5.5: Extracción de energía de la señal con distintos tamaños de vectores, 50, 100, 200 y 500 muestras respectivamente

Al aplicar el índice obtenido de la desviación estandar sobre las señales se pueden suprimir aquellas partes cuya energía sea inferior a la que nos indica. En la figura 5.6 se puede observar la discriminación de la señal. Con fines prácticos se volvieron infinitos aquellos valores que se consideraron como silencio, ya que gráficamente no se ven. Además se aplicó una división de cien muestras para la obtención de energía.

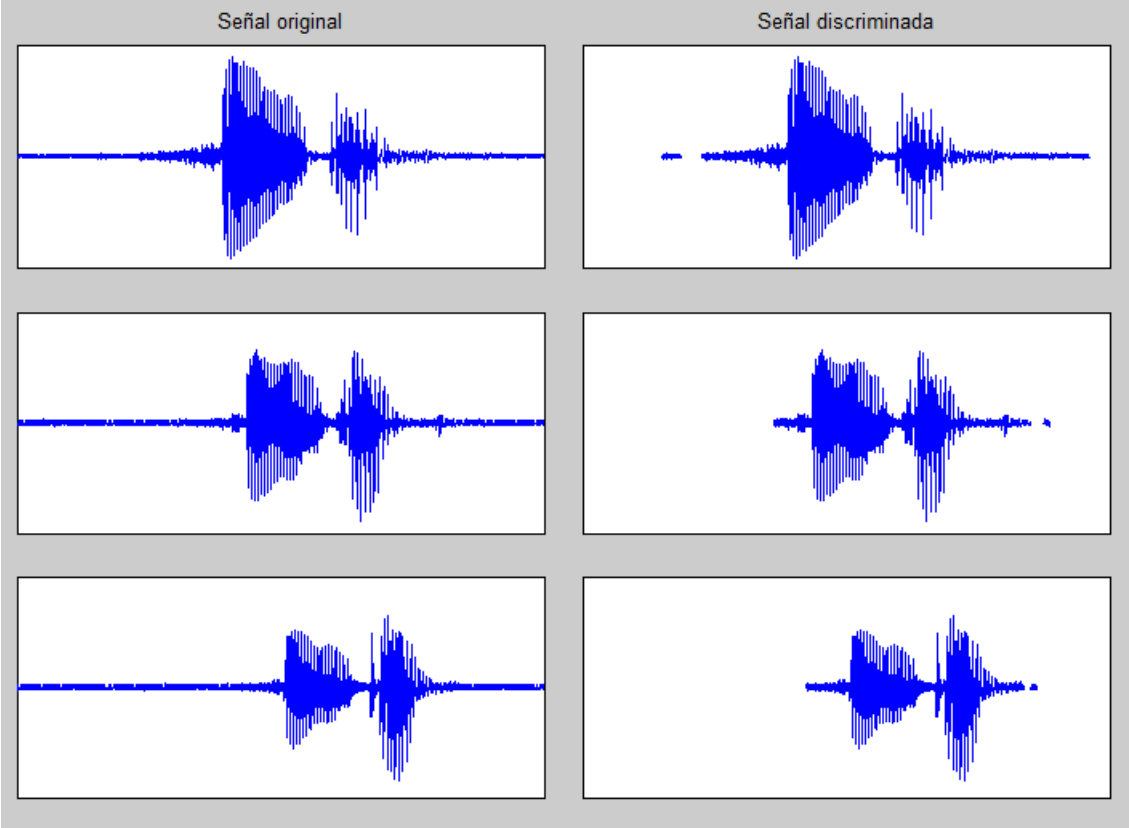


Figura 5.6: Eliminación del silencio dentro de la señal grabada

En la figura 5.7 se puede observar el espectrograma de las señales originales y las señales discriminadas. Se observa que son eliminadas aquellas partes donde la energía de la señal es baja, manteniendo las secciones que contienen a la señal de voz. El espectrograma permite observar que se mantienen todas las frecuencias contenidas en la voz.

Como se puede observar, este método logra suprimir el silencio que rodea a la señal que posee la energía con mayor relevancia y que consideramos nuestra señal de voz. Podemos aplicar el algoritmo para aislar los datos que aún permanecen. Este algoritmo se puede resumir de la siguiente forma

1. Calculo de índice de discriminación

    a) Se realiza una grabación de silencio de 1000 muestras, equivalente a 0.125 segundos

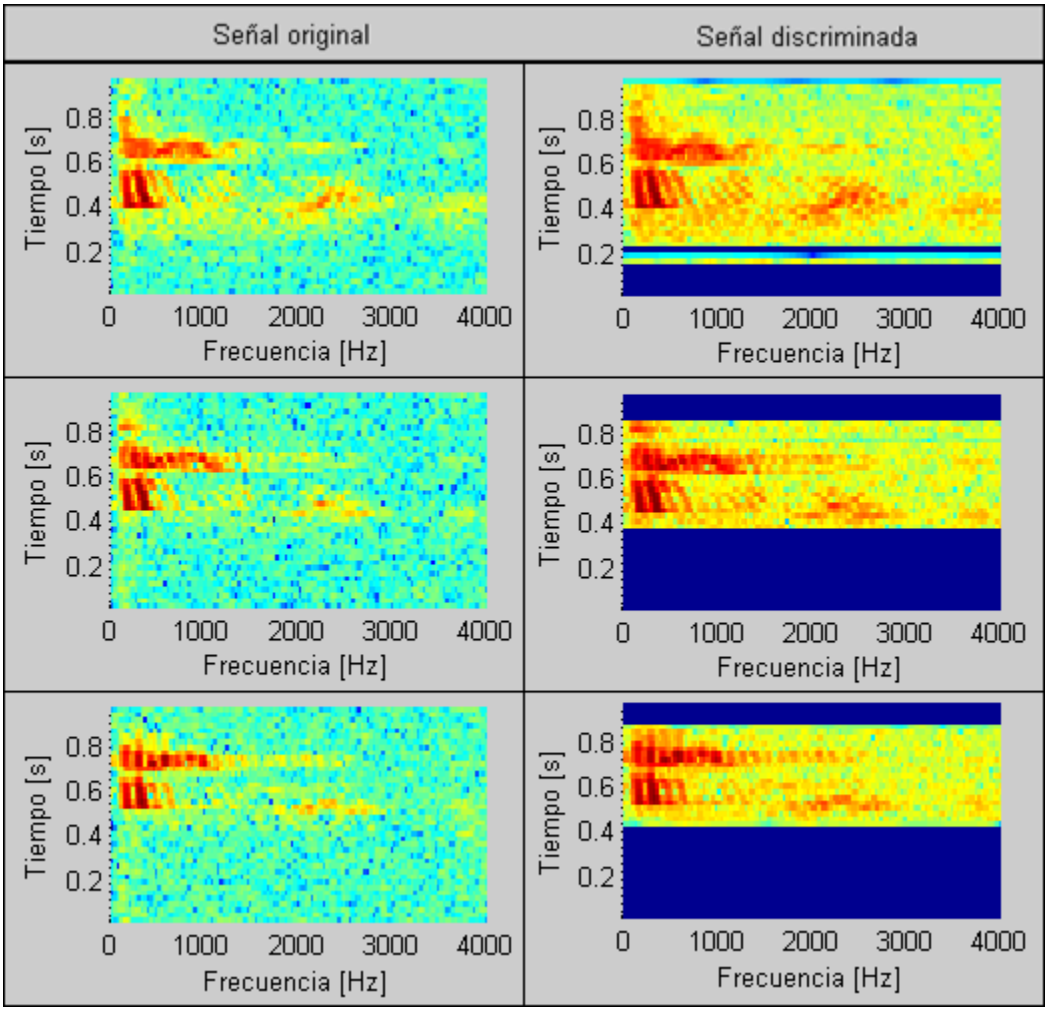


Figura 5.7: Espectograma de la eliminación del silencio dentro de señales grabadas

b) Se obtiene la energía de esté vector con segmentos de 100 muestras

c) Se calcula la media ($\tilde{x}$) y la desviación estándar (σ)

d) El índice es igual a

$$indice = \tilde{x} + 3\sigma$$

2. Grabación de la palabra aislada

   a) Se graba una palabra aislada con un tiempo a considerar por el usuario

3. Calculo de energía y discriminación

   a) Se calcula la energía de la palabra con segmentos de 100 muestras

   b) Se eliminan o vuelven cero todos aquellos valores que estén por debajo del índice calculado en el punto *1.d*

Una vez que se han identificado los valores considerados como silencio, debemos de eliminarlos para que la señal deseada permanezca aislada. Esto se puede lograr si se identifica el inicio y el final de la señal de voz y así poder eliminar el resto o realizar una copia en otro vector.

Para aislar la señal consideramos que la información está contenida en un solo vector de tamaño indefinido y que tanto la energía de la señal como el índice de discriminación ya han sido calculados. De esta forma podemos considerar a la señal como una serie de variaciones entre dos valores, por encima o por debajo del límite. En la figura 5.8 podemos apreciar un ejemplo de esto, donde "señal con voz" hace referencia a las partes de la señal donde la energía superan el límite y "señal de silencio" el caso contrario.

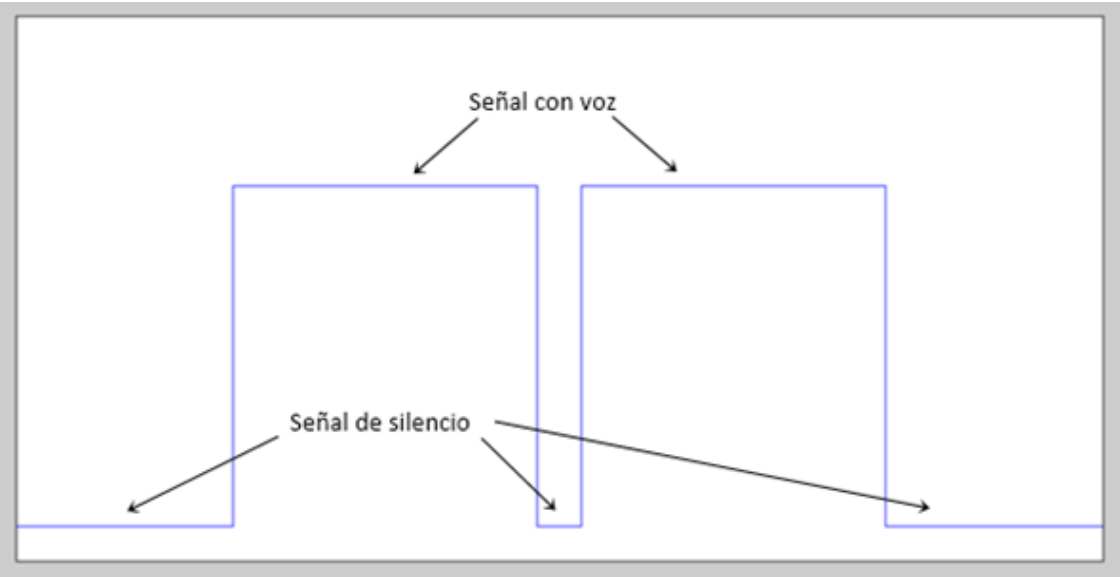


Figura 5.8: Representación de una señal de voz con referencia a su energía

El algoritmo que se propone trata de encontrar una determinada cantidad de valores de señal con voz consecutivos o separados por un lapso de silencio corto. Para esto, se requiere el considerar que todas las palabras que puedan ser usadas tendrán una duración mínima y que habrá un tiempo de silencio mínimo que podrían contener dichas palabras.

A continuación se enuncian los pasos a considerar al momento de realizar la búsqueda de los índices de inicio y fin de señal relevante dentro de la original:

1. Se considera que la señal siempre inicia en silencio, aún antes de los valores que se tienen registrados. Por lo tanto, cuando se detecte el cambio de "Señal de silencio" a "Señal de voz" se considera el inicio (*inicio*) de la palabra en la posición (i) de este último valor. Ver figura 5.9

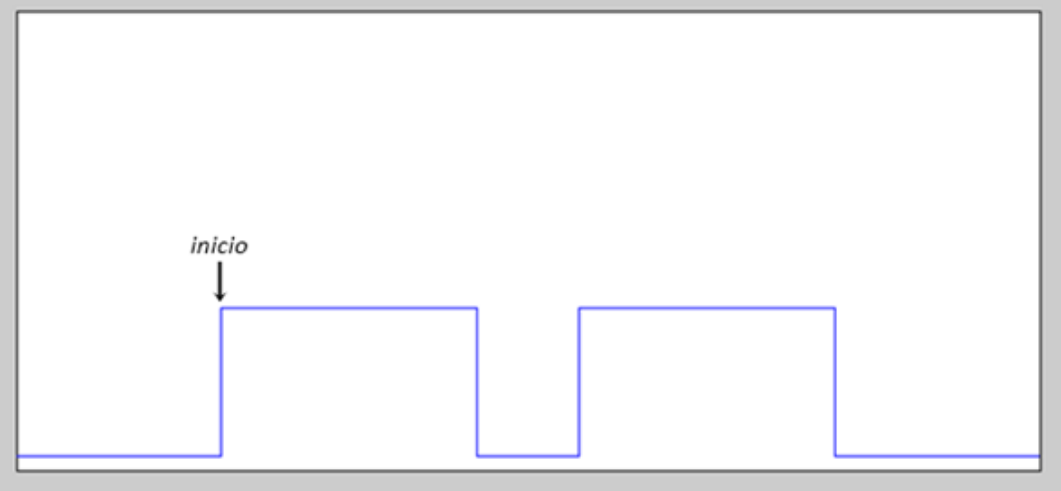


Figura 5.9: Inicio de la señal

2. Con cada intervalo que presenta energía, se incrementa un contador de energía ($ConEnergia$) en una unidad. De la misma forma, si se llegase a encontrar silencio, se incrementará un contador ($SinEnergia$) en una unidad, siempre y cuando este sea continuo. Ver figura 5.10

3. La comparación de $SinEnergia$ con el mínimo aceptable dentro de una palabra ($SinEMinima$) nos proporcionará una bandera con la que sabremos si se ha encontrado el final. Solo para el caso en que $SinEnergia$ supere $SinEMinima$, la bandera indicará que se ha encontrado el final. Asimismo, el contador $ConEnergia$ indicará si las muestras con energía son en número suficientes para considerarlas un segmento de señal de voz, esto al compararlo con el mínimo aceptable ($ConEMinima$).

4. $SinEnergia$ comienza con el valor de cero y se incrementará cuando se encuentren valores de silencio continuos. Por ejemplo, si el silencio se presenta dentro de

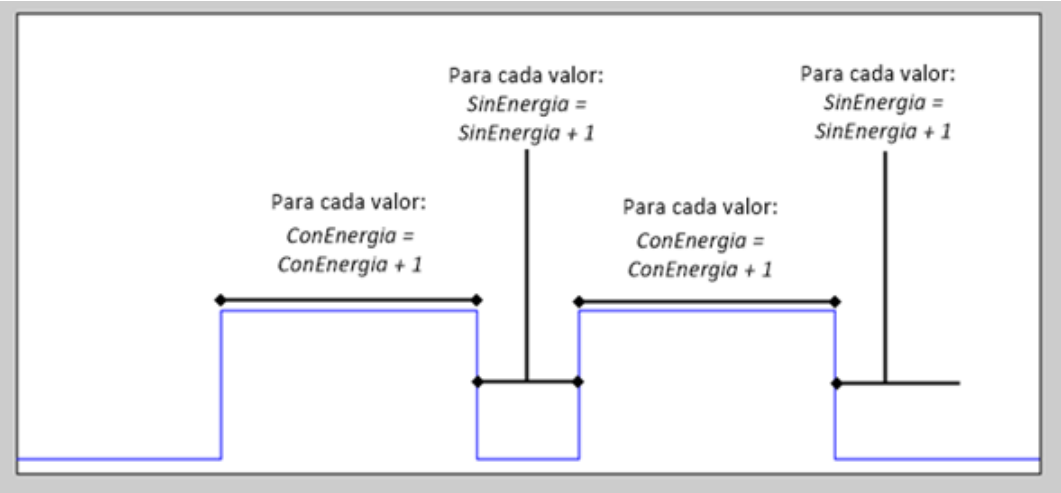


Figura 5.10: Secciones donde se incrementarán ConEnergia y SinEnergia

la palabra habrá en algún momento nuevamente valores con energía, por lo que SinEnergia deberá de ser reiniciado, siempre que no se haya cumplido la condición de silencio mínimo. Ver figura 5.11

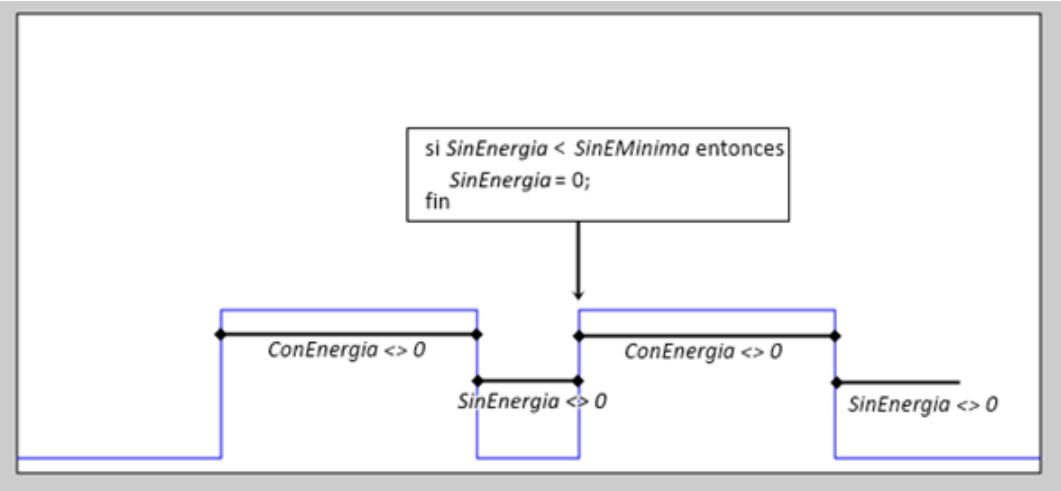


Figura 5.11: Puntos donde se modifica SinEnergia

5. Para encontrar el final de la señal de voz se debe cumplir que ConEnergia supere la constante ConEMinima. Asimismo se presenta una condicionante que involucra a SinEnergia y que puede presentarse en varios casos

   a) Cuando *SinEnergia* sobrepasa la constante *SinEMinima* se considera que el final de la señal de voz (fin) se localizará en la posición anterior al primer valor de silencio detectado. Ver figura 5.12

   b) Cuando se ha terminado de analizar el vector de audio y *SinEnergia* es diferente de cero pero no logra sobrepasar *SinEMinima*, se considera que

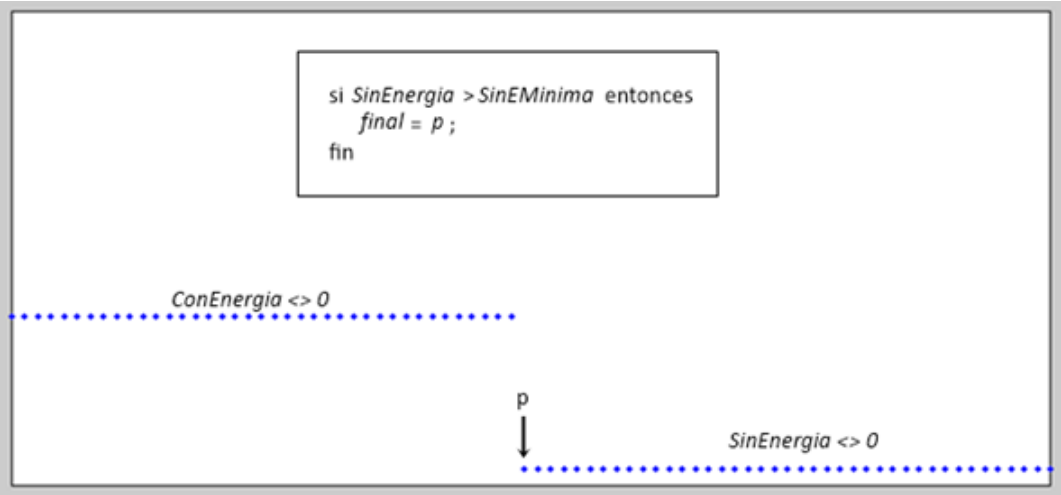


Figura 5.12: Primera forma en que puede localizarse el final de la señal

el final de la señal de voz se localiza en la posición anterior del primer valor de silencio detectado. Ver figura 5.13

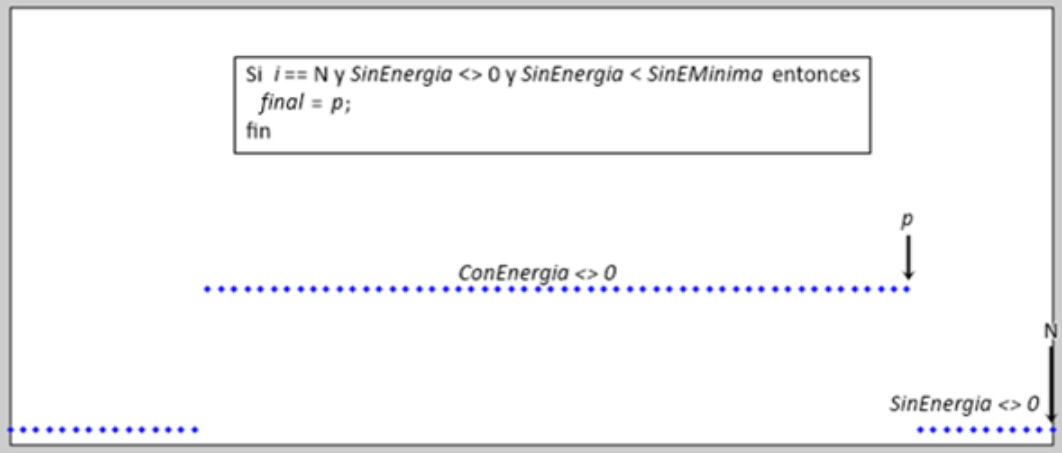


Figura 5.13: Segunda forma en que puede localizarse el final de la señal

c) Cuando se ha terminado de analizar el vector de voz pero *SinEnergia* es igual a cero se considera que el final de la señal de voz está localizado en el mismo final del vector.

6. Una vez que han localizados los índices de inicio y fin de la palabra se pueden copiar los valores dentro de un nuevo vector o eliminar aquellos que se encuentran fuera del mismo.

El algoritmo de este procedimiento se puede representar como se muestra en el diagrama de la figura 5.15

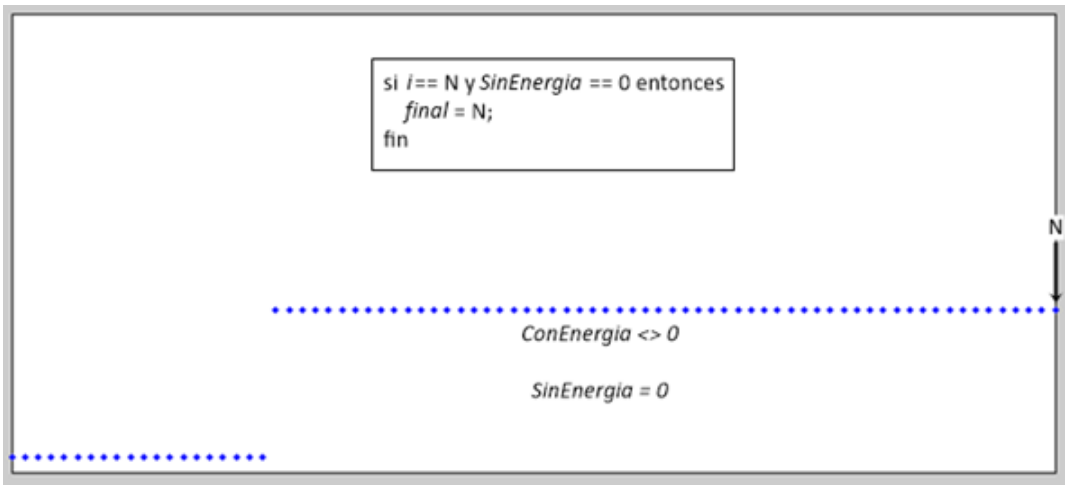


Figura 5.14: Tercera forma en que puede localizarse el final de la señal

El algoritmo analiza el vector de energía de la señal y busca el posible inicio y final de la señal de voz que se contiene. Esto está delimitado por las constantes *SinEMinima* y *ConEMinima*, las cuales son sugeridas por el usuario. Por ejemplo, sea un vector de audio, la cual contiene una señal de voz como se muestra en la figura 5.16

Al discriminar los valores que no cumplen con la energía mínima, se tiene que la información relevante de la señal, es la mostrada en la figura 5.17

Al aplicar el algoritmo para separar la señal con *SinEMinima* equivalente a 400 muestras, se tiene que la señal resultante equivale a la mostrada en la figura 5.18

Debido a que el valor de *SinEMinima* es muy bajo, el algoritmo considera la primera parte de la información aislada como la señal que estamos buscando. Ahora, si se aplica un valor equivalente a 500 muestras se obtendría un resultado como el mostrado en la figura 5.19

Aquí se puede observar que el número de muestras ha disminuido en los extremos de la señal, sin embargo, no ha cambiado mucho. Esto se debe al valor de energía que se empleado para la discriminación. En el ejemplo de la figura 5.1.1 se muestra un segundo ejemplo de la aplicación del algoritmo. Para este caso, se puede observar como la señal separada ha eliminado un número mayor de muestras de sus extremos con ayuda del mismo algoritmo. Esto resalta del hecho de estar manipulando señales aleatorias cuyo comportamiento está en función de los parámetros del momento de la grabación.

El algoritmo de detección se basa en la búsqueda de intervalos cuya energía supere a la del ruido. Esto nos permite actualizar este límite cada vez que se realice una grabación. Sin embargo, el algoritmo aquí presentado solo se encarga de buscar señales con energía que cumpla las condicionantes planteadas. Este algoritmo no determina se la señal que ha encontrado será una señal de voz o cualquier otro tipo de sonido.

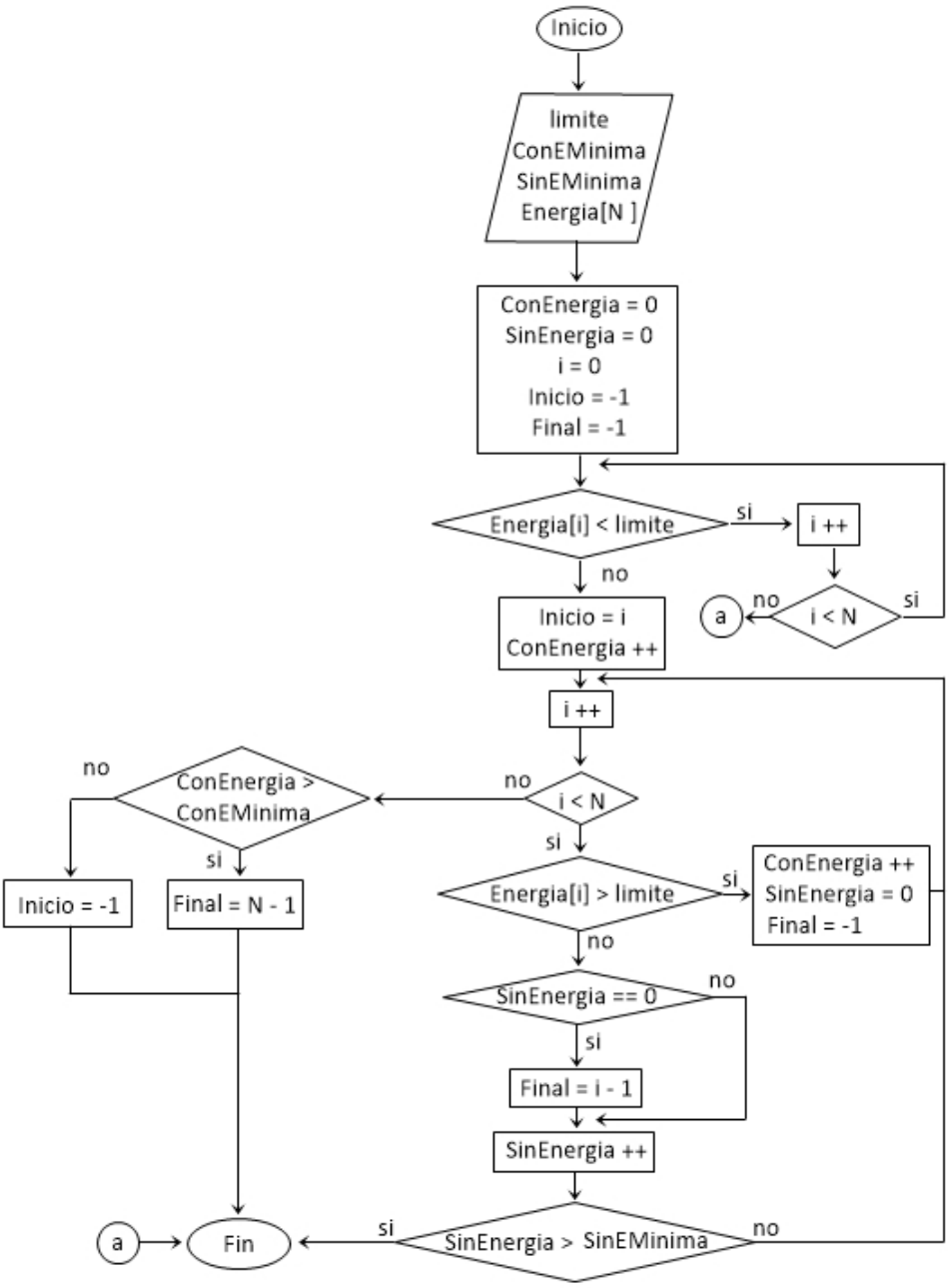


Figura 5.15: Diagrama para la eliminación de zonas de silencio

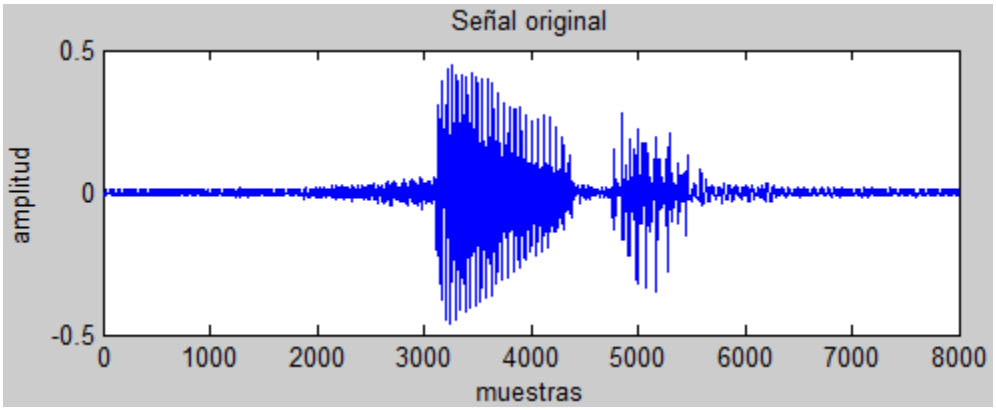


Figura 5.16: Señal en la que se buscará secciones de relevancia

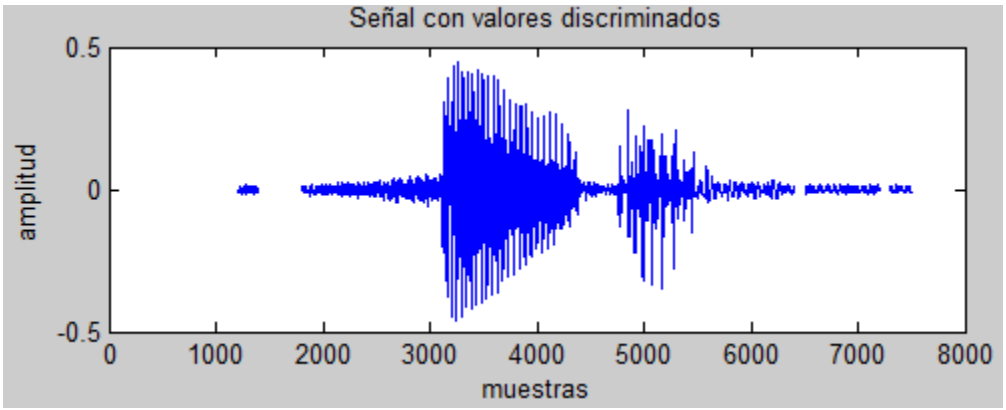


Figura 5.17: Señal discriminada por el algoritmo de búsqueda de información relevante

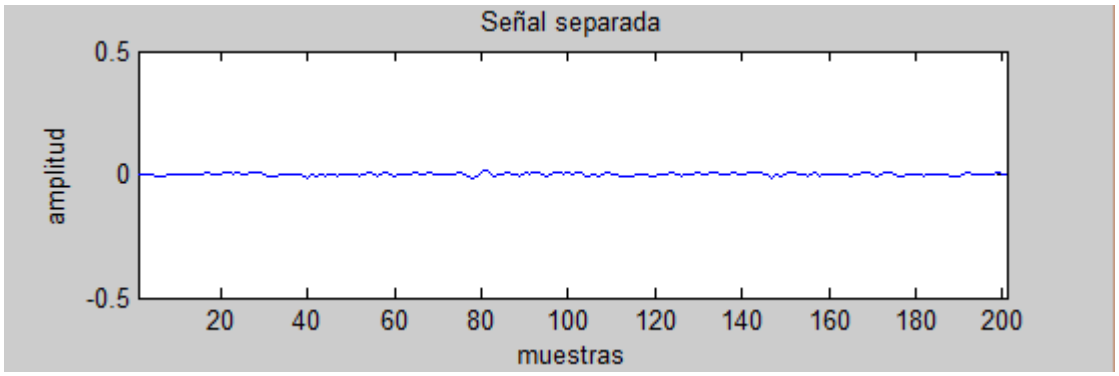


Figura 5.18: Señal discriminada con SinEMinima = 400

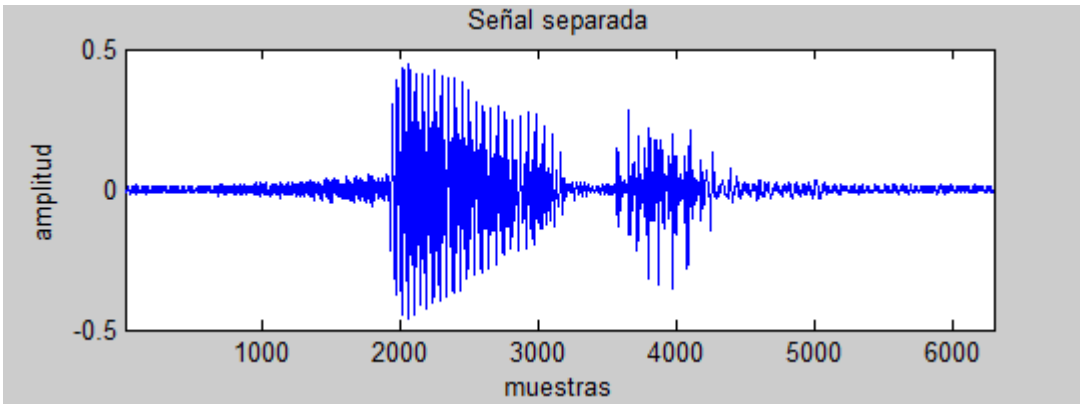
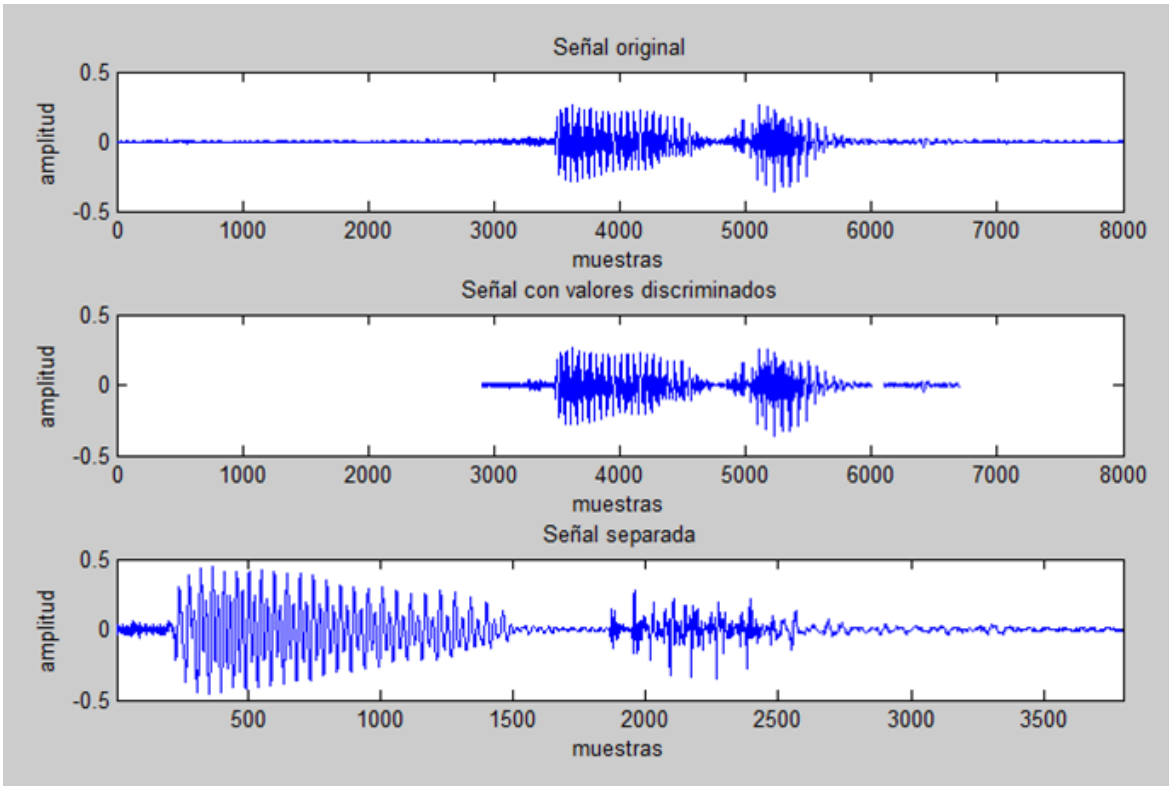


Figura 5.19: Señal discriminada con SinEMinima = 500



Extracción de datos relevantes de una señal adquirida en tiempo real

Cuando se analiza una señal en tiempo real, debe de considerarse que no se sabe en qué momento se susitará alguna señal de voz o que cantidad de ellas se presentarán en este periodo. Esto nos obliga a mantenernos al tanto de las variaciones que se presenten para detectar los principios y finales de las señales, y poder almacenarlas o analizarlas según sea necesario.

Para poder aplicar el algoritmo que se explica en el diagrama de la figura 5.15, se plantea que la adquisición en tiempo real se realice en intervalos fijos. Es decir, que se adquieran valores durante un determinado tiempo y tras culminar este periodo de forma inmediata se adquieran nuevamente la misma cantidad de datos, repitiendo el ciclo hasta que se detenga la adquisición. De esta forma, se obtendrán vectores de tamaño fijo que en conjunto formarán la señal completa y bajo los que se puede aplicar el algoritmo mencionado. Sin embargo, al realizar las particiones no se puede considerar que la señal de voz quedara en medio de alguna de estas particiones. Por lo que, se debe de asumir que la señal puede quedar repartida en más de una división. En el ejemplo que se muestra en la figura 5.20, se presenta una simulación de cómo se verían los vectores colocados de forma consecutiva, donde se han adquirido con intervalos de medio segundo o 4000 muestras.

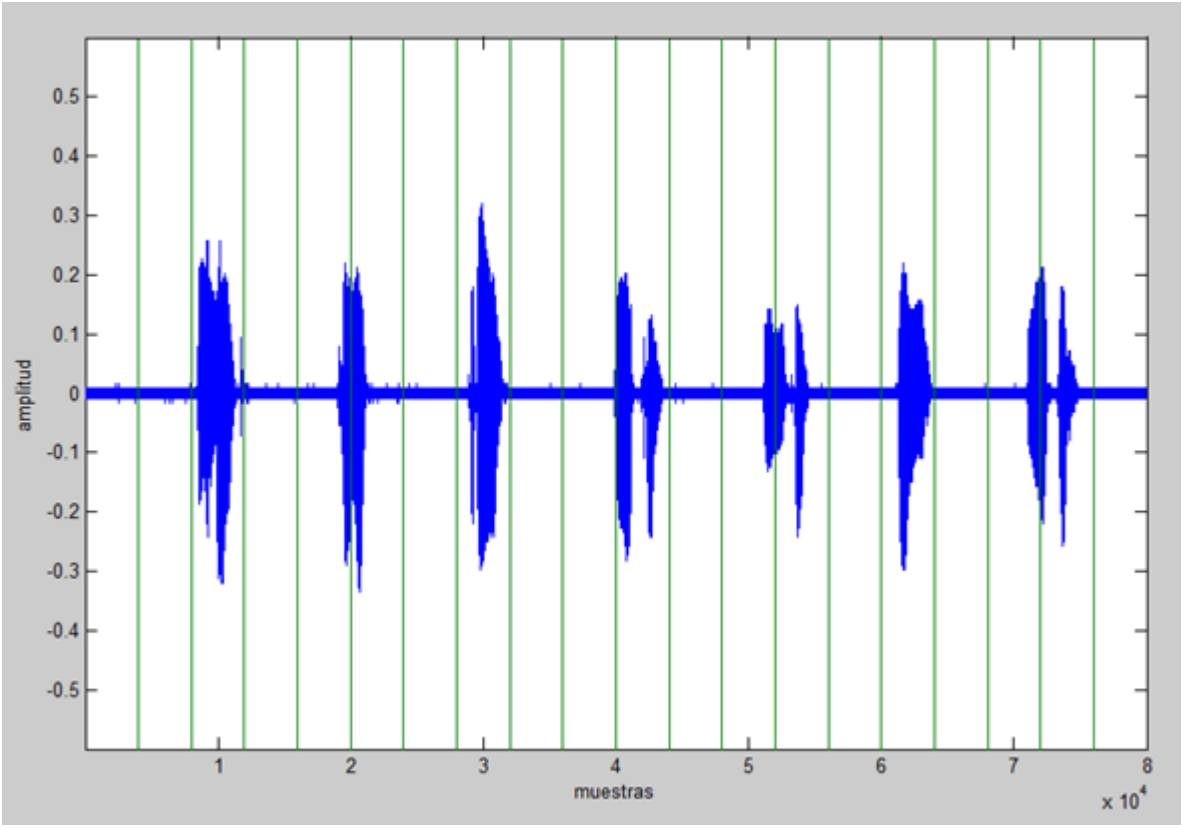


Figura 5.20: Segmentos de la captura en tiempo real de una señal

El ejemplo de la figura 5.20 nos muestra que en algunos casos se podría considerar que una sola división logra englobar a toda la señal. En otros casos se observa que una misma señal está repartida en más de una división consecutiva. Esto implica el tener que considerar que las divisiones consecutivas no son independientes entre sí, mientras que se considere que se tiene una señal de voz.

El algoritmo que se propone resuelve este último problema al almacenar en un vector de tamaño variable la información de la señal de voz. Este comenzará a llenarse con las secciones de información que sean encontradas después del inicio y se mantendrá activo hasta que se encuentre su final. Una vez que se finalice la identificación el vector deberá ser vaciado en espera de una nueva perturbación.

En las figuras 5.21 y 5.22 se presenta el diagrama de flujo en el que se engloba el algoritmo propuesto para la identificación de la señal de voz dentro de una grabación continua.

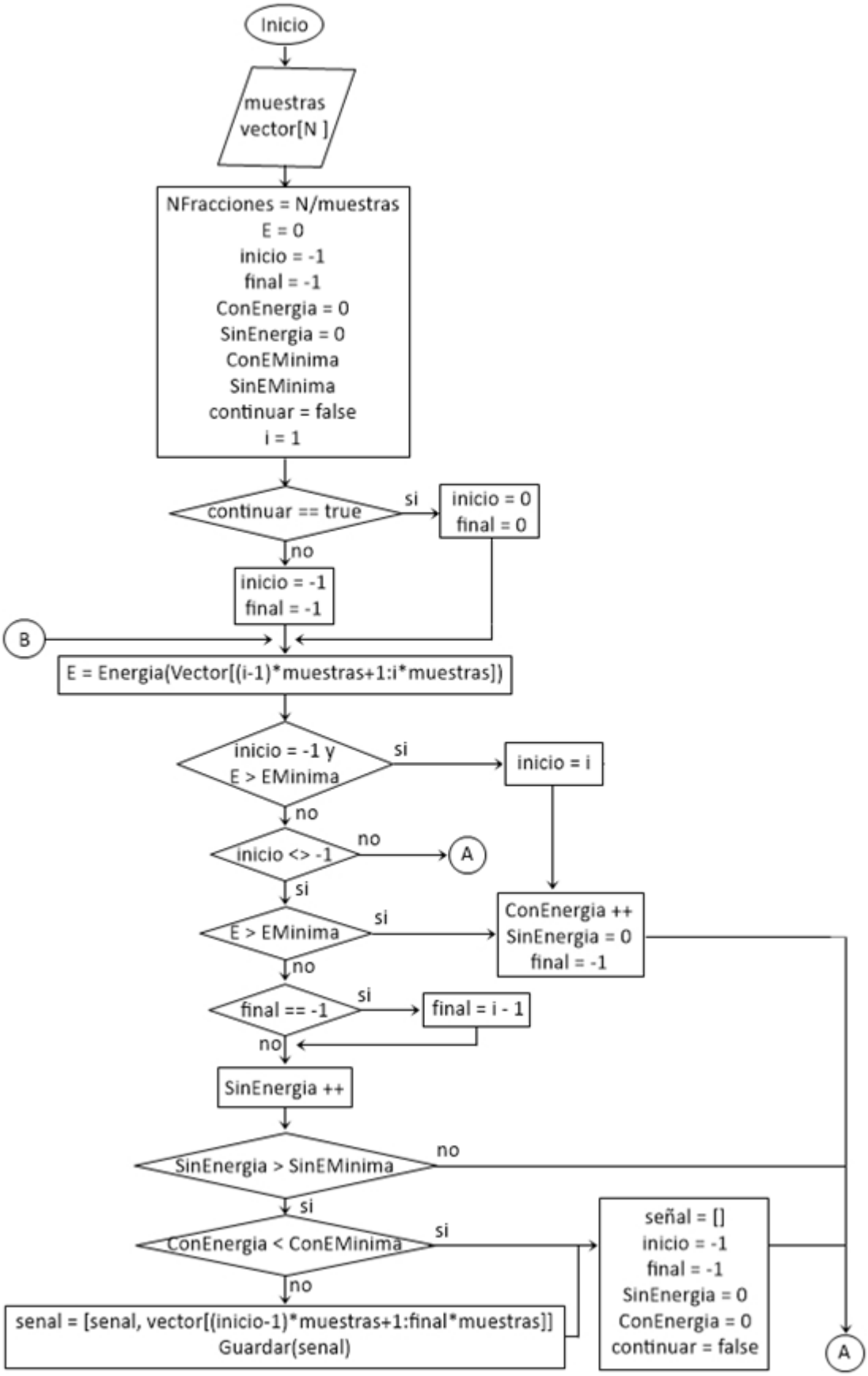


Figura 5.21: Algoritmo para adquirir audio en tiempo real y extraer una señal relevante (Parte 1/2)

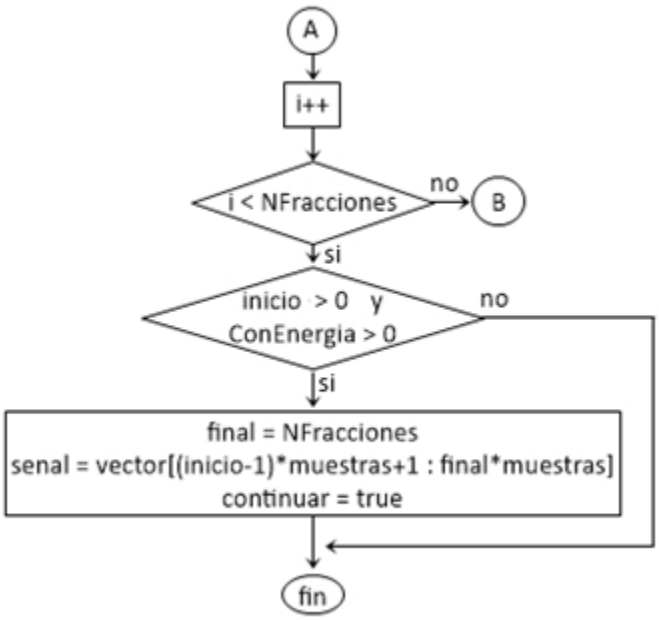


Figura 5.22: Algoritmo para adquirir audio en tiempo real y extraer una señal relevante (Parte 2/2)

El algoritmo maneja como entrada la variable *muestras*, en donde se especificará la cantidad de valores a considerar para obtener la energía. Asimismo, se recibe un vector de tamaño N en el que se contiene la información a analizar, equivalente a una sección de la señal que está siendo adquirida.

Posteriormente, se calcula una constante llamada *NFracciones* que indicará en cuantos segmentos se dividirá el vector de entrada y de los cuales se obtendrá la energía. Para esta última se recomienda que *muestras* sea un valor que al dividir entre N dé como resultado un entero.

También se inicializa la variable en la que se almacenará la energía (E) y las variables que indicarán el inicio y fin de la señal relevante (*inicio* y *final*, respectivamente). Estas dos últimas se inicializan con un valor negativo, esto como indicativo de que no han sido modificadas.

Se emplea como bandera la variable booleana *continuar*, la cual nos permitirá saber si dos secciones recibidas (*vector*) de la señal son dependientes (*true*) o si se pueden considerar independientes (*false*). Cuando se tiene que *continuar* a *inicio* y *final* se les asigna el valor de cero ya que la información podría estar en el principio del nuevo conjunto de datos a analizar, y que el final también podría localizarse en este punto.

El siguiente paso hace referencia al algoritmo planteado en el diagrama de la figura 5.21. Se calcula la energía en función de la variable *muestras*. Si la energía no es aceptable en esta primera iteración, se seguirán analizando los siguientes conjuntos de valores. Cuando se localice algún conjunto que logre sobrepasar el valor de *EMinima* se localizará

en este punto a la variable *inicio*, y se incrementará el contador *ConEnergia*, además de poner el valor de cero a *SinEnergia*.

Tras haber localizado el primer conjunto aceptable, y si sus precursores también lo son, por cada nueva iteración se seguirá incrementando *ConEnergia* hasta localizar nuevamente conjuntos no aceptables, en cuyo caso se dará incremento al contador *SinEnergia*. Cuando se registra el primer valor con baja energía o silencio, se considera que la posición anterior a este podría corresponder al final de la señal.

Cuando se comienza el incremento de *SinEnergia* se evalúa el momento en que se sobrepasa la constante *SinEMinima*, en cuyo caso se evalúa si *ConEnergia* satisface a *ConEMinima* y de ser afirmativo se almacenarán los valores en el vector llamado *senal*. En caso contrario, se les asignarán sus valores iniciales a las variables *senal*, *ConEnergia*, *SinEnergia*, *final* e *inicio*.

Para el caso en que se han encontrado conjuntos de silencio y posteriormente se encontrarán valores con energía, se reinicia el contador de *SinEnergia* y *final*, mientras que se sigue incrementando el contador *ConEnergia*.

Para el caso de haber encontrado un inicio pero no un final, y que se hubiera analizado todo el vector, se realiza el almacenamiento de los valores desde donde indica *inicio* hasta el último de *vector*. También se le da el valor de *true* a la variable *continuar* y se finaliza el algoritmo. De esta forma, la información de la señal permanecerá almacenada en el vector *senal*, complementados con la contenida en el siguiente conjunto de valores a analizar y su reinicio se dará tras haber encontrado un final en las muestras futuras.

A diferencia del algoritmo del diagrama de la figura 5.21, este es capaz de eliminar los pequeños conjuntos de datos con energía que se presentan de forma aislada en la señal. Evitando de esta forma el incluir información sin relevancia.

Sin embargo, una de las desventajas que se podrían presentar corresponde a la presencia de periodos de ruidos o perturbaciones muy extensos. Esto provocaría que nuestro valor de energía mínima fuera despreciable y se comenzarían a almacenar una cantidad indefinida de datos. Por esta razón, se recomienda el colocar un indicador que sugiera al usuario la cantidad de valores que se han almacenado, y, en dado caso, poder detener el algoritmo y ajustar el valor límite.

5.1.2. Discriminación de parámetros LPC por medio de la distancia euclidiana

La distancia euclidiana proporciona un valor de similitud entre dos puntos en el espacio. Al aplicarse a un par de señales, es posible determinar si estas son parecidas o no. Asimismo, es posible determinar qué conjunto de puntos de la segunda señal tienen mayor relación con la primera.

La distancia euclidiana para dos puntos en el plano cartesiano XY queda definida por la ecuación 5.1 [10]. Esta expresión maneja los valores de dos conjuntos de coordenadas bidimensionales para determinar la distancia (d) más corta existente entre ellos. El cálculo realizado corresponde a la hipotenusa del triángulo rectángulo que se forma.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5.1}$$

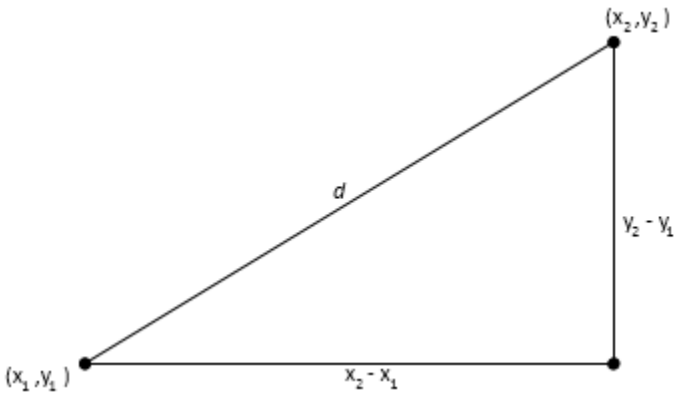


Figura 5.23: Forma gráfica de la ecuación de distancia Euclidiana entre dos puntos

De forma similar, cuando se majean más de dos valores por punto, la distancia es calculada sacando la raíz cuadrada de la suma de las diferencias cuadráticas de cada pareja, como se expresa en la función 5.2 [10]. Esta distancia tomará el valor de cero si los dos puntos son los mismos, y su magnitud aumentará en proporción a la separación entre ellos.

Dado los puntos del espacio de k dimensiones:

$$P = p_1, p_2, ..., p_k$$
$$Q = q_1, q_2, ..., q_k$$

$$d = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_k - p_k)} \tag{5.2}$$

Los parámetros LPC pueden ser considerados como un conjunto de puntos en un espacio multidimensional que representan una parte especifica de la señal. Cuando se comparan dos señales, se trata de buscar aquellos puntos de la segunda señal que se parezcan a los de la primera. Para lograr esto, se debe de comparar cada uno de los puntos de la primera con todas las posibilidades de la segunda. Una forma de visualizar las distintas distancias es a través de una tabla como se muestra en el cuadro 5.1

P(n)	$d_{n,1}$	$d_{n,2}$	$d_{n,3}$	...	$d_{n,m}$
...	...	...	...	...	...
P(3)	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	...	$d_{3,m}$
P(2)	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	...	$d_{2,m}$
P(1)	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	...	$d_{1,m}$
	Q(1)	**Q(2)**	**Q(3)**	...	**Q(m)**

Cuadro 5.1: Distribución de distancias de dos conjuntos de puntos

Con la estructura de esta tabla se pueden visualizar el total de las combinaciones posibles de las dos señales. Se podrán apreciar las variaciones de las combinaciones de puntos del conjunto de parámetros $Q(i)$ de la primera señal con el conjunto de parámetros $P(i)$ de la segunda señal. Esta variación puede ser apreciada con mayor facilidad cuando a cada valor se le es asignado un color. Por ejemplo, cuando se comparan dos señales exactamente iguales se tendrá un comportamiento como el mostrado en la figura 5.24. Como se puede observar las magnitudes más bajas están representadas por un color intenso de azul, formando una diagonal que inicia en la combinación (0,0) y termina en la combinación (m,n).

Esté método nos permite estimar fácilmente las distancias mínimas para asociar cada conjunto de parámetros, formando un camino único que cruza el diagrama. Sin embargo se pueden presentar algunas señales en las que el camino que formen no se defina claramente o existan más de uno. Por ejemplo, cuando se poseen un par de señales como las expresadas en la figura 5.25, se tendrán una gran cantidad de distancias con magnitudes bajas, como se muestra en la figura 5.26. Esto implica que se puedan tomar distintos caminos para realizar las asociaciones.

Cuando se manejan distancias euclidianas, se deben de considerar las restricciones locales y las restricciones globales. Estas facilitan el cálculo y la comparación de las señales en este método al imponer determinadas restricciones.

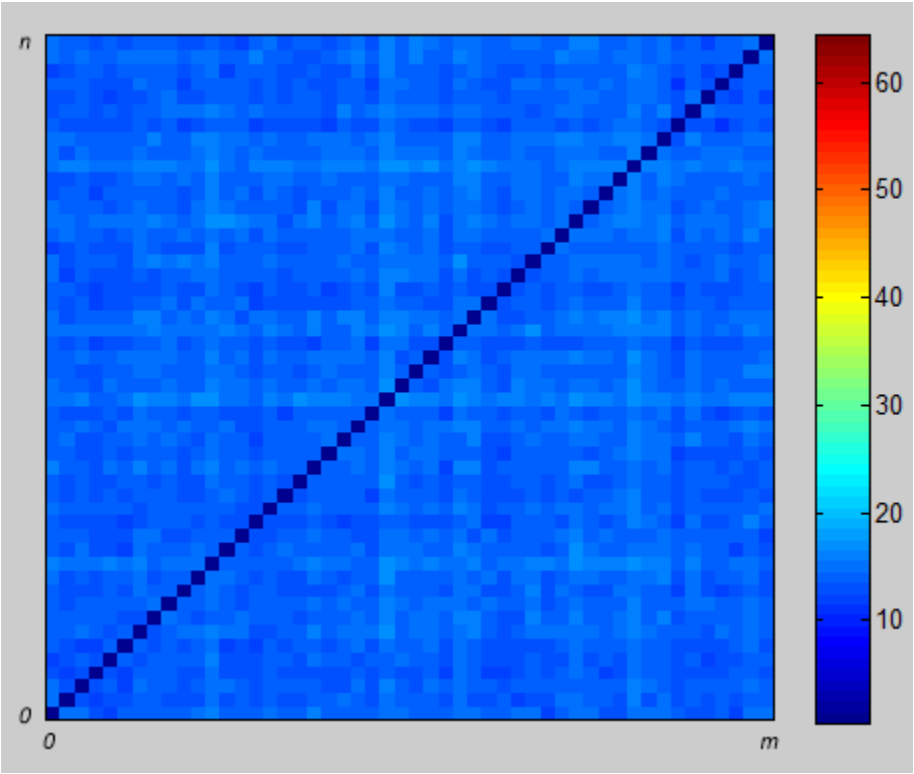


Figura 5.24: Representación gráfica de la distancia Euclidiana de dos señales iguales

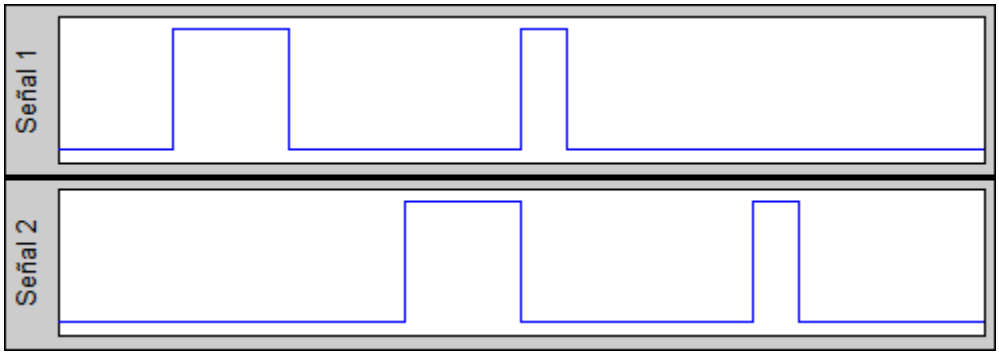


Figura 5.25: Señales con desfasamiento en el tiempo

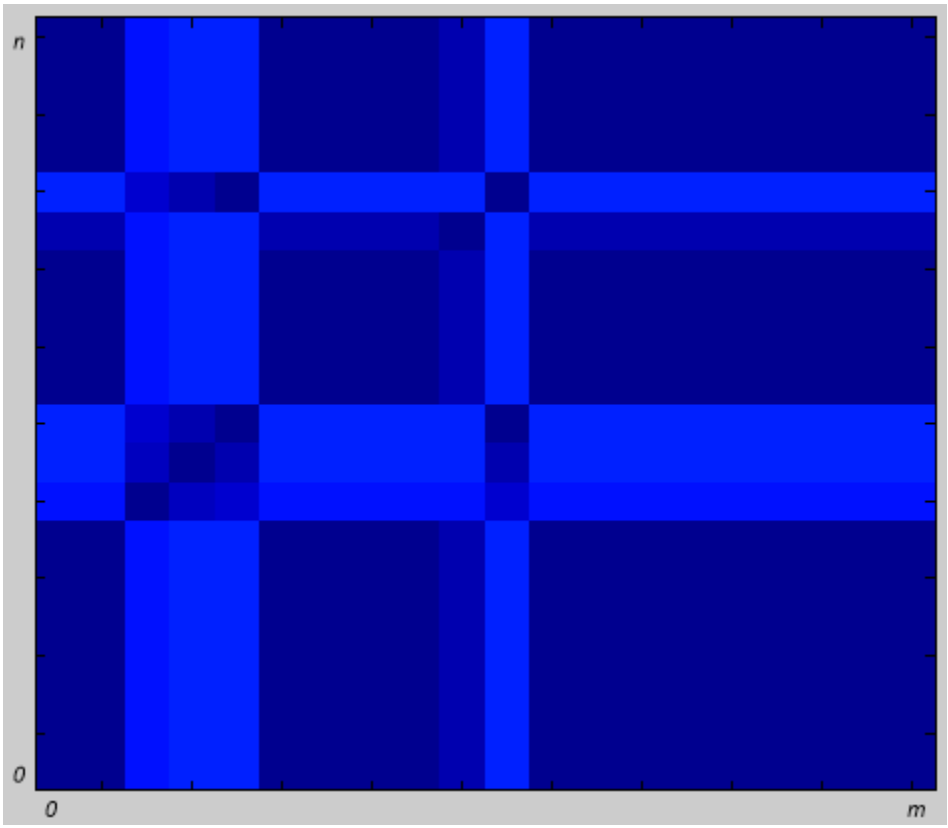


Figura 5.26: Distancia Ecuclidiana de dos señales con desfasamiento temporal

Restricciones locales

Para poder establecer el camino con mayor relevancia, se suele implementar el método *Dinamic Time Warping* (DTW). El método calcula las distancias acumuladas de la matriz, la cual consiste en la suma del valor de la distancia actual y el mínimo valor de la distancia acumulada anterior. Existen distintos métodos para obtener la distancia acumulada, conocidos como restricciones locales. Estos se basan en distintas combinaciones de los valores acumulados pasados. En la figura 5.27 se pueden apreciar distintas posibilidades para calcular dicha distancia acumulada [17].

Los valores obtenidos con la distancia acumulada resaltan los valores de distancia con menor magnitud o intensidad. Esto permite poder visualizar con una mayor facilidad aquel camino que puede ser de utilidad para asociar los parámetros. En la figura 5.28 se ejemplifica la aplicación del primer caso de cálculo de distancia acumulada mostrado en la tabla de la figura 5.27 haciendo referencia al ejemplo mostrado en 5.25.

Como se puede apreciar el camino que se genera mantiene un color azul intenso, mientras que los valores en sus extremos adquieren colores asignados a valores de mayor intensidad. Sin embargo, aún se pueden tener distintos caminos dado que se tienen presentes diversos grupos de color azul. Estos indican que las combinaciones de puntos son similares para distintos momentos y por lo tanto pueden ser asociados a uno sólo valor.

En la figura 5.29 se puede apreciar la elección de uno de los caminos para asociar los valores de la segunda señal con los de la primera. Se muestra la asociación de más de un punto de una señal con uno solo de la segunda. En este ejemplo se puede destacar que para aquellas partes donde la señal es relevante se forma un camino de un mismo color que actúa como una diagonal, mientras que en el resto son rectas horizontales o verticales.

Restricciones globales

Al emplear la distancia euclidiana y la distancia acumulada para comparar señales se desea que las dos señales, cuando son similares, tengan un comportamiento similar al que se muestra en la figura 5.23. Gracias a esta consideración, se pueden aplicar un conjunto de límites que discriminen algunos de los valores de los extremos superior-izquierda e inferior-derecho. A estos límites se les conoce como restricciones globales y permiten discriminar señales que son diferentes con una mayor facilidad.

Cuando las distancias se localizan fuera de estos límites se puede considerar que las señales son diferentes ya que no se está formando una diagonal. En la figura 5.30 se pueden apreciar las restricciones globales y sus respectivas ecuaciones [17].

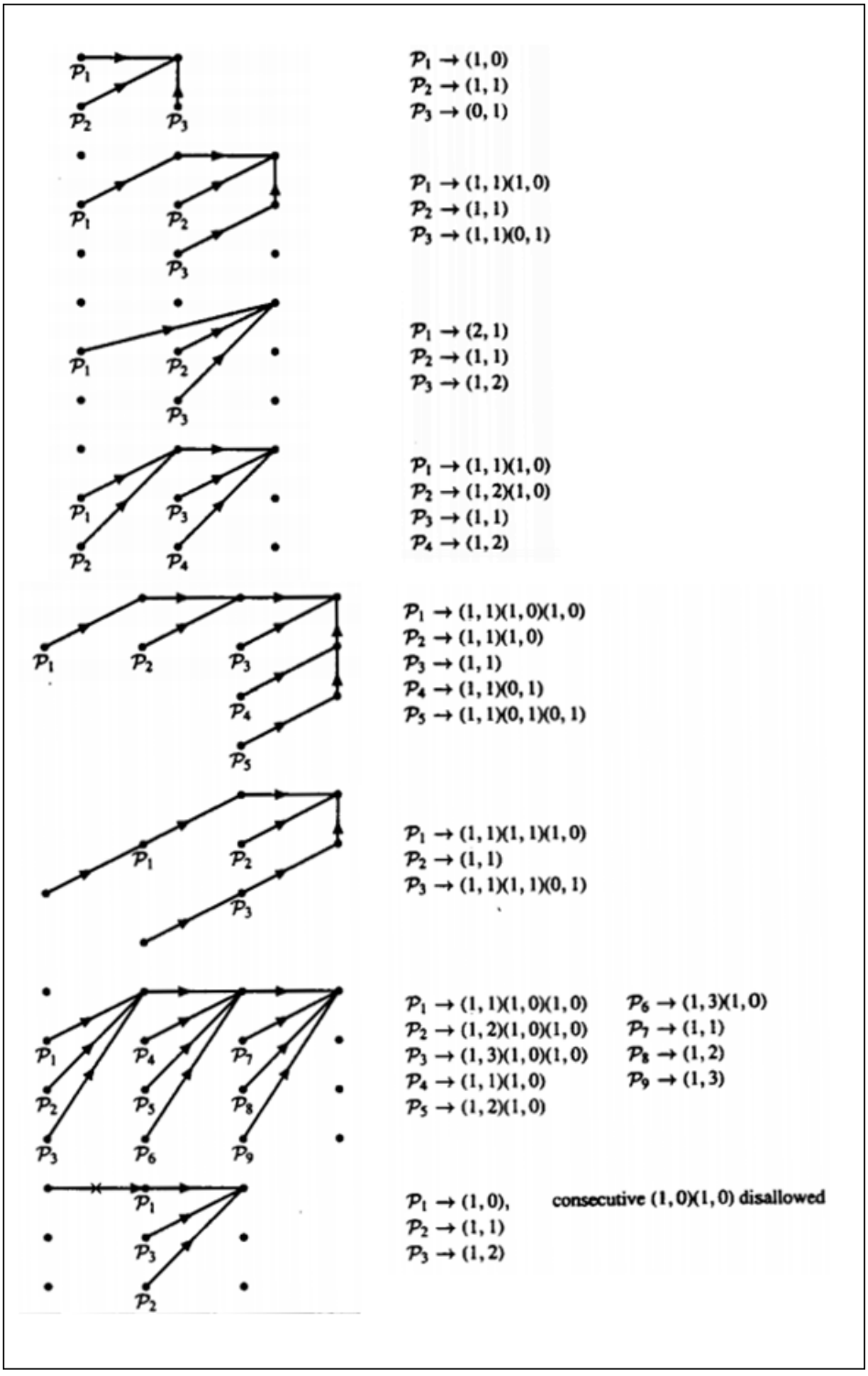


Figura 5.27: Formas de calcular la distancia acumulada [17]

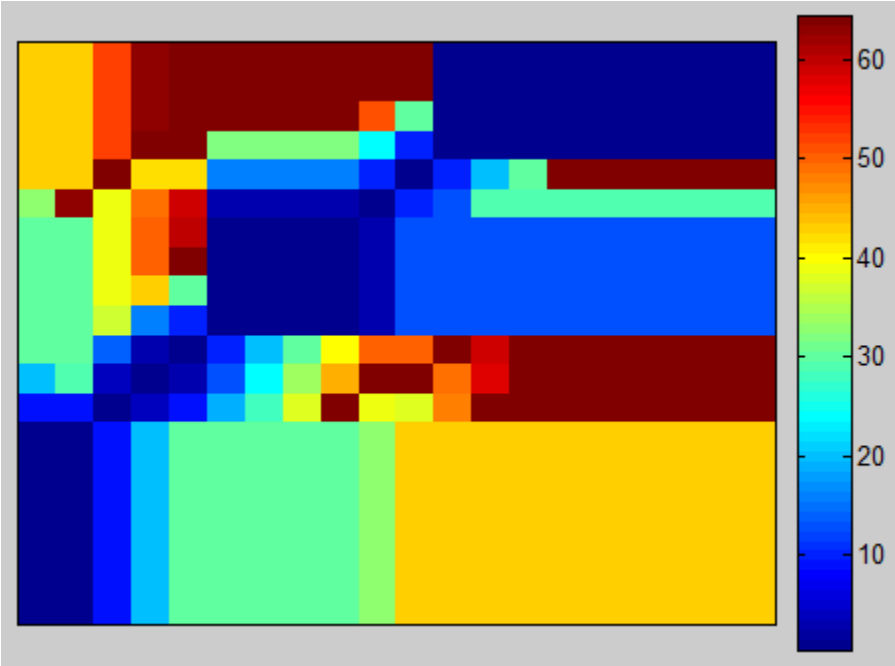


Figura 5.28: Distancia acumulada de dos señales con desfasamiento temporal

Algoritmo para la asociación de parámetros por medio de la distancia euclidiana

Para relacionar los parámetros LPC de una señal de voz adquirida con una señal de voz de referencia se realiza el cálculo de distancias acumuladas. El algoritmo propuesto se basa en la restricción global definida por la primera opción de la figura 5.27 [17]. Por lo tato, el cálculo de la distancia acumulada se define de la siguiente forma.

Sea x una matriz que contiene el conjunto de parámetros de la señal base e y una matriz que contiene los parámetros de la señal que se desea comparar. Se sabe que para todos los puntos de las dos señales estos estarán compuestos de un número igual de componentes, es decir, el número de parámetros LPC calculados para cada segmento de la señal serán los mismos. La distancia acumulada queda definida las ecuaciones 5.3 y 5.4.

$$d[i][j] = \sqrt{\sum_{k=1}^{na}(x[i][k] - y[j][k])^2} \tag{5.3}$$

$$D[i][j] = min \left\{ \begin{array}{c} d[i][j] + D[i-1][j], \\ d[i][j] + 2D[i-1][j-1], \\ d[i][j] + D[i][j-1] \end{array} \right\} \tag{5.4}$$

Donde na es el total de parámetros LPC, y la función $min(a, b, c)$ regresa el valor entero con la menor magnitud.

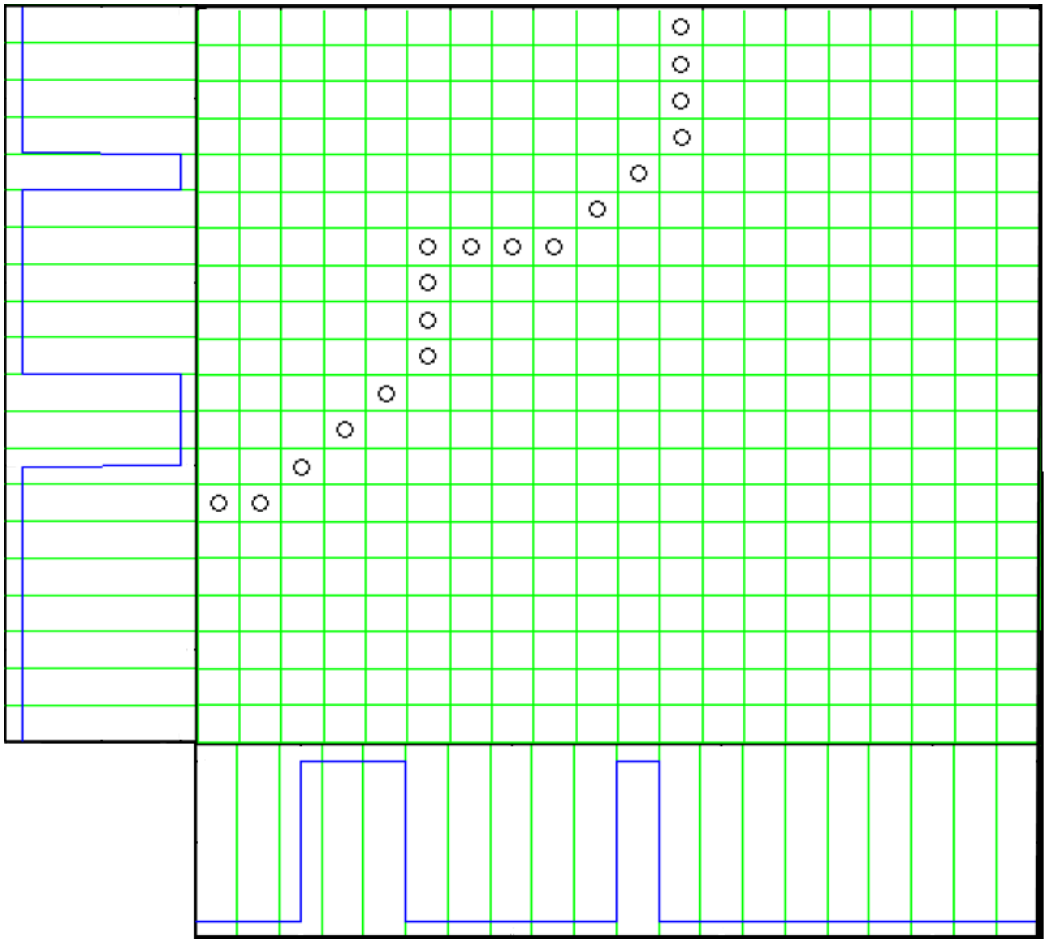


Figura 5.29: Camino calculado a partir de la distancia acumulada

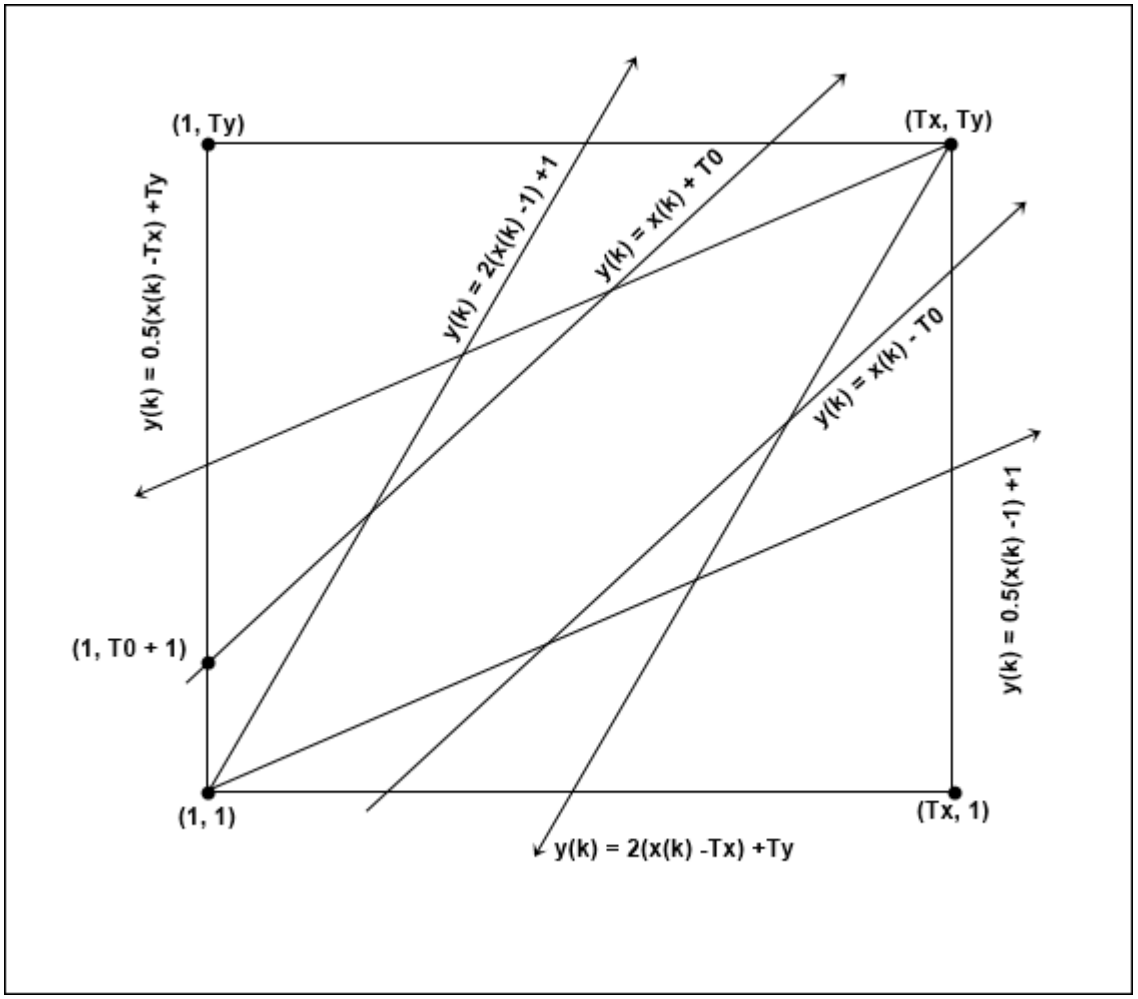


Figura 5.30: Restricciones globales

La restricción global que se considera para este algoritmo se basa en un par de rectas posicionadas como se observa en la figura 5.31. Estas dan la posibilidad de que el inicio de la segunda señal no sea necesariamente su primer punto.

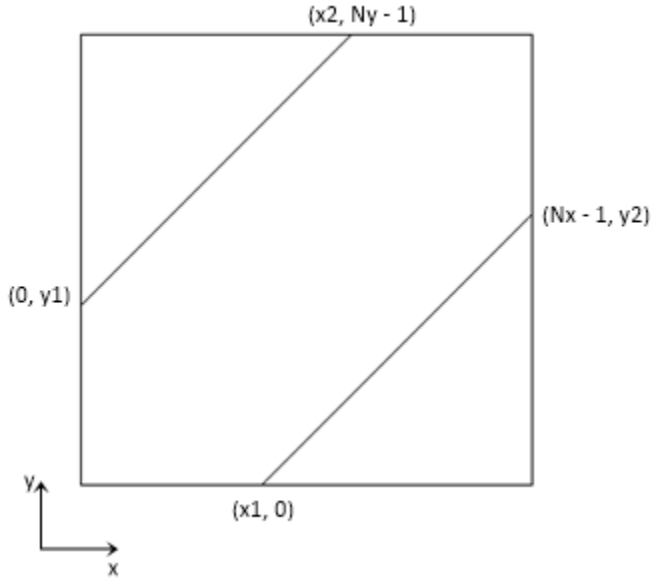


Figura 5.31: Restricción global a usar

Los límites se obtendrán calculando las ecuaciones de cada una de las rectas. Para la línea inferior se puede realizar el cálculo por medio de la ecuación 5.5

$$
\begin{aligned}
y[k] &= mx[k] + b \\
m &= \frac{y2}{Nx - x1 - 1} \\
b &= -m(x1) \\
y2 &= Ny - y1 \\
\therefore y[k] &= \frac{Ny - y1}{Nx - x1 - 1}(x[k] - x1)
\end{aligned} \tag{5.5}
$$

Para la línea superior se considera que el valor de entrada corresponderá al valor de y[k]. En la ecuación 5.6.

$$
\begin{aligned}
x[k] &= my[k] + b \\
m &= \frac{x2}{Ny - y1 - 1} \\
b &= -m(y1) \\
x2 &= Nx - x1
\end{aligned}
$$

$$\therefore x[k] = \frac{Nx - x1}{Ny - y1 - 1}(y[k] - y1) \tag{5.6}$$

Por lo tanto, para que una distancia sea aceptable su posición deberá de estar dentro del rango establecido por los límites anteriores. Es decir que para un punto coordenado $(x[k], y[k])$ se debe de cumplir que

$$x[k] > \frac{Nx - x1}{Ny - y1 - 1}(y[k] - y1)$$
$$y[k] > \frac{Ny - y1}{Nx - x1 - 1}(x[k] - x1)$$

Donde los valores de $x1$ y $y1$ pueden ser definidos como una fracción del total de datos para cada vector. Dicho coeficiente variará entre cero y uno.

$$x1 = cNx$$
$$y1 = cNy$$

Considerando estas restricciones locales es posible dar origen al algoritmo mostrado en el diagrama de la figura 5.32. En este se puede observar el cálculo de la matriz de distancias acumuladas D. Este método requiere de la función $sqrt(x)$ encargada de obtener la raíz cuadrada del valor de entrada x. Asimismo se tiene una función encargada de llenar una matriz con el valor especificado, $Llenar(Matriz[][], Valor)$. Donde $Matriz[][]$ corresponde a una matriz de dimensiones no definidas. También se requiere de la función $suma(x[], y[])$, quien recibe como entrada dos vectores y da como salida la diferencia cuadrada de cada componente. Donde $x[]$ y $y[]$ son vectores de dimensiones no definidas.

$$suma(x[], y[]) \Leftrightarrow \sum_{k=1}^{na}(x[k] - y[k])^2$$

Empleando la matriz D que proporciona el algoritmo anterior, se puede buscar el camino más óptimo para relacionar los puntos de las dos señales. Para esto se emplean las restricciones globales como se muestra en la figura 5.33

5.2. Obtención de parámetros LPC representativos de una palabra

Para poder realizar el reconocimiento de palabras aisladas se deben de tener almacenados los parámetros que representen las palabras a identificar. Estos valores pueden

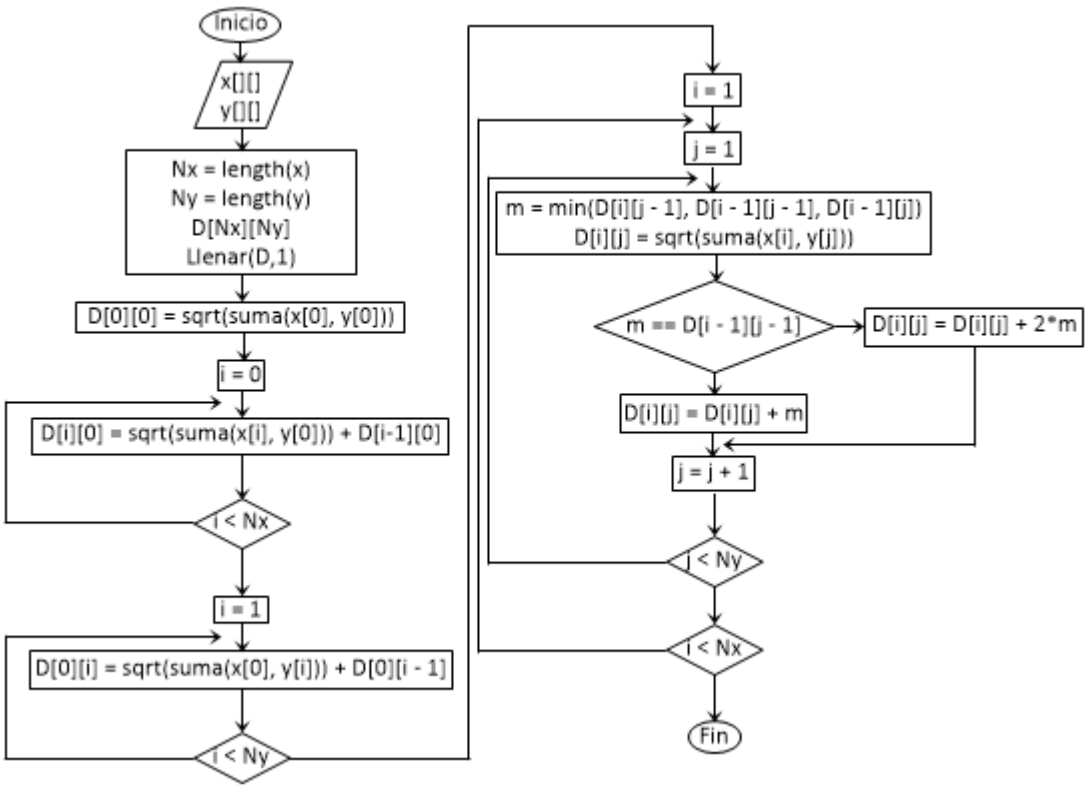


Figura 5.32: Cálculo de las distancias acumuladas

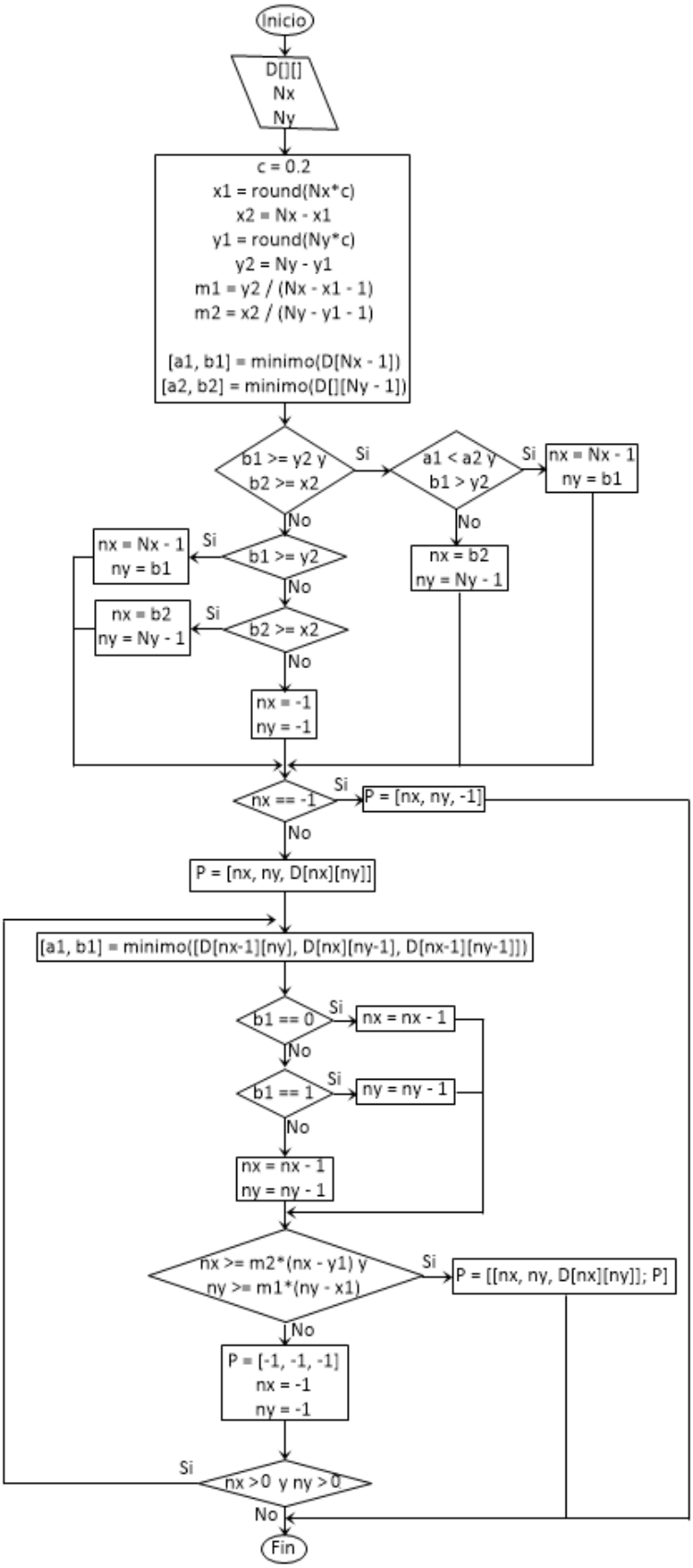


Figura 5.33: Camino óptimo a partir de las Distancias Euclidianas

ser obtenidos comparando distintas señales de una misma palabra y extrayendo de ellos los parámetros con mayor influencia sobre los demás.

Cuando se realiza la comparación de dos señales por medio de su distancia Euclidiana se sabe que la segunda señal será completamente similar a la primera si para cada punto de la señal base se tiene asignado al menos un valor del conjunto de la segunda señal. Por lo tanto, se puede establecer que las señales tendrá un porcentaje de similitud del 100 % cuando la segunda señal pueda cubrir todos los puntos de la primera.

Para obtener aquella señal con la mayor influencia del conjunto, se compara cada señal contra las restantes del conjunto. Por cada comparación se obtendrá un porcentaje de similitud y del total obtenido se calculará el promedio. Este valor representará el nivel de similitud de la señal base actual con el resto del conjunto. Posteriormente, se calcularán los valores de similitud usando como base cada una de las señales restantes.

Se considerará a la señal con el mayor índice de similitud como la señal de mayor influencia. Sin embargo, existe la posibilidad de que algunas señales presenten un valor de similitud muy bajo. Estas señales podrían considerarse como ajenas al conjunto, permitiendo su eliminación. Al realizar este acto se debe de tener en cuenta que habría que calcular nuevamente el nivel de similitud para todas las señales. Este acto de supresión ha demostrado de forma experimental que se mejora el indice de similitud de las señales restantes.

Una vez que se ha localizado la señal con mayor relevancia se procede a calcular nuevamente la distancia Euclidiana. En esta nueva iteración se promedian todos los puntos que se consideren iguales. La matriz que se obtenga tras finalizar el cálculo de las distancias euclidianas corresponderá al conjunto de parámetros asociados a la palabra. En las figuras 5.34 a 5.38 se observa el diagrama que describe el algoritmo para realizar estos cálculos. En este se emplea la función $P[\ ][\ ] = DistanciaE(x[\ ][\ ], y[\ ][\ ])$, la cual recibe como entrada dos matrices con los parámetros de dos señales y proporciona una matriz de dos columnas con la asociación de los puntos de la señal x con los de la señal y.

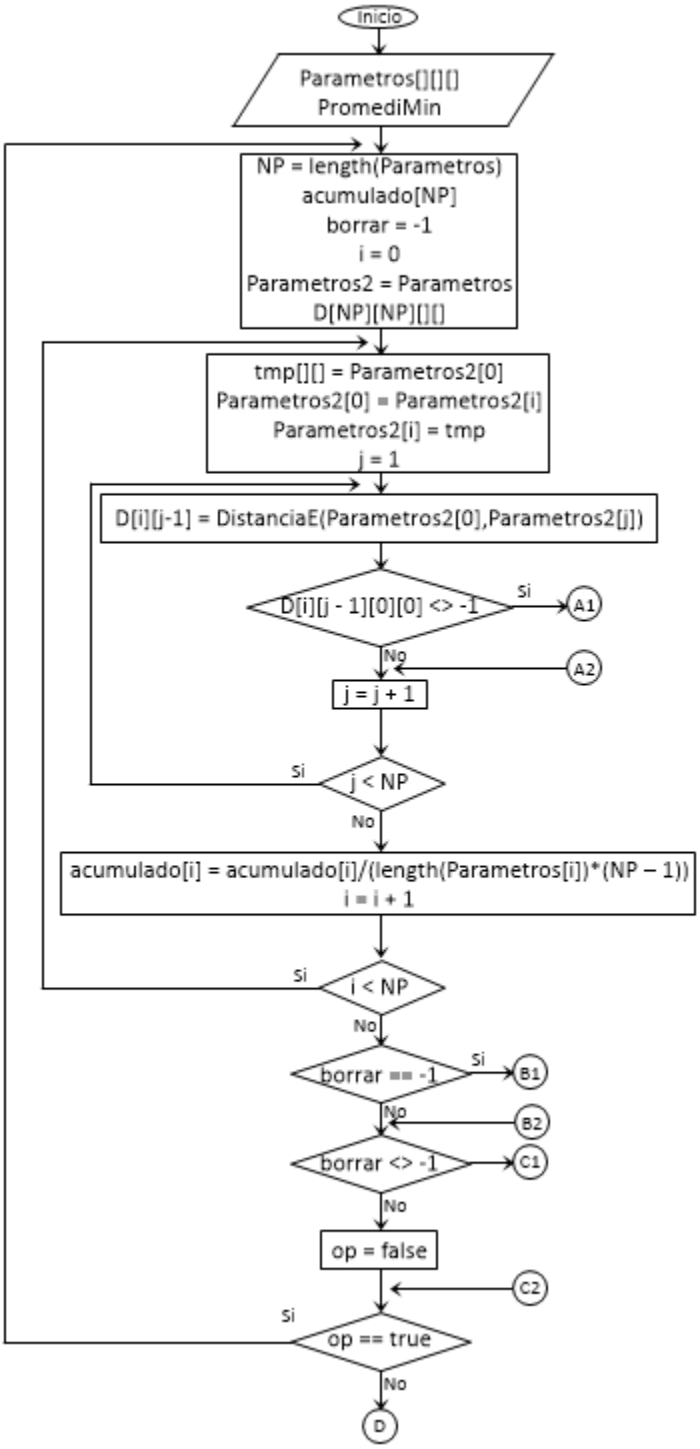


Figura 5.34: Cálculo de parámetros relevantes (Parte 1/5)

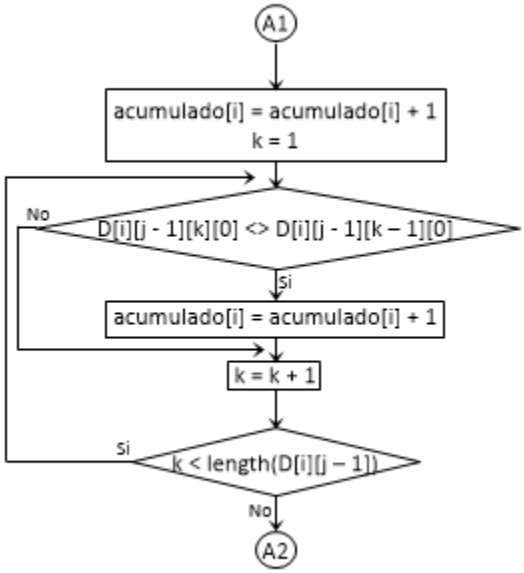


Figura 5.35: Cálculo de parámetros relevantes (Parte 2/5)

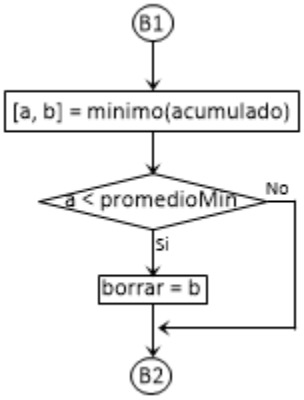


Figura 5.36: Cálculo de parámetros relevantes (Parte 3/5)

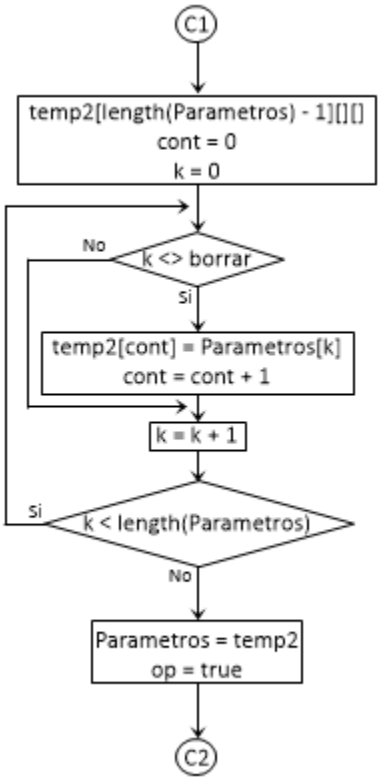


Figura 5.37: Cálculo de parámetros relevantes (Parte 4/5)

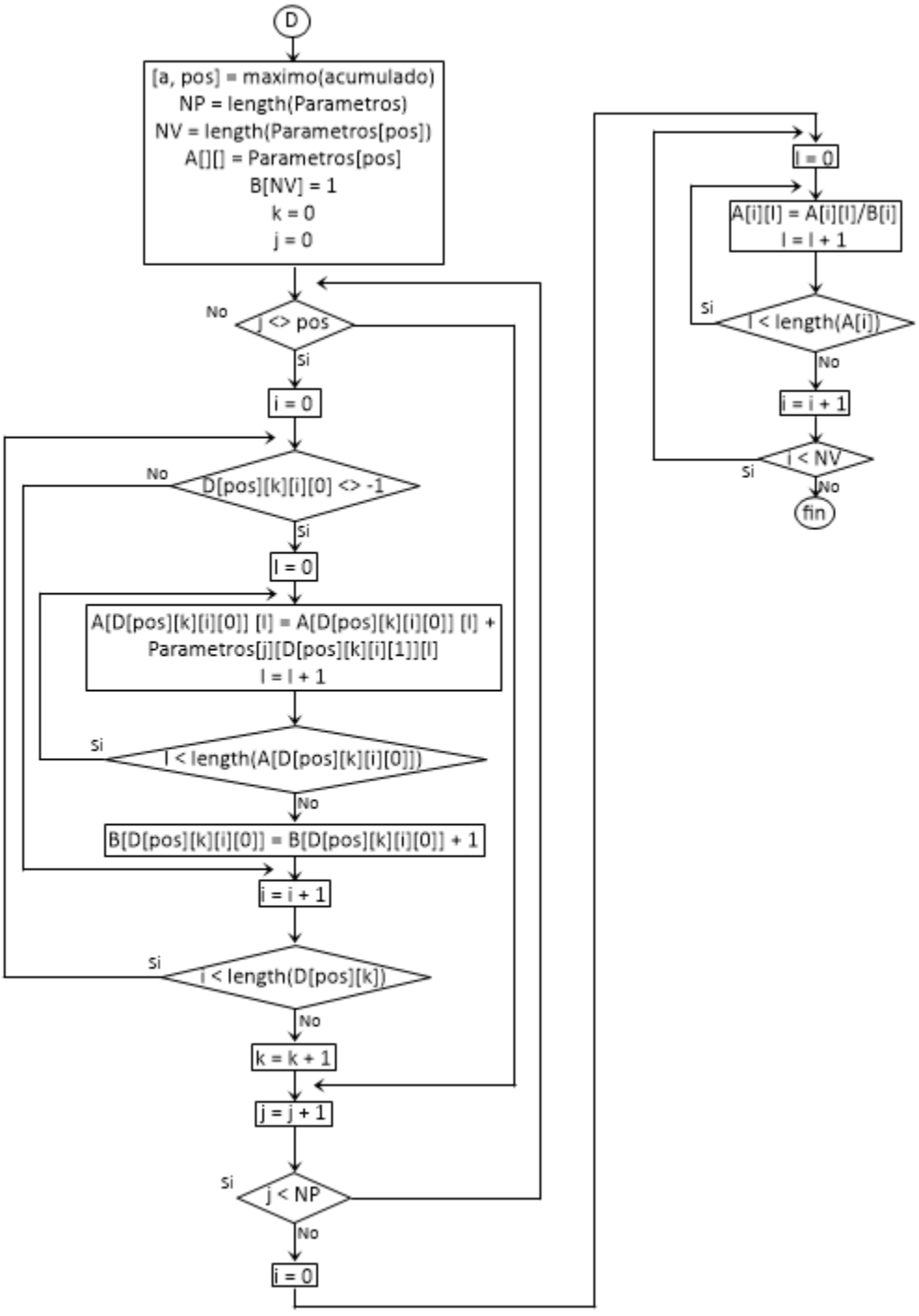


Figura 5.38: Cálculo de parámetros relevantes (Parte 5/5)

5.3. Comparación de parámetros LPC para la detección de palabras

La detección de palabras se basa en la comparación de una señal proporcionada por el usuario contra las diferentes señales almacenadas en la base de datos. Esta se realiza por medio de la distancia euclidiana calculando el número de coincidencias que tiene la señal de entrada con las señales de referencia.

Este método extrae los parámetros LPC de la señal de entrada y los compara con cada conjunto de parámetros contenidos en la base de datos. Por cada comparación realizada se obtiene una matriz con las asociaciones asignadas para ambos conjuntos de puntos. Para cada matriz se calculará el porcentaje de puntos que son asociados de la señal de entrada a la señal de referencia. Los distintos porcentajes obtenidos son comparados para seleccionar aquella señal con mayor similitud.

En el diagrama de la figura 5.39 se puede observar el algoritmo que calcula los porcentajes de asociación entre la señal de referencia y las señales de la base de datos. Esta recibe de entrada tanto la matriz de parámetros LPC de la señal de entrada y la matriz con todos los parámetros que se desean comparar. La información de interés es almacenada en el vector *Parametros* que posee el mismo tamaño que el número de parámetros de la base de datos.

5.4. Implementación del reconocedor de palabras aisladas

Como se ha mostrado en la figura 5.1, el reconocedor de palabras aislada en tiempo real se compone de dos módulos que se ejecutan en paralelo. El primero se encarga de realizar la captura de audio, y si es necesario adecuar los valores al formato utilizado. El segundo está a cargo de buscar dentro de los tramos de audio una señal con suficiente energía para ser analizada, y determinar si es de utilidad.

El sistema que se propone para realizar el reconocimiento de palabras en tiempo real ocupa los distintos algoritmos que se han descrito en secciones anteriores de este mismo documento. A continuación se describe en forma de lista el total del algoritmo propuesto. El primer punto es una operación que se realiza antes de ejecutar el programa de reconocimiento. Los puntos dos y tres son los elementos que corresponden a la adquisición y análisis del audio, respectivamente. La descripción que se realiza de estos dos últimos puntos corresponde a una iteración, para posteriormente ser usadas en el ciclo ya mencionado.

1. 1. Cálculo de la energía del silencio: para poder diferenciar entre una señal de

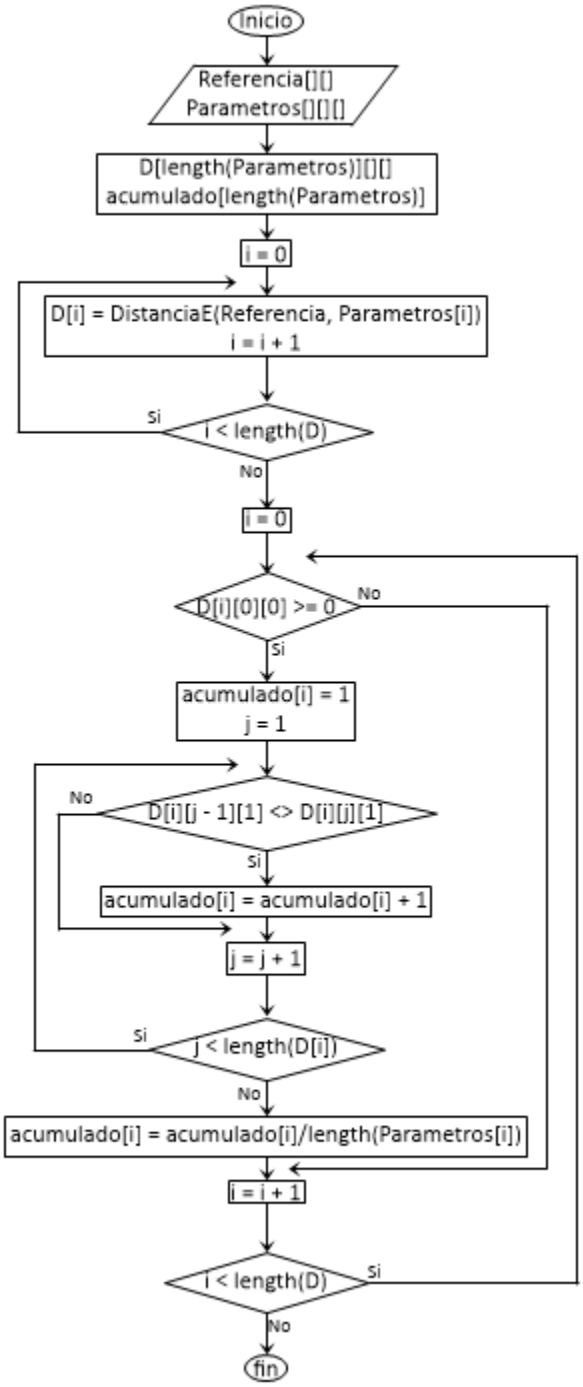


Figura 5.39: Comparación de una señal de entrada contra las contenidas por la base de datos

silencio y una señal de voz se propone el identificar la primera mediante su energía.

   a) Se realiza una adquisición de audio de 1000 datos o 0.125 segundos, considerando que se tratará de silencio.

   b) La señal adquirida deberá variar entre -1 y 1 y será seccionada en fragmentos de 50 datos.

   c) A cada segmento se la calculará su energía, como lo muestra la ecuación 2.3

   d) Del conjunto de energías calculadas se obtendrá el promedio con su tercera desviación estándar. Este valor será considerado como la energía máxima que puede tener el silencio.

2. Adquisición de audio: El audio será adquirido en segmentos de la misma duración y de forma continua. Se finalizará la adquisición en el momento en que se indique que se ha finalizado el algoritmo.

   a) Se adquiere un segmento de audio

   b) Se realizará una conversión de unidades según sean las necesidades. Para este algoritmo se requiere que las señales varíen entre -1 y 1.

   c) Se almacena la señal en una variable ajena a la usada para adquirir el audio

3. Análisis de audio: El proceso de análisis comenzará tras haberse realizado la primera adquisición de audio. Su tiempo de ejecución deberá de ser menor que el tiempo de adquisición de audio del punto 2.

   a) Se busca dentro de los distintos segmentos de sonidos una señal como se muestra en el diagrama 5.21 y 5.22.

   b) Cuando se ha identificado una señal y se ha aislado de la original, se realiza la segmentación en partes de 200 muestras con 80 de traslape, y se la aplica una ventana de Hamming. En cada sección se aplicará el filtro de pre-énfasis.

   c) Para cada una de las secciones de la señal se calcularán los parámetros LPC.

   d) Usando la distancia euclidiana se comparan cada una de las palabras almacenadas en la base de datos. De cada comparación se obtiene un valor que indica el nivel de similaridad entre señales, como se observa en el esquema de la imagen 5.39.

   e) Aquella señal de la base de datos que muestre la mayor similitud será considerada como su igual. Asimismo, si no hay coincidencia se considerará que la señal de entrada no es valida

5.5. Implementación del reconocedor en tiempo real en lenguaje Java

En programación, para realizar más de una actividad en el mismo periodo, se suelen implementar los métodos 'Thread'. Estos son mecanismos por medio de los cuales dos programas simulan una ejecución en paralelo. La función 'Thread' segmenta los hilos, con base en sus líneas de código, e intercala las partes. Posteriormente los ejecuta uno a uno, almacenando el estado de cada programa, para determinar en qué punto ha terminado la sección anterior. Por medio del uso de esta función se puede implementar el sistema de reconocimiento de palabras aisladas en tiempo real. Para ello se ejecutaran dos programas en paralelo, el primero encargado de adquirir la señal la señal desde la tarjeta de sonido, mientras que el segundo realizará los cálculos pertinentes.

Java permite realizar dos operaciones en paralelo por medio del uso de su función *thread*. Esta permite realizar operaciones simultáneas con una duración definida por el proceso más tardado. Asimismo, Java permite adquirir audio desde la tarjeta de sonido. En el anexo A se describe el uso de las librerías de Java para realizar la adquisición del audio, también se propone un programa para realizar dicho proceso. De la misma forma, en el anexo B se presenta el uso de la función de Java *thread*.

A continuación se explica la implementación del algoritmo desarrollado en la sección anterior. Se describe la aplicación, en el lenguaje de programación Java, de los distintos pasos que se han propuesto (código fuente completo: [19]).

1. Cálculo de la energía del silencio

```
try{
  // Formato de captura
  format = new AudioFormat(FM,NBits,NC,true,true);
  // Formato de la línea
  info = new DataLine.Info(TargetDataLine.class, format);
  // Generación de la línea
  linea = (TargetDataLine) AudioSystem.getLine(info);
  // Apertura de la linea de captura
  linea.open(format);
  // Captura de 1000 datos para determinar la energía del
  // silencio
  linea.start();
  Datos1 = new byte[1000];
  linea.read(Datos1, 0, Datos1.length);
  linea.stop();
}catch(LineUnavailableException ex){}

double bitMaximo = Math.pow(2, NBits-1);// Para 8 bits el b
Datos3 = new double[Datos1.length];
```

```
for(int i = 0;i<Datos1.length;i++){
  // Convierte datos en double
  if(Datos1[i]<0)
    Datos3[i] = (double)Datos1[i]/bitMaximo;
  else
    Datos3[i] = (double)Datos1[i]/(bitMaximo-1);
}

// Calculo de energía del silencio
double energia[] = new double[Datos3.length/50];
for(int i = 0;i<energia.length;i++){
  for(int j = 0;j<50;j++){
    energia[i] += Datos3[i*50+j]*Datos3[i*50+j];
  }
}
// Nivel de ruido
double Max = f.promedio(energia) + 3*f.std(energia);
```

En este segmento de código se inicializan las variables con las cuales se manipulará la adquisición de audio. Posteriormente, se realiza la adquisición de un fragmento de audio, al cual se le calcula la energía en fragmentos de 50 muestras. Finalmente, del conjunto de energías calculadas se obtendrá un promedio junto con su tercera desviación estándar.

2. Adquisición de audio

```
public void LeerArreglo(){
  linea.read(Datos1, 0, Datos1.length);
  // Para 8 bits el b
  double bitMaximo = Math.pow(2, NBits-1);
  for(int i = 0;i<Datos1.length;i++){
    // Convierte datos en double
    if(Datos1[i] < 0)
      Datos3[i] = (double)Datos1[i]/bitMaximo;
    else
      Datos3[i] = (double)Datos1[i]/(bitMaximo-1);
    Datos2[i] = Datos1[i];
  }
}
```

De forma similar que lo hace el punto anterior, la adquisición de audio queda a cargo de la función *read(byte b[],int off, int len)* de la variable *TargetDataLine* llamada *linea*. Esta adquiere una señal que almacenará en la variable *Datos1*. Una vez que se ejecuta esta función, el programa queda en pausa esperando a que se el mismo le indique que se ha culminado la adquisición de muestras.

La magnitud de los valores adquiridos puede variar en el rango cerrado $[-2^{NBits-1}, 2^{NBits-1}-1]$. Sin embargo, para poder implementar los algoritmos de una forma sencilla se requiere que el rango de variación sea entre [−1,1] . Por lo tanto, se realiza una conversión como se observa en el programa.

La variable *Datos2* almacenará la señal original por si se requiere para algún otro proceso.

3. Análisis de audio

   Tras realizar la adquisición de audio se debe de buscar la señal a analizar entre todas las muestras de sonido adquiridas. Para esto se implementa el siguiente código.

```
// ////////////////////////////////////////////////////////////// //
// Busca la señal en el vector que recibe de entrada. Si se       //
// encuentra una señal se almacena en un vector dinamico. El      //
// existerá hasta que se determine que se ha encontrado el final  //
// de la señal y en dado caso se mandará a analizar segun sea el  //
// caso (tipo)                                                    //
// ////////////////////////////////////////////////////////////// //
  public void BuscarSenal(double vector[]){
    // Evalua que el vector tenga un tamaño aceptable
    if(vector.length > muestras){
      //Fracciones que se harán del vector
      NFracciones = (int)Math.ceil((double)vector.length/
          (double)muestras);
      // Energía del fragmento
      double E;
      // Indica los indices de inicio y fin que se van a copiar
      int inicio = -1, fin = -1;

      // Si continuar es falso significa que se debe de iniciar
      // nuevamente el vector para almacenar la información
       if(continuar == true){
        inicio = 0;
        fin = 0;
      }

      // Iteraciones del número de fracciones
      for(int i = 0;i<NFracciones;i++){
        // Se copian los valores
        for(int j = 0;j<muestras;j++){
          if(i*muestras + j < vector.length){
            vector2[j] = vector[i*muestras + j];
          }
        }

        // Se calcula la energía
```

```
E = Energia(vector2);

// Cuando no había energía y ahora si hay
if(inicio == -1 && E > EMinima){
  inicio = i;
  conEnergia++;
  sinEnergia = 0;
  fin = -1;
// Cuando había energía en fragmentos anteriores
}else if(inicio != -1){
  // Cuando hay energía
  if(E > EMinima){
    conEnergia++;
    // Se reinicia contador sin energía
    sinEnergia = 0;
    fin = -1;
  // Cuando no hay energía
  }else{
    // Hay solo energía
    if(fin == -1){
      fin = i - 1;
    }
    sinEnergia++;
    // Límite de sin energía
    if(sinEnergia > sinEMinima){
      // Cuando el acumulado de fracciones no
      // es sufuciente
      if(conEnergia < conEMinimo){
        senal = new Vector<Double>();
        inicio = -1;
        fin = -1;
        sinEnergia = 0;
        conEnergia = 0;
        continuar = false;

      // Cuando el acumulado de fracciones es
      //suficiente
      }else{
        // Se copian los valores en el vector
        for(int j = inicio*muestras;j<(fin+1)
            *muestras;j++){
          senal.add(vector[j]);
        }

        // Normalización
        double maximo = 0;
        double x[] = new double[conEnergia
                                *muestras];
```

```
              for(int k = 0;k<x.length;k++){
                x[k] = senal.get(k);
                if(Math.abs(x[k])>maximo){
                  maximo = Math.abs(x[k]);
                }
              }

              for(int k = 0;k<x.length;k++){
                x[k] = x[k]/maximo;
              }
              // Eliminación de frecuencias bajas
              double y[] = f.IIR(x, 10, 8000, 60, "PA");

              // ////////////// SALIDA  ////////// //
              // Se manda a procesar la señal    //
              // para obtener sus parámetros
              ProcesarLPC(y);
              // ////////////////////////////// //

              // Reinicio de variables
              senal = new Vector<Double>();
              inicio = -1;
              fin = -1;
              sinEnergia = 0;
              conEnergia = 0;
              continuar = false;
            }
          }
        }
      }
    }

    // Cuando se ha terminado de analizar las fracciones pero
    // no hay final o no se detecto fracciones sin energía
    if(inicio != -1 && conEnergia > 0){
      // Se define el final cuando hay energía pero no fin
      fin = NFracciones - 1;
      // Se copian los valores según los índices de
      // inicio y fin
      for(int j = inicio*muestras;j<(fin+1)*muestras;j++){
        if(j<vector.length);
          senal.add(vector[j]);
      }
      continuar = true;
    }
  }
}
```

En esta función se recibe el vector de datos que será analizado y lo divide en fracciones de 50 muestras, si es posible. Como se explica en los diagramas 5.21 y 5.22, se comienza a realizar una serie de iteraciones en las cuales se compararán los trozos del vector para determinar cuáles de ellos tienen energía y si permanecen consecutivos el tiempo suficiente para considerarse una señal útil. Cuando se detecta un trozo de señal con energía, este se almacena en un vector. Las fracciones que le sean consecutivas serán almacenadas en el mismo vector hasta el momento en que regrese a un silencio prolongado. Si el vector resultante tiene un tamaño aceptable, se hace pasar por un filtro que elimine las frecuencias bajas no útiles y se manda a la función *ProcesarLPC(double[] x)*.

```
public void ProcesarLPC(double x[]){
  double V[][] = Ventaneo(x);
  double a[][] = new double[V.length][];
  for(int i = 0;i<V.length;i++){
    a[i] = lpc(V[i],10);
  }
  if(tipo == 1){
    Buscar(a);
  }else if(tipo == 0){
    PA.insertar(a,x.length);
  }
}
```

La función *ProcesarLPC* recibe el vector de una señal que no es silencio. Esta hace uso de la función *Ventaneo(double[] x)*, que tiene el trabajo de dividir la señal en segmentos de 280 muestras y aplicar la ventana de Hamming. Con la matriz obtenida se pueden obtener los parámetros LPC de la señal. Estos son usados en la función *Buscar(double[][] a)*, que determinará si hay parecido con algún conjunto de la base de datos.

```
public void Buscar(double Referencia[][]){
  // Matriz para almacenar distancias
  double D[][][] = new double[Parametros.length][][];
  // Calculo de distancias
  for(int i = 0;i<D.length;i++){
    D[i] = DistanciaE(Referencia,Parametros[i]);
  }

  double acumulado1[] = new double[D.length];
  double acumulado2[] = new double[D.length];

  for(int i = 0;i < D.length;i++){
    Vector<Double> A = new Vector<Double>();
    Vector<Double> B = new Vector<Double>();
    Vector<Double> d = new Vector<Double>();
```

```
  for(int j = 0;j<D[i].length;j++){
    A.add(D[i][j][0]);
    B.add(D[i][j][1]);
    d.add(D[i][j][2]);
  }

  int k = 1;
  while(k < A.size()){
    if(A.get(k-1) == A.get(k)){
      if(d.get(k-1) < d.get(k)){
        A.remove(k);
        B.remove(k);
        d.remove(k);
      }else{
        A.remove(k-1);
        B.remove(k-1);
        d.remove(k-1);
      }
    }else{
      k++;
    }
  }
  k = 1;
  while(k < B.size()){
    if(B.get(k-1) == B.get(k)){
      if(d.get(k-1) < d.get(k)){
        A.remove(k);
        B.remove(k);
        d.remove(k);
      }else{
        A.remove(k-1);
        B.remove(k-1);
        d.remove(k-1);
      }
    }else{
      k++;
    }
  }
  acumulado1[i] = f.suma(d)/d.size();
  acumulado2[i] = (double)d.size()/Referencia.length;
  if(acumulado1[i] == -1)
    acumulado1[i] = Double.POSITIVE_INFINITY;
}

double min[] = minimo(acumulado1);
double max[] = maximo(acumulado2);
```

```
    // Más del 75% de coincidencia, se considera igual
    if(min[0] != Double.POSITIVE_INFINITY && acumulado2[(int)min[1]] >= 0.75){
      Palabra = Nombres[(int)min[1]];
    }else{
      Palabra = "Null";
    }

    System.out.println(Palabra);
    for(int a = 0;a<acumulado1.length;a++){
      System.out.println(acumulado1[a] + " , " + acumulado2[a]);
    }
  }
```

La función *Buscar(double[][][Referencia)*, recibe una matriz con los parámetros LPC de la señal que se está analizando. El conjunto es comparado uno a uno con los de la base de datos, que permanecen almacenados en la matriz Parametros. Cada comparación proporciona una matriz con tres columnas, donde la primera almacena las coincidencias de la señal base, la segunda las coincidencias de la señal comparada y la tercera la distancia resultante. Estas matrices son almacenadas en un vector de matrices *D*.

El algoritmo compara cada una de las coincidencias de las dos primeras columnas y elimina aquellas que estén repetidas. Esto da como resultado una relación uno a uno entre las coincidencias de los dos conjuntos. Se obtienen el porcentaje de coincidencia, tomando como referencia el número de puntos de la señal base (*Referencia*). Si el porcentaje sobrepasa el límite establecido se considera como la mima señal y se asigna el nombre correspondiente.

Para facilitar la interacción del usuario con el reconocedor, se ha optado por diseñar una interfaz gráfica sencilla. Esta consta de una ventana con dos opciones, figura 5.40. La primera se encarga de realizar la búsqueda de las palabras (figura 5.42), mientras que la segunda se encargará de generar la base de datos (figura 5.42). Esta última se basa en la misma teoría del reconocedor, con la diferencia de que en lugar de buscar la palabra en la base de datos, las almacena (código fuente: [19]).

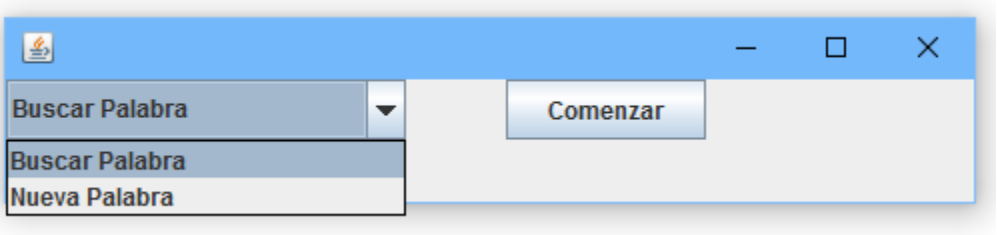


Figura 5.40: Interfaz gráfica principal

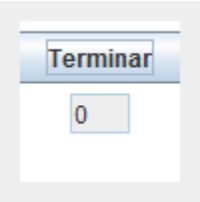


Figura 5.41: Interfaz gráfica que muestra la cantidad de palabras grabadas para hacer la base de datos

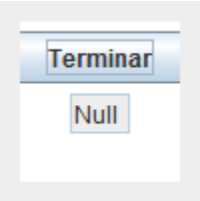


Figura 5.42: Interfaz gráfica que muestra la palabra reconocida

5.6. Evaluación de resultados

La evaluación del algoritmo de reconocimiento de palabras aisladas en tiempo real se realiza mediante la evaluación de un conjunto de palabras de la base de datos. Las palabras que se han seleccionado simulan los comandos básicos para una interfaz de control.

UNO
DOS
TRES
CUATRO
CINCO
SI
NO
ARRIBA
ABAJO
DERECHA
IZQUIERDA

Cada vez que el reconocedor detecte una perturbación y evalué que esta se trata de una palabra que pertenece a la lista, presentará el nombre asignado en la base de datos. Los diferentes nombres detectados serán observados en forma de lista.

Al comparar distintas señales que contienen la misma palabra con sus respectivos espectrogramas, se pueden observar patrones visuales propios de estas. En la figura 5.43 se observan cuatro versiones de la palabra 'Cinco'. Se puede observar que la distribución

y la intensidad de las frecuencias mantienen un patrón muy similar. Sin embargo, si la misma palabra es comparada contra otra, como se observan en los espectrogramas de la figura 5.44, se puede observar que dichos patrones ya no se cumplen. En el caso de la palabra 'Uno' las frecuencias se concentran en los valores bajos y son prácticamente continuas para toda la existencia de la palabra. Asimismo, se puede observar que el tiempo que dura varía entre palabras.

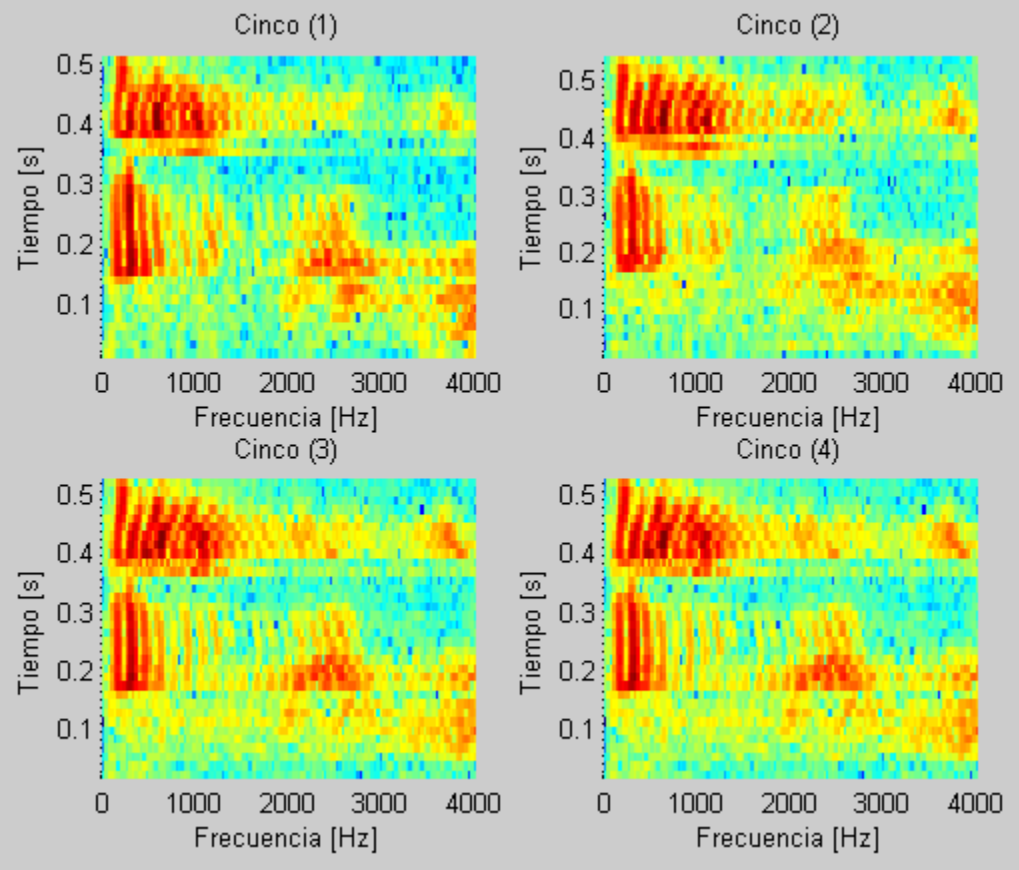


Figura 5.43: Comparación de espectrogramas de distintas señales de la palabra 'Cinco'

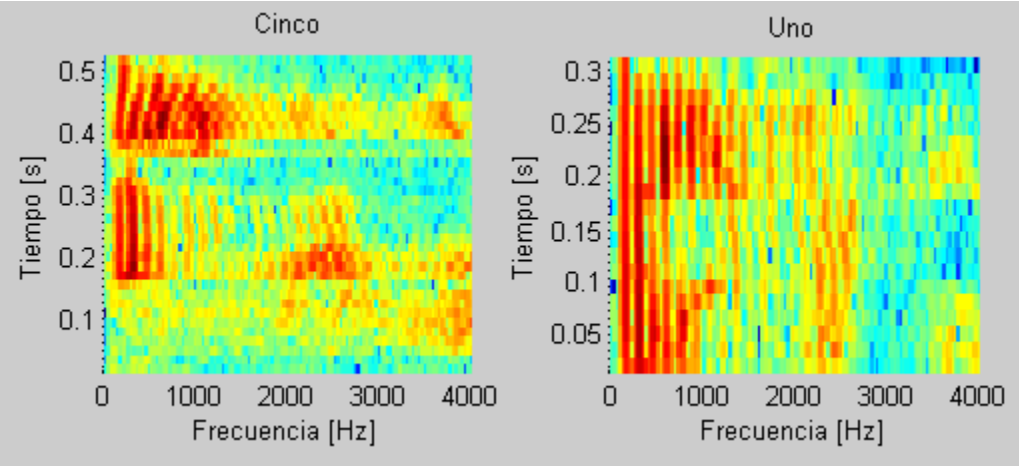


Figura 5.44: Comparación de espectrogramas de las palabras 'Cinco y 'Uno'

Experimento 1: Palabras en desorden

Para realizar una primera aproximación al nivel de precisión del algoritmo, se procede a repetir varias veces las palabras en diferente orden. A partir de las listas obtenidas se procederá a contar el número de éxitos de cada palabra y calcular el porcentaje de aciertos.

A continuación se presentan cinco tablas con todas las palabras en diferente orden. En cada una se presentan tres repeticiones para cada palabra, y en la última columna los aciertos obtenidos.

Palabras	**1ra repetición**	**2da repetición**	**3ra repetición**	**Exitos**
UNO	UNO	UNO	Null	2
DOS	Null	Null	Null	0
TRES	TRES	Null	TRES	2
CUATRO	CUATRO	CUATRO	CUATRO	3
CINCO	CINCO	Null	CINCO	2
SI	SI	SI	SI	3
NO	Null	NO	Null	1
ARRIBA	ARRIBA	ARRIBA	ARRIBA	3
ABAJO	ABAJO	Null	ABAJO	2
DERECHA	Null	DERECHA	Null	1
IZQUIERDA	Null	IZQUIERDA	IZQUIERDA	2

Palabras	1ra repetición	2da repetición	3ra repetición	Exitos
CINCO	CINCO	Null	Null	1
ABAJO	ARRIBA	ABAJO	ABAJO	2
IZQUIERDA	IZQUIERDA	Null	IZQUIERDA	2
UNO	UNO	UNO	ARRIBA	2
DOS	DOS	Null	DOS	2
NO	NO	NO	NO	3
SI	SI	Null	SI	2
DERECHA	Null	Null	Null	0
TRES	TRES	TRES	TRES	3
CUATRO	CUATRO	CUATRO	CUATRO	3
ARRIBA	ARRIBA	ARRIBA	Null	2

Palabras	1ra repetición	2da repetición	3ra repetición	Exitos
NO	NO	NO	NO	3
CUATRO	Null	CUATRO	CUATRO	2
ABAJO	Null	Null	ARRIBA	0
TRES	Null	TRES	TRES	2
DOS	Null	CUATRO	DOS	1
ARRIBA	ARRIBA	ARRIBA	ARRIBA	3
SI	CINCO	CINCO	Null	0
CINCO	CINCO	CINCO	CINCO	3
UNO	UNO	UNO	UNO	3
DERECHA	DERECHA	Null	Null	1
IZQUIERDA	ARRIBA	SI	SI	0

Palabras	1ra repetición	2da repetición	3ra repetición	Exitos
SI	SI	Null	Null	1
ABAJO	ABAJO	Null	Null	1
TRES	TRES	TRES	TRES	3
UNO	UNO	UNO	UNO	3
DERECHA	Null	TRES	DERECHA	1
ARRIBA	ARRIBA	ARRIBA	ARRIBA	3
CINCO	CINCO	SI	Null	1
DOS	Null	Null	CUATRO	0
IZQUIERDA	Null	Null	IZQUIERDA	1
NO	NO	NO	Null	2
CUATRO	Null	CUATRO	CUATRO	2

Palabras	1ra repetición	2da repetición	3ra repetición	Exitos
CUATRO	CUATRO	CUATRO	CUATRO	3
UNO	UNO	UNO	UNO	3
TRES	TRES	TRES	TRES	3
IZQUIERDA	IZQUIERDA	DERECHA	Null	1
ABAJO	ABAJO	ABAJO	Null	2
DERECHA	Null	DERECHA	DERECHA	2
SI	Null	Null	Null	0
CINCO	DOS	CINCO	CINCO	2
DOS	DOS	Null	SI	1
ARRIBA	ARRIBA	Null	ARRIBA	2
NO	Null	Null	Null	0

Para determinar los porcentajes de aciertos se suman los éxitos de las mismas palabras para todas las tablas. De esta forma se da origen a los siguientes valores

Palabra	Aciertos por tabla					Aciertos	
	1ra	2da	3ra	4ta	5ta	Totales	%
UNO	2	2	3	3	3	13	86
DOS	0	2	1	0	1	4	26
TRES	2	3	2	3	3	13	86
CUATRO	3	3	2	2	3	13	86
CINCO	2	1	3	1	2	9	60
SI	3	2	0	1	0	6	40
NO	1	3	3	2	0	9	60
ARRIBA	3	2	3	3	2	13	86
ABAJO	2	2	0	1	2	7	46
DERECHA	1	0	1	1	2	5	33
IZQUIERDA	2	2	0	1	1	6	40

Como se puede observar en la última columna de la tabla, seis de las once palabras tienen un porcentaje de aciertos mayor al 50 %. El menor porcentaje que se puede observar está referido a la palabra "DOS". Mientras que el mayor porcentaje se observa en las palabras "UNO", "TRES", "CUATRO" y "ARRIBA". En este ejemplo, si se suman todos los aciertos de las palabras se puede calcular el porcentaje total de éxito del sistema, equivalente a 0.59.

Experimento 2: Repetición múltiple

El segundo experimento realizado consiste en repetir la misma palabra veinte veces y estimar un porcentaje éxito con base en la lista proporcionadas. En la siguiente cuadro

muestra las listas obtenidas y los aciertos totales por palabra junto con sus respectivos porcentajes.

Palabra	**UNO**	**DOS**	**TRES**	**CUATRO**	**CINCO**	**SI**
Repeticiones	UNO	SI	TRES	CUATRO	CINCO	SI
	UNO	DOS	Null	CUATRO	CINCO	SI
	UNO	Null	TRES	CUATRO	CINCO	SI
	Null	DOS	NO	CUATRO	CINCO	Null
	UNO	DOS	TRES	CUATRO	CINCO	Null
	Null	DOS	NO	CUATRO	CINCO	SI
	UNO	UNO	TRES	CUATRO	Null	SI
	UNO	DOS	Null	Null	Null	SI
	CINCO	DOS	TRES	CUATRO	CINCO	Null
	UNO	DOS	TRES	Null	Null	Null
	UNO	DOS	TRES	Null	CINCO	SI
	UNO	NO	TRES	CUATRO	Null	Null
	Null	DOS	Null	CUATRO	CINCO	SI
	Null	Null	TRES	CUATRO	CINCO	SI
	Null	DOS	TRES	CUATRO	Null	Null
	UNO	Null	Null	CUATRO	CINCO	Null
	UNO	UNO	TRES	CUATRO	CINCO	CINCO
	Null	DOS	TRES	CUATRO	Null	SI
	ARRIBA	UNO	TRES	CUATRO	CINCO	SI
	UNO	Null	Null	Null	CINCO	CINCO
Aciertos	12	11	13	16	14	11
Aciertos [%]	60	55	65	80	70	55

Palabras	NO	ARRIBA	ABAJO	IZQUIERDA	DERECHA
Repeticiones	NO	ARRIBA	ABAJO	IZQUIERDA	DERECHA
	NO	UNO	Null	IZQUIERDA	DERECHA
	NO	ARRIBA	ABAJO	IZQUIERDA	DERECHA
	NO	ARRIBA	ABAJO	IZQUIERDA	DERECHA
	NO	ARRIBA	ABAJO	IZQUIERDA	DERECHA
	UNO	ARRIBA	ABAJO	IZQUIERDA	Null
	UNO	ARRIBA	ABAJO	Null	DERECHA
	NO	ARRIBA	Null	Null	Null
	NO	ARRIBA	ABAJO	Null	DERECHA
	NO	ARRIBA	ABAJO	IZQUIERDA	Null
	NO	ARRIBA	ABAJO	IZQUIERDA	DERECHA
	NO	ARRIBA	ABAJO	Null	Null
	NO	ARRIBA	ABAJO	IZQUIERDA	DERECHA
	NO	ARRIBA	Null	Null	DERECHA
	NO	ARRIBA	ABAJO	IZQUIERDA	Null
	Null	ARRIBA	Null	IZQUIERDA	DERECHA
	NO	ARRIBA	ABAJO	IZQUIERDA	Null
	Null	ARRIBA	ABAJO	IZQUIERDA	DERECHA
	Null	ARRIBA	ABAJO	UNO	DERECHA
	UNO	ARRIBA	ABAJO	Null	DERECHA
Aciertos	14	19	16	13	14
aciertos [%]	70	95	80	65	70

Los resultados que se muestran en este segundo experimento sobrepasan el 50 % de aciertos para todos los casos. En este ejercicio la palabra con el mayor porcentaje de aciertos es “ARRIBA”, seguidos por las palabras “ABAJO” y “CUATRO”. Mientras que el que posee un menor porcentaje es “DOS” y “SI”.

5.6.1. Conclusiones

Los resultados del segundo ejercicio, a comparación de los observados en el primero, poseen un mayor índice de asertividad. Esto podría deberse a que en el segundo caso la repetición continua de la misma palabra permite que se mantengan las mismas características, a diferencia de las repeticiones del primero en el que se realizaban en desorden.

En la figura 5.45 se puede observar la comparación de los porcentajes de cada palabra para ambos ejercicios. Al observar el comportamiento de los dos conjuntos se puede resaltar que actúan parecido pero con diferentes magnitudes.

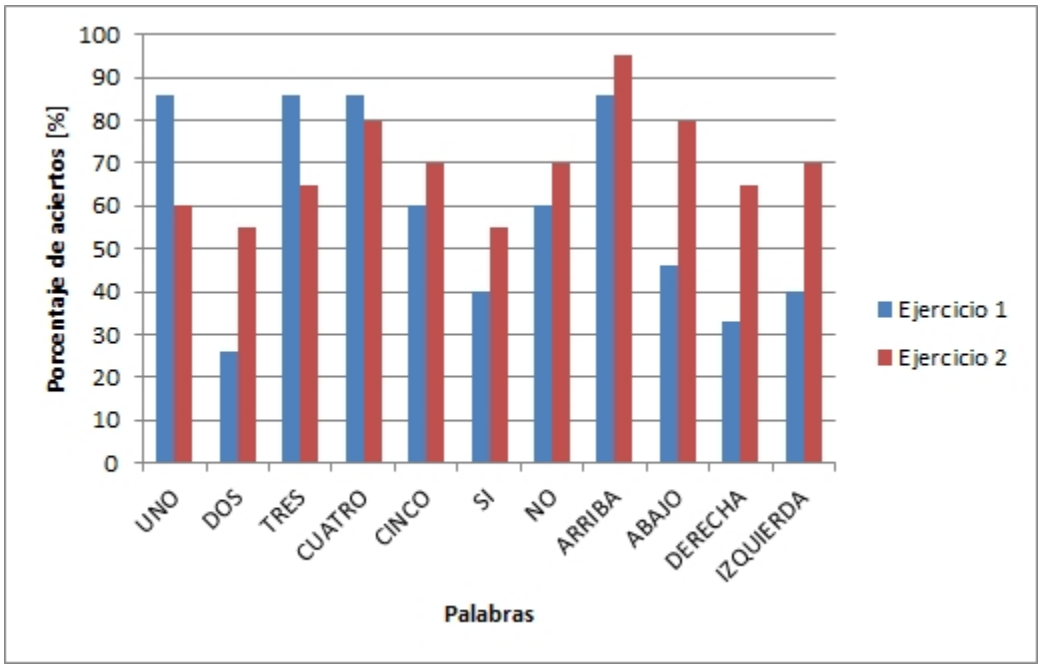


Figura 5.45: Comparación de resultados de la evaluación del reconocedor

Capítulo 6

Conclusiones

El reconocimiento de voz es un sistema que trata de imitar la acción humana e intercambiar información por medio del habla. Pretende establecer una forma de interacción hombre-máquina que resulte natural e intuitiva. Los sistemas de reconocimiento de voz aumentan su complejidad mientras mayor sea la fidelidad con la que se trate de reconocer el habla. Por esta razón, los sistemas más simples de implementar son aquellos que solo manejan un conjunto limitado de palabras de un único individuo.

Los sistemas de reconocimiento de voz de palabras aisladas se enfocan en analizar una lista de palabras, las cuales serán caracterizadas para el usuario que los utilice. Uno de los algoritmos que se suelen emplear para este tipo de reconocimiento se basan en los parámetros LPC y su comparación a través de la distancia euclidiana.

Los parámetros LPC se obtienen a partir del análisis de una señal y permiten su reconstrucción, si se emplean como coeficientes de un filtro de ecuaciones en diferencias. Señales similares producirán parámetros similares, por lo que es posible establecer si dos señales están relacionadas o no. Esta comparación se lleva a cabo por medio de la distancia Euclidiana, la cual nos proporcionará un valor cada vez que se comparen dos conjuntos de parámetros. Este valor indicará que tan parecido son dos señales.

El algoritmo que se ha propuesto en este documento se basa en la comparación de los parámetros LPC de dos señales. Los parámetros contra los que se compara fueron obtenidos tras analizar distintas señales de una misma palabra. Estos son almacenados en una base de datos, de la cual son leídos cada vez que se requiere la identificación de una nueva señal. A cada conjunto se le asigna un nombre con el cual identificarlo. Este será usado cada vez que el sistema identifique una palabra y se presentará en forma de una lista junto con otras que hayan sido reconocidas.

En esta propuesta también se contempla que la señal que contiene las palabras a identificar es una señal en tiempo real. El algoritmo se encarga de analizarla y extraer aquellas secciones donde se considere que existe una posible palabra. Sin embargo,

pueden existir momentos en los que un ruido con una intensidad y tiempo prolongados se presente, y el algoritmo lo considere como una señal válida. En la lista de palabras reconocidas se representarán a todas aquellas señales que son válidas o no identificadas con la palabra "Null".

Se ha demostrado que el sistema propuesto tiene una mayor asertividad cuando las palabras son repetidas en múltiples ocasiones. Esto podría deberse a que las características en la pronunciación no varían de la misma forma que lo hacen cuando se pronuncian distintas palabras seguidas. En este último caso se observaron que se acertaba en más del 50 % de los casos solo en seis de las once palabras de la base de datos, mientras que en el otro experimento se observó un índice de aciertos superior al 50 % en todos los casos.

El reconocimiento de voz es un tema que se puede abordar por más de un método. Cada uno de ellos tendrá ventajas y desventajas dependiendo de las necesidades que se requieran cubrir. La solución que se ha propuesto aquí resuelve el problema de reconocer palabras aisladas en tiempo real. Este es un método fácil de aplicar y que puede ser llevado a otros lenguajes de programación con relativa facilidad. Una posibilidad podría ser el traducir los algoritmos a un lenguaje compatible con los dispositivos móviles. Facilitando la interacción con el programa por parte del usuario.

Bibliografía

[1] Emma Rodero Antón, *El tono de la voz masculina y femenina en los informativos radiofónicos: un análisis comparativo.* Universidad Pontificia de Salamanca http://www.bocc.ubi.pt/pag/rodero-emma-tono-voz-femenina.pdf, 10/03/2016

[2] Oppenheim Alan V., *Señales y sistemas, Segunda Edición*, Prentice Hall Hispanoamericana S.A., México, 1998

[3] Oppenheim Alan V., *Tratamiento de señales en tiempo discreto*, 3ra edición, PEARSON EDUCACIÓN S.A., España 2011

[4] Sáenz Peña Roque, *Teoría de Telecomunicaciones*, Universidad Nacional de Quilmes, Departamento de Ciencia y Tecnología, http://iaci.unq.edu.ar/materias/telecomunicaciones/apuntes.htm, 28/03/2016

[5] G. Proakis, Johan, *Tratamiento digital de señales 3ª Edición*, 1998, PRENICE HALL, Madrid

[6] Dr. Alvarado Moya, José Pablo. *Procesamiento digital de señales*, 2011, Tecnológico de Costa Rica, Escuela de Ingeniería Electrónica.

[7] J. I. Huircán, *Filtros Activos, Conceptos Básicos y Diseño*, http://quidel.inele.ufro.cl/∼jhuircan/PDF_ CTOSII/ieeefact.pdf (18/11/2015)

[8] C.Montgomery, Douglas, *Probabilidad y Estadistica aplicada a la ingeniería 2ª Edicion*, LIMUSA WILEY, 2002

[9] W. Leon, Couch, *Sistemas de comunicación digitales y analógicos. Séptima edición.*, PEARSON EDUCACIÓN, México, 2008

[10] Papamichalis E. Panos., *Practical Approaches to Speech Coding*, Prentice-Hall, EUA, 1987

[11] Varona Fernández, Amparo. *Antecedentes y desarrollo de los sistemas actuales de reconocimiento automático del habla.* Universidad del País Vasco. 15/06/16 http://hedatuz.euskomedia.org/6564/1/04321346.pdf

[12] López Martín, Alberto. *Ingeniería de ondas. Formatos de Audio digital*. Universidad de Valladolid. E.T.S. Ingenieros de Telecomunicación. (http://www.analfatecnicos.net/archivos/32.FormatosDeAudioDigital.pdf, Octubre de 2015)

[13] Belén Ruiz, Mezcua. *LA VOZ Y SU ESPECTRO*. Noviembre 2005. (http://www.hezkuntza.ejgv.euskadi.net/r43-573/es/contenidos/informacion/dia6_sigma/es_sigma/adjuntos/sigma_27/11_la_voz.pdf, Septiembre de 2016)

[14] Francisco Pastor *Sistemas Informáticos de Tiempo Real*, Universidad de Valencia, 19 de Marzo de 2005, (http://www.uv.es/gomis/Apuntes_SITR/Programacion.pdf, Septiembre de 2016)

[15] Aarón García, Nelson Torres *Diseño de filtro de tiempo discreto tipo Butterworth por el método de transformada bilineal*. Ingeniería de Telecomunicaciones e Ingeniería en Automatización. Universidad Don Bosco.

[16] Álvarez Fernández, Luis Alberto *Sistema de alarma residencial empleando voz sintética vía tenefónica*. Facultad de Ingeniería, Universidad Nacional Autonoma de México.

[17] Lawrence Rabiner *Fundamentals of Speech Recognition*. Prentice-Hall International, Inc. 1993

[18] Orlando Ortega *El Órgano Melódico*, Mayo 12 de 2009, (https://ortegareyes.wordpress.com/tag/ernesto-hill-olvera/ Octubre de 2016)

[19] Hernández Escobedo, Cristian. *Código fuente de la tesis de licenciatura: diseño e implementación en tiempo real de un reconocedor de palabras aisladas*. Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2016. (crs.che@gmail.com)

Anexos

Anexo A

Adquisición de audio con java

En Java, la adquisición de audio está a cargo de la librería javax.sound.sampled, en donde encontramos las clases, funciones y métodos dedicados a este tema. Con esta podemos configurar la tarjeta de sonido, la transferencia de información desde esta última y los formatos del audio para su almacenamiento en memoria. Mientras que para poder almacenar la información adquirida se tiene que recurrir a la librería java.io, la cual permite la manipulación de archivos en general.

Algunos de los métodos que podrían emplearse en la adquisición de audio en java y su correspondiente almacenamiento en memoria se mencionan a continuación.

- *java.io* - Librería para la manipulación, creación o reemplazo de la información de archivos, tanto de salida como de entrada.
  - *File* – Representación en Java de un archivo y su ubicación en memoria. Su sintaxis es la siguiente

    public File(String pathname)

    Donde la variable *pathname* es la dirección de almacenamiento del archivo.
- *javax.sound.sampled* – Librería para la adquisición, reproducción y procesamiento de muestras de audio.
  - *AudioFormat* – En esta función se especifican las características de la adquisición de la señal. Contiene diferentes formas de configuración, una de ellas es la presentada a continuación:

    public AudioFormat(float sampleRate, int sampleSizeRate, int Channels, boolean signed, boolean bigEndian);

    Donde los parámetros se definen de la sigueinte manera:

- *sampleRate*: Frecuencia de muestreo. Algunas frecuencias típicamente usadas son:
  - 8 kHz y 16 kHz - Estándares de telefonía (voz)
  - 44.1 khz - Señales de audio para la gama perceptible por el oído humano
- sompleSizeRate: Bits por muestra. Donde las más habituales son 8 y 16 bits. [12]
- Channels: 1 para señal mono y 2 para señal estéreo
- signed: Indica si la información será signada o no
- bigEndian: Especifica el formato de escritura, ya sea big-endian (verdadero) o little-endian (falso)

- DataLine.Info – Almacena información para ser empleada en la línea de datos. Una de las sintaxis que acepta es la siguiente

  DataLine.Info info = new DataLine.Info(Class¡?¿lineClass, AudioFormat format);

  Dónde

  - *lineClass*: Clase de la línea de datos
  - *format*: Formato deseado para capturar la señal

- *AudioSystem* – Genera la línea de datos que se emplea. Contiene los métodos para la configuración del formato y manipulación de la información

- *getLine* – Regresa una línea de datos con las configuraciones especificadas en su argumento

  public static Line getLine(Line.Info info)

  Donde la variable info contiene las configuraciones de la línea

- *write* – Transfiere la información adquirida hacia un archivo de audio con los valores especificados.

  public static int write(AudioInputStream stream, AudioFileFormat.Type fileType, File out)

  Donde

  - *stream*: El flujo de información que se escribirá en el archivo
  - *fileType*: El formato o tipo de archivo en el que se va a escribir
  - *out*: referencia al archivo en el que se va a escribir

- *AudioFileFormat.Type* – Permite especificar el formato de audio a manejar. Java es capaz de soportar los siguientes formatos:

  - AIFF – C
  - AIFF

- AU
- SND
- WAVE

Su sintaxis para un archivo de tipo WAVE es la siguiente

AudioFileFormat.Type fileType = AudioFileFormar.type.WAVE

Para cualquier otro formato, se debe de cambiar la extensión WAVE por su correspondiente.

- *TargetDataLine* – Genera una línea de transmisión de datos del tipo DataLine. Contiene funciones para la manipulación del flujo.
  - *open*: Método heredado, abre la línea de datos con el formato especificado

    void open(AudioFormat format)
  - *close*: Método heredado, cierra la línea de datos

    void close()
  - *read*: leer los datos de audio adquiridos especificados por len con un adelanto de off y los almacena en el vector b.

    int read(byte[] b, int off, int len)
- *DataLine* – Línea de datos para la transmisión de información
  - *start*: inicia la línea de captura.

    void start()
  - *stop*: detiene la adquisición de información

    void stop()
- *AudioInputStream* – Construye un formato de entrada con los valores especificados

Con este conjunto de comandos es posible el diseñar un programa que capture una señal de audio desde la tarjeta de sonido de la computadora con determinada configuración y un tiempo determinado. Además de poder almacenar en memoria permanente la información capturada con un formato de audio específico. A continuación se enunciará un programa que captura audio durante un minuto a una frecuencia de muestreo de ocho mil con un tamaño de muestra de ocho bits en un solo canal.

De forma generalizada, el programa puede ser descrito en los siguientes puntos

- Se especifican los datos para dar el formato de adquisición: tiempo de captura (t), frecuencia de muestreo (FM), bits por muestra (NBits) y Número de canales (NC)

- Se configura la línea de captura con los datos antes especificados (AudioFormat), el buffer de transferencia (DataLine.Info) y la línea de transmisión de datos (TargetDataLine)
- Se adquiere la información con N muestras, equivalentes a t*FM, a través de la función read de TargetDataLine
- Se configura el arreglo (ByteArrayInputStream) para almacenar el archivo. También se configuran el formato especifico del archivo (AudioInputStream) y se abre el archivo en el que se almacenará (File).
- Se escribe la información dentro del archivo (AudioSystem.write)

A continuación se muestra un programa diseñado a partir del listado anterior de sentencias. Este realiza un adquisición de audio de una duración *t*.

- Frecuencia de muestreo: 8000 Hz
- Tamaño por muestra: 8 bits
- Número de canales: 1
- Valores signados
- Formato de almacenamiento: WAVE
- Ruta y nombre del archivo: "/Prueba.wav"

```
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.DataLine;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.TargetDataLine;
import javax.sound.sampled.AudioFileFormat.Type;

public class CapturaAudio{
    public static void main(String[] args){
        int t = 1;
        int FM = 8000;
        int NBits = 8;
        int NC = 1;
```

```
        int N = t * FM;
        byte Datos[] = new byte[N];
        String Nombre = "Archivo.wav";

        AudioFormat format = null;
        DataLine.Info info;
        TargetDataLine linea;

        InputStream bIS;
        AudioInputStream stream;
        File file;

        try{
            format = new AudioFormat(FM,NBits,NC,true,true);
            info = new DataLine.Info(TargetDataLine.class, format);
            linea = (TargetDataLine) AudioSystem.getLine(info);
            linea.open(format);
            linea.start();

            linea.read(Datos, 0, Datos.length);
            linea.stop();
        }catch(LineUnavailableException ex){}

        bIS = new ByteArrayInputStream(Datos);
        stream = new AudioInputStream(bIS,format,Datos.length);
        file = new File(Nombre);
        try{
            AudioSystem.write(stream, Type.WAVE, file);
        }catch(IOException e){}
    }
}
```

Anexo B

Procesos en paralelo en Java, función *Thread*

Java posee una librería llamada *Thread*, la cual permite llevar a cabo acciones en paralelo siguiendo una estructura ya establecida. Esta consiste en una función principal (*run*), donde se contendrán las acciones a ejecutar, contenida dentro de una clase que hereda la librería.

```
class Paralelo extends Thread{
    public void run(){
        // Funciones
    }
}
```

Para comenzar la ejecución de la acción en paralelo se hace uso de una función ya contenida, *start*. Una vez que se ha iniciado comenzará a correr el programa al mismo tiempo que el principal, desde donde se le realizó la llamada. Este continuará su ejecución hasta el momento en que el código sea finalizado. La librería nos facilita la función *join*, la cual pausará el programa principal hasta que se haya finiquitado la ejecución en paralelo.

```
public void funcion(){
    Paralelo p = new Paralelo();
    p.start();
    try{
        p.join();
    }catch(Exception e){}
}
```