
# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRIA Y DOCTORADO EN INGENIERIA

FACULTAD DE INGENIERÍA

## SISTEMA TELEFÓNICO CON SÍNTESIS DE VOZ Y DETECCIÓN DE TONOS

## T E S I S

QUE PARA OPTAR POR EL GRADO DE:

## MAESTRO EN INGENIERÍA

**INGENIERÍA ELÉCTRICA – PROCESAMIENTO DIGITAL DE SEÑALES**

P R E S E N T A

**CARLOS MAYA LEÓN**


TUTOR:

**M. I. LARRY HIPÓLITO ESCOBAR SALGERO**

MÉXICO, D. F. 2010

**JURADO ASIGNADO:**

Presidente: Dra. Medina Gómez Lucia

Secretario: Dr. Peña Cabrera Mario

Vocal: M. I. Escobar Salguero Larry

1er. Suplente: Dr. Psenicka Bohumil

2do. Suplente: M. I. Quintana Thierry Sergio

Lugar o lugares donde se realizó la tesis:

**Posgrado, Facultad de Ingeniería U.N.A.M.**

**TUTOR DE TESIS:**

M. I. Escobar Salguero Larry H.

_____

**FIRMA**

# Dedicatoria

Dedico este trabajo a mi esposa Vicky que sin su apoyo no hubiera sido posible y a mis hijos Carlos y José que llenan de alegría mi vida.

Agradezco a mi tutor Larry y a todos mis profesores que compartieron su conocimiento sin esperar nada a cambio.

# Índice


**1. INTRODUCCIÓN ..... 3**

1.1 OBJETIVO ..... 3
1.2 JUSTIFICACIÓN ..... 3
*1.2.1 Estado Del Arte ..... 4*
1.3 DESCRIPCIÓN GENERAL DEL TRABAJO ..... 4
*1.3.1 Metodologías ..... 7*
*1.3.2 Método a utilizar ..... 8*
*1.3.3 Resultados esperados ..... 9*
*1.3.4 Estructura de la tesis ..... 9*

**2. PRODUCCIÓN DE VOZ ..... 10**

2.1 GENERACIÓN DE VOZ ..... 10
*2.1.1 Producción del habla ..... 11*
2.2 CARACTERÍSTICAS DE LA VOZ ..... 14
*2.2.1 Vocales y consonantes ..... 14*
*2.2.2 Oralidad y nasalidad ..... 14*
*2.2.3 Tonalidad ..... 14*
*2.2.4 Lugar y modo de articulación (consonantes) ..... 15*
*2.2.5 Posición de los órganos articulatorios (vocales) ..... 16*
*2.2.6 Duración ..... 17*
*2.2.7 Sonido voceado y no voceado ..... 17*
2.3 ANÁLISIS DE LA GENERACIÓN DE VOZ ..... 17
*2.3.1 Modelo acústico de producción de voz ..... 18*
*2.3.2 Modelo de tubos concatenados sin pérdidas ..... 18*
*2.3.3 Efecto de radiación de los labios ..... 18*
*2.3.4 Excitación del sonido en el tracto vocal ..... 19*
*2.3.5 Modelo general de síntesis de voz ..... 19*
*2.3.6 Ventaneo de la señal de voz ..... 21*
2.4 REPRESENTACIÓN DE LA SEÑAL DE VOZ ..... 25
*2.4.1 Análisis de tiempo-corto de Fourier ..... 25*
*2.4.2 Energía del segmento de voz ..... 26*
2.5 PITCH ..... 26
*2.5.1 Métodos de determinación de pitch ..... 26*
*2.5.2 Método de Recorte Central ..... 26*
2.6 RESUMEN ..... 30

**3. SÍNTESIS DE VOZ ..... 31**

3.1 PRODUCCIÓN DE VOZ SINTÉTICA ..... 31
3.2 SÍNTESIS DE VOZ POR FORMANTES ..... 32
3.3 MODELO TODO-POLO Y TODO-CERO ..... 33
*3.3.1 Forma Directa ..... 34*
3.4 PREDICCIÓN LINEAL ..... 35
*3.4.1 El problema de Predicción Lineal ..... 36*
*3.4.2 Minimización de Error ..... 38*
*3.4.3 Ecuación normal y principio de ortogonalidad ..... 40*
*3.4.4 Método de Autocorrelación ..... 40*
*3.4.5 Algoritmo de Levison-Durbin ..... 41*
3.5 CODIFICADOR DECODIFICADOR DE VOZ LPC (VOCODER) ..... 46
3.6 RESUMEN ..... 48

**4. COMUNICACIÓN TELEFÓNICA ..... 49**

4.1 PRINCIPIOS DE TELEFONÍA ..... 49
*4.1.1 Circuito telefónico ..... 50*
*4.1.2 Realizando una Llamada ..... 51*
*4.1.3 Medios de transmisión ..... 54*
*4.1.4 Marcación por Multi-frecuencia de doble tono ..... 55*
4.2 TRANSMISIÓN DE VOZ ..... 56
*4.2.1 Ancho de banda ..... 56*
4.3 RESUMEN ..... 57

**5. CODIFICACIÓN EN MULTI-FRECUENCIA DE DOBLE TONO ..... 58**

5.1 DEFINICIÓN Y GENERACIÓN DE TONOS DTMF ..... 58
*5.1.1 Generación de tonos por aproximación de polinomios ..... 61*
*5.1.2 Generación de tonos con un oscilador digital recursivo ..... 62*
*5.1.3 Generación por búsqueda en tabla ..... 65*
5.2 DECODIFICACIÓN DE TONOS DTMF ..... 66
*5.2.1 Especificaciones ..... 67*
*5.2.2 Algoritmo de detección de tonos ..... 68*
5.3 CONSIDERACIONES DE IMPLEMENTACIÓN ..... 73
*5.3.1 Prueba de Magnitud ..... 75*
*5.3.2 Prueba de Twist ..... 75*
*5.3.3 Prueba de potencia de las frecuencias ..... 75*
*5.3.4 Prueba de energía total ..... 75*
*5.3.5 Prueba del segundo armónico ..... 76*
*5.3.6 Decodificador de dígito ..... 76*
5.4 RESUMEN ..... 77

**6. DISEÑO Y DESARROLLO DEL SISTEMA ..... 78**

6.1 DESCRIPCIÓN GENERAL DEL SISTEMA ..... 79
6.2 ELEMENTOS BÁSICOS DE UN SISTEMA DE PDS EN TIEMPO REAL ..... 80
*6.2.1 Acondicionamiento de señal y muestreo ..... 81*
*6.2.2 Opciones de hardware ..... 82*
*6.2.3 Dispositivos de punto fijo y punto flotante ..... 83*
6.3 DSP TMS320C5402 ..... 84
*6.3.1 Características generales ..... 84*
*6.3.2 Sistema de Desarrollo TMS320C5402 DSK ..... 85*
6.4 SOFTWARE ..... 88
*6.4.1 Análisis fuera de línea ..... 88*
*6.4.2 Procesamiento en línea o Tiempo Real ..... 95*
6.5 RESUMEN ..... 102

**7. RESULTADOS ..... 103**

7.1 EVALUACIÓN DE LA VOZ SINTÉTICA GENERADA ..... 103
7.2 DECODIFICACIÓN DE TONO DTMF ..... 106
7.3 EVALUACIÓN DE RECURSOS DEL SISTEMA ..... 110
*7.3.1 Hardware y software ..... 110*
7.4 RESUMEN ..... 113

**8. CONCLUSIONES ..... 114**

**9. GLOSARIO ..... 115**

**10. ANEXOS ..... 117**

ANEXO A - CÓDIGO DEL DSP. ..... 117
**11. BIBLIOGRAFÍA ..... 127**

# Introducción

La línea telefónica desde hace varias décadas se ha convertido en un canal flexible de comunicación, no sólo de voz, si no para transmitir datos. Gracias a esto es posible navegar en internet, controlar dispositivos como son el fax, encender y apagar luces, etc. Uno de los métodos más utilizado para este tipo de comunicación es el denominado Multifrecuencia de doble tono (DTMF por sus siglas en ingles Dual Tone Multi-Frecuency); el cual nos permite enviar códigos por tonos del 0 al 9, de A hasta D, y los caracteres "*" y "#"; con lo que se pueden mandar datos confiables desde un emisor a un receptor, donde este último interpretará los dígitos recibidos y a su vez realizará una acción en función de ello.

Dado que existen múltiples sistemas, y por lo regular la propiedad intelectual está al resguardo de empresas o individuos, surgió la idea de implementar por medios propios, un sistema completo de interacción por voz y detección de tonos, a través de una línea telefónica, y apoyados por una tarjeta de desarrollo con un Procesador Digital de Señales (DSP) que nos permita estar almacenando la información enviada por los usuarios.

Por otro lado, la idea también surgió como una forma de aprovechar la infraestructura con que se cuenta en el laboratorio de procesamiento digital de señales de posgrado de la Facultad de Ingeniería de la UNAM y desarrollar experiencia práctica en este tipo de sistemas.

# CAPITULO UNO

# 1. Introducción

La línea telefónica desde hace varias décadas se ha convertido en un canal flexible de comunicación, no sólo de voz, si no para transmitir datos. Gracias a esto es posible navegar en internet, controlar dispositivos como son el fax, encender y apagar luces, etc. Uno de los métodos más utilizado para este tipo de comunicación es el denominado Multifrecuencia de doble tono (DTMF por sus siglas en ingles Dual Tone Multi-Frecuency); el cual nos permite enviar códigos por tonos del 0 al 9, de A hasta D, y los caracteres "*" y "#"; con lo que se pueden mandar datos confiables desde un emisor a un receptor, donde este último interpretará los dígitos recibidos y a su vez realizará una acción en función de ello.

Dado que existen múltiples sistemas, y por lo regular la propiedad intelectual está al resguardo de empresas o individuos, surgió la idea de implementar por medios propios, un sistema completo de interacción por voz y detección de tonos, a través de una línea telefónica, y apoyados por una tarjeta de desarrollo con un Procesador Digital de Señales (DSP) que nos permita estar almacenando la información enviada por los usuarios.

Por otro lado, la idea también surgió como una forma de aprovechar la infraestructura con que se cuenta en el laboratorio de procesamiento digital de señales de posgrado de la Facultad de Ingeniería de la UNAM y desarrollar experiencia práctica en este tipo de sistemas.

## 1.1 Objetivo

Diseñar y desarrollar un sistema de detección de tonos, en conjunto con un sintetizador de voz, que interactué con un usuario a través de una línea telefónica almacenando las opciones seleccionadas.

## 1.2 Justificación

Con los algoritmos y técnicas estudiadas durante los cursos de maestría, se tiene la información y capacidad necesaria para realizar proyectos que son usados actualmente en la industria, con la posibilidad de ser mejorados o integrarlos con otros sistemas. En este proyecto se lleva a cabo el diseño de un sistema de detección de tonos, incluyendo síntesis de voz para realizar un sistema "stand alone" donde la utilización de una Computadora Personal (PC) sea opcional.

### *1.2.1 Estado Del Arte*

Históricamente, desde la Antigua Grecia se han realizado intentos por generar voces artificiales, en muchos casos eran simplemente juegos de tuberías conectadas a un locutor humano, en otros auténticos ingenios acústicos capaces de producir sonoridades vocálicas.

Algunos ejemplos de ello son las “cabezas hablantes” (speaking heads) que involucran a Gerbert de Aurillac (1003), Alberto Magnus (1198-1280), y Roger Bacon (1214-1294). Christian Kratzenstein, en 1779 construye un modelo del tracto vocal que puede producir las cinco vocales largas. Wolfgang von Kempelen, en 1791 desarrolla una máquina de voz “acústica-mecanica”. En 1837 Charles Wheatstone produce una “Máquina de voz” que reproduce consonantes y vocales basado en el diseño de Kempelen. En 1857 M. Fuber, construye “Euphonia” y en 1923 Paget retoma el diseño de Wheatstone [19], [2].

El desarrollo de la telefonía a principios del siglo XX motivó intensas investigaciones sobre las propiedades de la voz y la audición con el fin de mejorar la calidad de la comunicación telefónica. El proceso sigue y hoy en día las tecnologías existentes permiten, por ejemplo, disponer de sistemas de comunicación oral hombre-máquina [3].

Actualmente existen muchos sistemas donde la detección de tonos y la síntesis de voz están presentes; en lo que respecta a la síntesis de voz, se realiza básicamente con el método de concatenación de fonemas, además se investiga los aspectos de “emoción” o “voz expresiva”, el cual consiste en dar más información a través de las variaciones de tono al hablar para expresar algo en particular. Por otro lado, también existe un término denominado “voz multimodal”, que consiste en mostrar la información gráfica, por medio de gestos y expresiones faciales, por lo que se debe guardar de igual manera la información de los gestos y sincronizarlos al sintetizar la voz.

En lo que respecta a la detección de tonos, se tiene una gran variedad de aplicaciones como son: el control de sistemas de forma remota, la navegación por diferentes opciones en un sistema telefónico, o bien la realización de operaciones bancarias de manera automática.

## 1.3 Descripción general del trabajo

En este trabajo se desarrolla un sistema que detecta los tonos DTMF y que responde con voz sintética, como se ilustra en la figura 1.1.

El sistema se conecta a una línea telefónica y los procesos que realiza son:

- El usuario marca al número telefónico donde el sistema está conectado.
- Se recibe la llamada y el sistema contesta con voz sintética, dando al usuario un menú con diferentes opciones a escoger por medio del teclado telefónico.
- El usuario a través de los dígitos del teclado manda la opción seleccionada.
- El sistema detecta el dígito presionado y da otro submenú dependiendo de la elección del usuario.

- El usuario selecciona otra opción por medio del teclado telefónico.
- El sistema detecta el dígito presionado y da un segundo submenú dependiendo de la elección del usuario.
- El usuario a través de los dígitos del teclado manda una opción seleccionada.
- El sistema almacena las opciones escogidas por el usuario en una base de datos, para posteriormente recuperar por algún medio la información.
- El sistema da por terminada la operación.

En la figura 1.2 se muestra en forma secuencial a través del tiempo los procesos de comunicación entre el usuario y el sistema.

El usuario cuenta con una terminal telefónica con capacidad de marcado con tonos, y el sistema cuenta con un Procesador Digital de Señales (DSP), específicamente el TMS320C5042 de Texas Instruments (TI). El sistema atiende un proceso a la vez, es decir, cuando se está enviando la voz sintética no atiende la decodificación DTMF y viceversa.

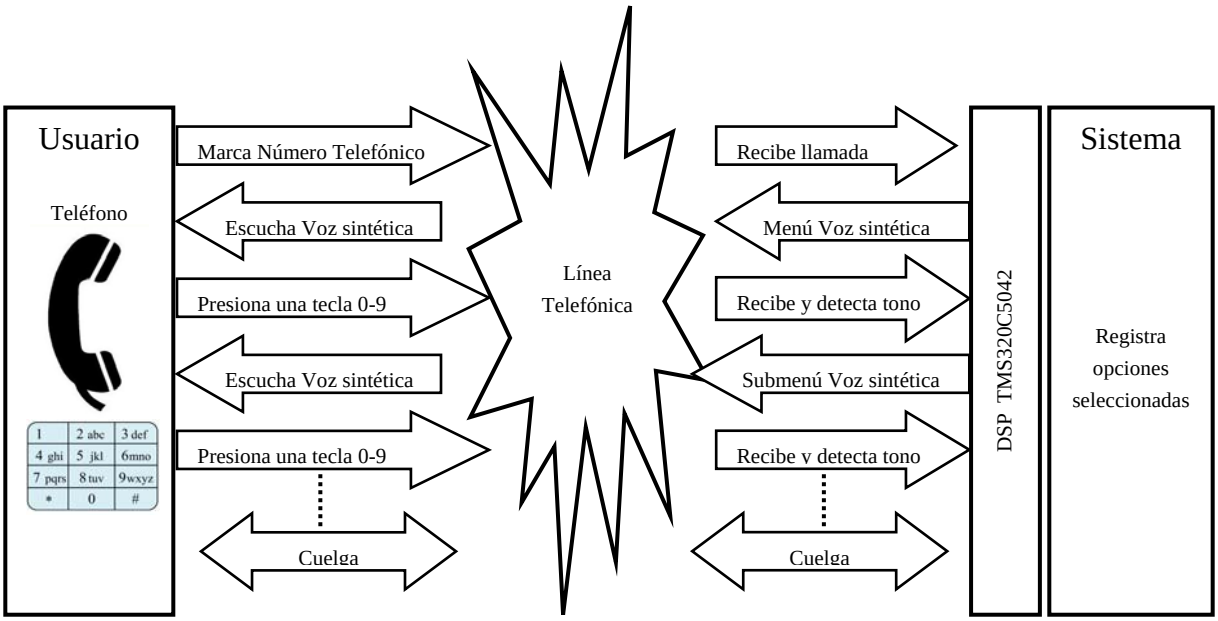



Figura 1.1. Planteamiento del sistema.

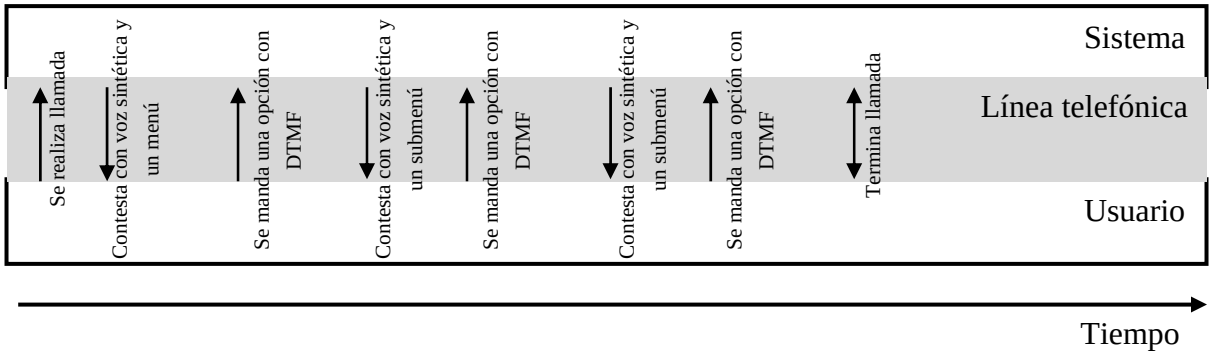



Figura 1.2 Procesos a través del tiempo.

En la figura 1.3 se muestra el diagrama a bloques de la constitución del sistema, éste se compone básicamente de un receptor de llamadas, es decir, se encuentra conectado en el otro extremo de la línea telefónica.

Las funciones que debe realizar son las siguientes:

1. El sistema al iniciar queda en espera de una llamada.

2. Detecta una llamada entrante.

3. Realiza la función de “descolgar”.

4. Con parámetros de voz previamente cargados en memoria, sintetiza voz y lo manda por la línea telefónica.

5. Queda en espera de un tono de DTMF que indica una opción seleccionada por el usuario.

6. Al detectar la presencia de tono lo decodifica a un dígito del 0 al 9.

7. Según el digito decodificado se genera una nueva respuesta con voz sintética y se envía por la línea. La operación descrita desde el punto 4 al 7 se repite dos veces más.

8. Se guardan las opciones seleccionadas por el usuario.

9. El sistema termina la llamada.

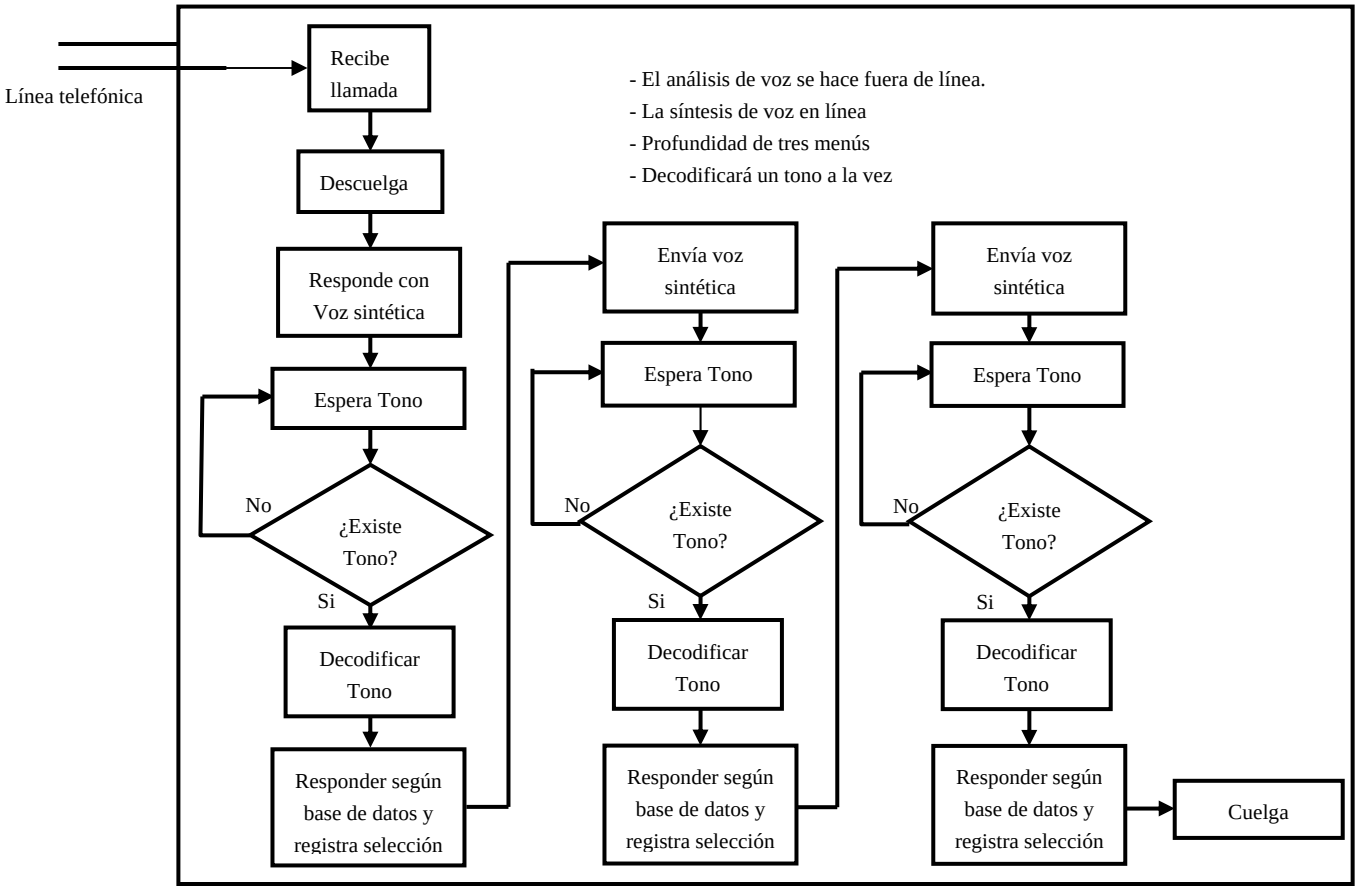



Figura 1.3. Diagrama general del sistema.

## *1.3.1 Metodologías*

Para llevar a cabo dicho proyecto, existen diferentes metodologías de detección de tonos y de síntesis de voz, sin embargo, hay que distinguir cual de todas es la más adecuada dependiendo de sus características.

En lo que respecta a la *detección de tonos* se pueden enumerar los siguientes métodos, junto con sus ventajas y desventajas:

- **Banco de Filtros.**
  - Ventaja: De manera analógica el tiempo de respuesta solía ser pequeño aunque con la velocidad actual de los microprocesadores puede ser superada.
  - Desventajas: Requiere de muchos componentes y son variantes respecto a la temperatura si se desarrolla en forma analógica. En forma digital consume muchos recursos y tiempo de procesamiento.

- **Transformada Discreta de Fourier (DFT).**
  - Ventaja: Se implementa completamente por software por lo que no está expuesta a variables físicas.
  - Desventaja: Requiere de mucho procesamiento ya que debe realizar $\mathbf{N^2}$ operaciones complejas donde **N** es el número de muestras a procesar.

- **Transformada Rápida de Fourier (FFT).**
  - Ventaja. Requiere menos operaciones que la DFT.
  - Desventaja: Aunque requiere menos operaciones que la DTF realiza $\mathbf{(N/2)\log_2 N}$ operaciones complejas. Para implementaciones en tiempo real no resulta ser práctica.

- **Algoritmo de Goertzel**
  - Ventaja: El algoritmo obtiene el espectro de la señal de entrada, pero existen aplicaciones en donde sólo se desean obtener cierto número de valores. En este caso el algoritmo de Goertzel se puede aplicar para calcular sólo los valores deseados [14].
  - Desventaja: Por ser un método recursivo los errores de cálculo se pueden incrementar.

Por otro lado se tiene básicamente dos métodos para formar la *voz sintética*:

- **Concatenación**
  - Ventaja: Se produce una voz más clara, menos "Robótica" y fácil de programar.
  - Desventaja: Requiere de una base de datos para almacenar fonemas para generar frases, la cual requiere mucha memoria.

- **Parametrizada**
  - Ventaja: Requiere de un mínimo de memoria para almacenar los parámetros.
  - Desventaja: Se produce una voz "Robótica" y más difícil en su procesamiento.

### *1.3.2 Método a utilizar*

Para escoger el método que se acople más a nuestras necesidades es necesario conocer nuestros requerimientos que serán expuestos en los siguientes capítulos. El Hardware a utilizar es una tarjeta de desarrollo Development Starter Kit (DSK) de Texas Instruments con el DPS TMS320C5402 cuya memoria está limitada a 16K de 16 bits interna y 256K palabras de memoria externa tipo Flash.

### 1.3.3 Resultados esperados

Lo que pretendemos es diseñar y realizar un sistema capaz de trabajar de forma autónoma en tiempo real, sin requerir de una interfaz con la PC. Las características del sistema son:

- Detectar el tono "Ring" de la línea telefónica.
- Contar con la funcionalidad de "descolgar" para establecer comunicación en la línea telefónica.
- Contestar con Voz Sintética inteligible.
- Detectar los tonos de una manera confiable y con el mínimo de error.
- Almacenar los datos del usuario para un proceso posterior.

### 1.3.4 Estructura de la tesis

El *primer capítulo* comienza con la introducción y motivación del trabajo, seguido del *capítulo dos*, donde se estudia cómo se genera la voz y sus características. En el *capítulo tres* se revisa la forma de generar la voz sintética a partir de parámetros propios de la voz, también se estudia el método de Codificación por Predicción Lineal (LPC). En el *capitulo cuatro* se explica el funcionamiento del sistema telefónico que será nuestro canal de comunicación. Una de las formas en la cual se puede transmitir información a través de la línea telefónica es por medio de Multi-frecuencias de doble tono (DTMF) el cual se verá a más detalle en *capítulo cinco*. En el *capítulo seis* se describe las herramientas utilizadas y el software para la aplicación en tiempo real. En los *capítulos siete y ocho* se dan los resultados obtenidos y las *conclusiones*.

# CAPITULO DOS

# 2. Producción de voz

Desde la prehistoria la comunicación por medio del habla ha sido el modo dominante en las sociedades humanas para el intercambio de información. En la actualidad la palabra hablada se ha extendido a través de tecnologías como es el teléfono, cine, radio, televisión e Internet.

En la interacción humano-humano se prefiere una comunicación por medio del lenguaje hablado y de forma similar se busca en la comunicación humano-máquina. La mayoría de las computadoras frecuentemente utilizan una Interfaz Gráfica de Usuario (GUI), basado en interfaces con representaciones gráficas de objetos y funciones como ventanas, iconos, menús e indicaciones; sin embargo, la mayoría de los sistemas operativos de computadoras y aplicaciones dependen también de un teclado, mouse y un monitor para la retroalimentación. Actualmente las computadoras carecen de las habilidades humanas para hablar, escuchar, entender y leer. El habla en un futuro será uno de los medios primarios de la interfaz con computadoras, sostenido por otras modalidades naturales, y antes de alcanzar la total madurez en la interacción basada en el habla; las aplicaciones en el hogar, en dispositivos móviles, y en la oficina se están incorporando a la tecnología hablada del lenguaje para cambiar la manera como vivimos y trabajamos.

Un sistema de lenguaje hablado necesita tener reconocimiento y síntesis del habla, sin embargo, estos dos componentes por sí mismos no son suficientes para construir un sistema de lenguaje hablado útil. La comprensión y diálogo requieren manejar interacciones con el usuario, dominio y conocimiento del lenguaje similar, por lo que deberá proporcionar una guía de sistemas de interpretación del discurso que le permita determinar la acción apropiada. El propósito de construir comercialmente sistemas de lenguaje hablados viables han atraído la atención de científicos e ingenieros de todo el mundo [2].

Para producir la voz es necesario entender cómo se genera, la voz se modela por medio de un sistema capaz de reproducir el funcionamiento de la glotis y del tracto vocal por medio de parámetros que lo representan. En este capítulo analizamos los mecanismos físicos de la producción de la voz y su modelado digital para posteriormente realizar el análisis para obtener sus parámetros y por último la reconstrucción de la señal de voz por medio de la síntesis.

## 2.1 Generación de voz

La voz es la base de la comunicación habitual del ser humano, con la que se transmite la cultura, se expresan los sentimientos y las emociones. Por su cotidianidad, muchas veces pasa desapercibida su extraordinaria importancia, sin embargo por su carácter específico y exclusivamente humano ha sido estudiada desde los inicios de nuestra civilización [3].

La comunicación del lenguaje hablado se puede representar en términos de *mensaje, contenido o información* donde se considera como una representación abstracta de una idea a transmitir desde el *emisor* o *hablante* a un *receptor*. En los sistemas de comunicación de voz los principales objetivos son:

1. Preservación del mensaje contenido en la señal del habla.
2. Representación de la señal del habla en forma conveniente para la transmisión, y almacenamiento de tal forma que sea flexible para realizar modificaciones sin causar una degradación seria del contenido del mensaje [1].

La voz es producida por una secuencia de sonidos, la transición entre esos sonidos sirven como símbolos sonoros para generar una representación de información. Los arreglos de estos sonidos están sometidos por reglas del lenguaje [1].

El sonido son ondas longitudinales de compresión y expansión de las moléculas del aire en dirección paralela a la aplicación de la energía como se representa en la figura 2.1 [2], en 2.1 a) podemos ver que la parte más obscura representa mayor compresión del aire y mientras esa presión disminuye y se vuelve más clara, en 2.1 b) se representa en forma de onda donde la mayor presión nos da mayor amplitud de la onda, entre cresta y cresta nos resultara la longitud de onda si se trata de un sonido periódico.

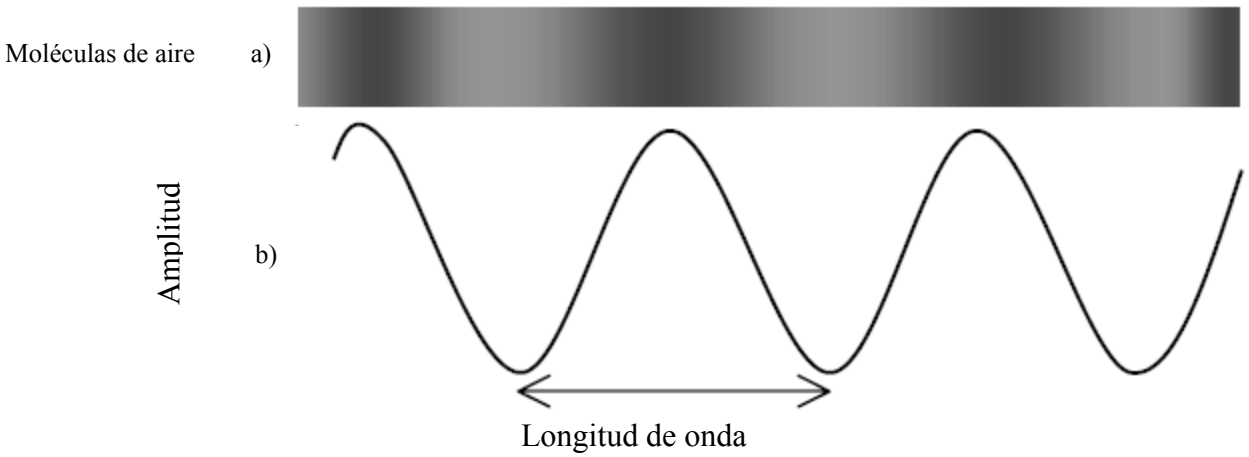



Figura 2.1. Producción de sonido.

## 2.1.1 Producción del habla

La voz humana se produce por medio del aparato fonatorio, formado por los pulmones, como fuente de energía en la forma de un flujo de aire, la *laringe*, que contiene las *cuerdas vocales*, la *faringe*, las *cavidades oral* (o bucal) y *nasal* y una serie de elementos articulatorios: los *labios*, los *dientes*, el *alvéolo*, el *paladar*, el *velo del paladar* y la *lengua* como muestra la figura 2.2.

Las *cuerdas vocales*, son en realidad dos membranas dentro de la laringe orientadas de adelante hacia atrás como se ve en la figura 2.3. Por adelante se unen en el *cartílago tiroides* (que puede palparse sobre el cuello, inmediatamente por debajo de la unión con la cabeza; en los varones suele apreciarse como una protuberancia conocida como *nuez de Adán*). Por detrás, cada cuerda está sujeta a uno de los dos *cartílagos aritenoides*, los cuales pueden separarse voluntariamente por medio de músculos, la abertura entre ambas cuerdas se denomina *glotis* [3].

Cuando las cuerdas vocales se encuentran separadas, la glotis adopta una forma triangular. El aire pasa libremente y prácticamente no se produce sonido. Es el caso de la respiración. Cuando la glotis comienza a cerrarse, el aire que la atraviesa proveniente de los pulmones experimenta una turbulencia, emitiéndose un ruido de origen aerodinámico conocido como *aspiración* (aunque en realidad acompaña a una espiración o exhalación). Esto sucede en los sonidos denominados "aspirados" (como la h inglesa).

Al cerrarse más, las cuerdas vocales comienzan a vibrar a modo de lengüetas, produciéndose un sonido tonal, es decir periódico. La frecuencia de este sonido depende de varios factores, entre otros del tamaño y la masa de las cuerdas vocales, de la tensión que se les aplique y de la velocidad del flujo del aire proveniente de los pulmones. A mayor tamaño, menor frecuencia de vibración, lo cual explica por qué en los varones, cuya glotis es en promedio mayor que la de las mujeres, la voz es en general más grave. A mayor tensión la frecuencia aumenta, siendo los sonidos más agudos. Así, para lograr emitir sonidos en el registro extremo de la voz es necesario un mayor esfuerzo vocal. También aumenta la frecuencia (a igualdad de las otras condiciones) al crecer la velocidad del flujo de aire, razón por la cual al aumentar la intensidad de emisión se tiende a elevar espontáneamente el tono de voz.

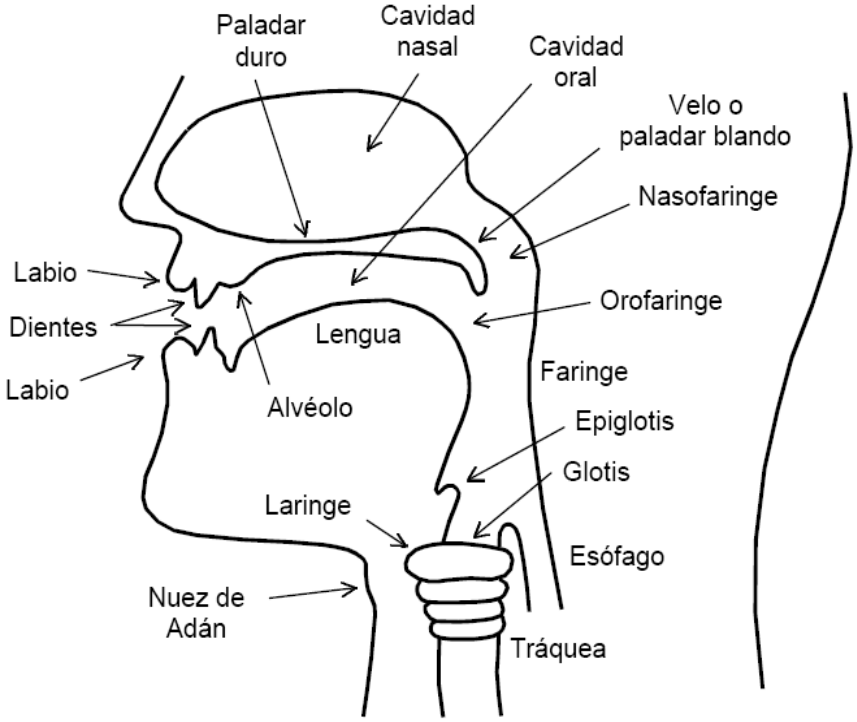



Figura 2.2. Aparato productor de voz

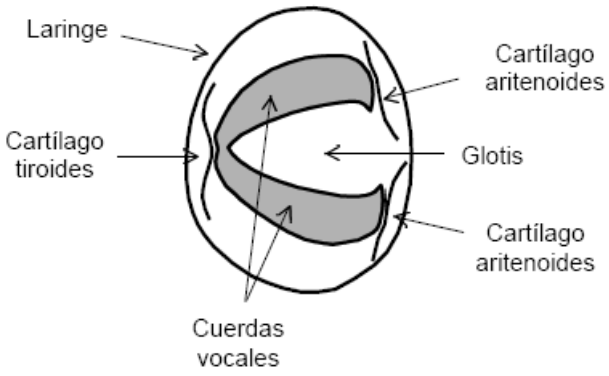



Figura 2.3. Corte transversal de la laringe

Finalmente, es posible obturar la glotis completamente, en ese caso no se produce sonido. Sobre la glotis se encuentra la *epiglotis*, un cartílago en la faringe que permite tapar la glotis durante la deglución para evitar que el alimento ingerido se introduzca en el tracto respiratorio. Durante la respiración y la fonación (emisión de sonido) la epiglotis está separada de la glotis permitiendo la circulación del flujo de aire. Durante la deglución, en cambio, la laringe ejecuta un movimiento ascendente de modo que la glotis apoya sobre la epiglotis.

La porción que incluye las cavidades faríngea, oral y nasal junto con los elementos articulatorios se denomina genéricamente *cavidad supraglótica*, en tanto que los espacios por debajo de la laringe, es decir la tráquea, los bronquios y los pulmones, se denominan *cavidades infraglóticas*.

Varios de los elementos de la cavidad supraglótica se controlan a voluntad, permitiendo modificar dentro de márgenes muy amplios los sonidos producidos por las cuerdas vocales o agregar partes distintivas a los mismos, e inclusive producir sonidos propios. Todo esto se efectúa por dos mecanismos principales: el *filtrado* y la *articulación*.

- El *filtrado* actúa modificando el espectro del sonido, tiene lugar en las cuatro cavidades supraglóticas principales: la faringe, la cavidad nasal, la cavidad oral y la cavidad labial. Las mismas constituyen resonadores acústicos que enfatizan determinadas bandas del espectro generado por las cuerdas vocales, conduciendo al concepto de *formantes*, es decir, una serie de picos de resonancia ubicados en frecuencias o bandas de frecuencia que son bastante específicas para cada tipo de sonido.

- La *articulación,* es una modificación principalmente a nivel temporal de los sonidos, y está directamente relacionada con la emisión de los mismos y con los fenómenos transitorios que los acompañan. Está caracterizada por el lugar del tracto vocal en que tiene lugar, por los elementos que intervienen y por el modo en que se produce, factores que dan origen a una clasificación fonética de los sonidos [3].

## 2.2 Características de la voz

Los sonidos emitidos por el aparato fonatorio pueden clasificarse de acuerdo a diversos criterios que toman en cuenta los diferentes aspectos del fenómeno de emisión que son:

a) Carácter vocálico o consonántico
b) Oralidad o nasalidad
c) Carácter tonal (sonoro) o no tonal (sordo)
d) El lugar de articulación
e) El modo de articulación
f) La posición de los órganos articulatorios
g) La duración

### *2.2.1 Vocales y consonantes*

Desde un punto de vista mecano-acústico, las *vocales* son los sonidos emitidos por la sola vibración de las cuerdas vocales sin ningún obstáculo o constricción entre la laringe y las aberturas oral y nasal. Dicha vibración se genera por el principio del oscilador de relajación, donde interviene una fuente de energía constante en la forma de un flujo de aire proveniente de los pulmones. Son siempre sonidos de carácter tonal (cuasi-periódicos), y por consiguiente de espectro discreto. Las *consonantes*, por el contrario, se emiten interponiendo algún obstáculo formado por los elementos articulatorios. Los sonidos correspondientes a las consonantes pueden ser tonales o no, dependiendo de si las cuerdas vocales están vibrando o no. Funcionalmente, en el castellano las vocales pueden constituir palabras completas, no así las consonantes [3].

### *2.2.2 Oralidad y nasalidad*

Los fonemas producidos por aire que pasa por la cavidad nasal se denominan *nasales*, en tanto que aquéllos en los que sale por la boca se denominan *orales*. La diferencia principal está en el tipo de resonador principal por encima de la laringe (cavidad nasal y oral, respectivamente). En castellano son nasales sólo las consonantes “m”, “n” y “ñ”.

### *2.2.3 Tonalidad*

Los fonemas en los que participa la vibración de las cuerdas vocales se denominan *tonales* o, también, *sonoros*. La tonalidad lleva implícito un espectro cuasi-periódico. Como se puntualizó anteriormente, todas las vocales son tonales, pero existen varias consonantes que también lo son por ejemplo “b”, “d”, “m”. Aquellos fonemas producidos sin vibraciones glotales se denominan *sordos*. Varios de ellos son el resultado de la turbulencia causada por el aire pasando a gran velocidad por un espacio reducido, como las consonantes “s”, “z”, “j” y “f” [3].

### 2.2.4 Lugar y modo de articulación (consonantes)

La *articulación* es el proceso mediante el cual alguna parte del aparato fonatorio, interpone un obstáculo para la circulación del flujo de aire. Las características de la articulación permitirán clasificar las consonantes. Los órganos articulatorios son los labios, los dientes, las diferentes partes del paladar (alvéolo, paladar duro, paladar blando o velo), la lengua y la glotis. Salvo la glotis, que puede articular por sí misma, el resto de los órganos articula por oposición con otro, según el lugar o punto de articulación se tienen los fonemas [3]:

- *Bilabiales*: oposición de ambos labios
- *Labiodentales*: oposición de los dientes superiores con el labio inferior
- *Linguodentales*: oposición de la punta de la lengua con los dientes superiores
- *Alveolares*: oposición de la punta de la lengua con la región alveolar
- *Palatales*: oposición de la lengua con el paladar duro
- *Velares*: oposición de la parte posterior de la lengua con el paladar blando
- *Glotales*: articulación en la propia glotis

A su vez, para cada punto de articulación, puede efectuarse de diferentes modos, dando lugar a fonemas:

- *Oclusivos*: la salida del aire se cierra momentáneamente por completo
- *Fricativos*: el aire sale atravesando un espacio estrecho
- *Africados*: oclusión seguida por fricación
- *Laterales*: la lengua obstruye el centro de la boca y el aire sale por los lados
- *Vibrantes*: la lengua vibra cerrando el paso del aire intermitentemente
- *Aproximantes*: la obstrucción muy estrecha que no llega a producir turbulencia

Los fonemas oclusivos (correspondientes a las consonantes “b” inicial, “c”, “k”, “d”, “g”, “p” y “t”) también se denominan *explosivos*, debido a la liberación repentina de la presión presente inmediatamente antes de su emisión. Pueden ser sordos o sonoros, al igual que los fricativos (“b” intervocálica, “f”, “j”, “h” aspirada, “s”, “y”, “z”). Sólo existe un fonema africado en castellano, correspondiente a la “ch”. Los laterales (“l”, “ll”) a veces se denominan líquidos, y son siempre sonoros. Los dos fonemas vibrantes del castellano (consonantes “r”, “rr”) difieren en que en uno de ellos (“r”) se ejecuta una sola vibración y es intervocálico, mientras que en el otro (“rr”) es una sucesión de dos o tres vibraciones de la lengua. Finalmente, los fonemas aproximantes (la “i” y la “u” cerradas que aparecen en algunos diptongos) son a veces denominados semivocales, pues en realidad suenan como vocales. Pero exhiben una diferencia muy importante: son de corta duración y no son prolongables. En la tabla 2.1 se indican las consonantes clasificadas según el lugar y el modo de articulación, la sonoridad, la oralidad y nasalidad. En algunos casos una misma consonante aparece en dos categorías diferentes, correspondiente a las diferencias observadas [3].

| Lugar de Articulación | Modo de articulación | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Oral | | | | | | | | Nasal |
| | Oclusiva | | Fricativa | | Africada[1] | Lateral | Vibrante | Aproximante | Sonora |
| | Sorda | Sonora | Sorda | Sonora | Sorda | Sonora | Sonora | Sonora | |
| Bilabial | P | B,V | | B,V | | | | W | M |
| Labiodental | | | F | | | | | | |
| Linguodental | | | Z | | | | | | |
| Alveolar | T | D | S | Y | CH | L | R,RR | | N |
| Palatal | | | | (Y) | (CH) | LL | | I | Ñ |
| Velar | K | G | J | | | | | | |
| Glotal | | | H | | | | | | |

[1] Un sonido africado de produce con una obstrucción total de la corriente de aire y de una inmediata apertura leve que permite la salida continua de aire; es decir, es la combinación de una oclusiva y una fricativa [18].

Tabla 2.1. Clasificación de consonantes según el modo y lugar de articulación y sonoridad.

## 2.2.5 Posición de los órganos articulatorios (vocales)

En el caso de las vocales, la articulación consiste en la modificación de la acción filtrante de los diversos resonadores, lo cual depende de las posiciones de la lengua (tanto en elevación como en profundidad o avance), de la mandíbula inferior, de los labios y del paladar blando. Estos órganos influyen sobre los formantes, permitiendo su control. Podemos clasificar las vocales según la posición de la lengua como se muestra en la tabla 2.2.

| Posición vertical | Tipo de vocal | Posición horizontal (avance) | | |
|---|---|---|---|---|
| | | Anterior | Central | Posterior |
| Alta | Cerrada | i | | U |
| Media | Media | e | | O |
| Baja | Abierta | | a | |

Tabla 2.2. Clasificación de las vocales según la posición de la lengua.

Otra cualidad controlable es la *labialización*, es decir el hecho de que se haga participar activamente los labios. Las vocales labializadas, también definidas como *redondeadas*, son las que redondean los labios hacia adelante, incrementando la longitud efectiva del tracto vocal. La única vocal labializada en el castellano es la "u". En otros idiomas, como el francés, el portugués, el catalán y el polaco, así como en lenguas no europeas como el guaraní o el hindi, existe también el matiz de oralidad o nasalidad. En las vocales orales el velo (paladar blando) sube, obturando la nasofaringe, lo cual impide que el aire fluya parcialmente por la cavidad nasal. En las vocales *nasalizadas* (u *orales-nasales*) el velo baja, liberando el paso del aire a través de la nasofaringe. Se incorpora así la resonancia nasal [3].

### 2.2.6 Duración

La duración de los sonidos, especialmente de las vocales, no tiene importancia a nivel semántico en el castellano, pero sí en el plano expresivo, es decir el énfasis o acentuación a través de la duración. En inglés, en cambio, la duración de una vocal puede cambiar completamente el significado de la palabra que la contiene.

### 2.2.7 Sonido voceado y no voceado

Los sonidos producidos por el aparato productor de voz humano pueden ser divididos según se utilicen las diferentes partes del sistema como son: nasales, guturales, labiodental etc. Pero la diferencia más importante es si el sonido es voceado o no voceado. Los sonidos *voceados*, como las vocales, son aquellos que tienen una estructura periódica en el transcurso del tiempo del sonido, a diferencia de los *no voceados* en donde esta estructura es nula, típicamente los sonidos voceados contienen mayor energía que los sonidos no voceados. En la figura 2.4 se muestra un ejemplo de los dos tipos de sonidos [2].

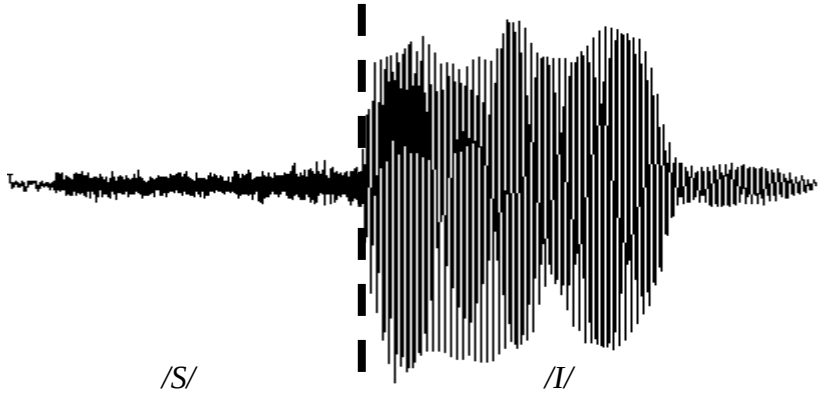



Figura 2.4. Forma de onda de un sonido no voceado como la /*S*/, uno voceado como la /*I*/.

## 2.3 Análisis de la generación de voz

Las señales de voz son producidas cuando un volumen de aire desde los pulmones excita el conducto vocal, que se comporta como una cavidad resonante. El conducto vocal es usualmente modelado como la concatenación de tubos acústicos sin pérdidas, con distintas secciones transversales, que comienza en las cuerdas vocales y termina en los labios como se observa en la figura 2.5. La apertura de las cuerdas vocales se denomina *glotis*. Los diferentes sonidos genéricamente pueden ser clasificados en: sonidos tonales (en inglés *voiced*, como el de las vocales) y sonidos no tonales (en inglés *unvoiced*, como por ejemplo el de una 's' final de palabra) [4].

## 2.3.1 Modelo acústico de producción de voz

Para aplicar una técnica de procesamiento digital de señal al problema de producción de voz es importante entender los fundamentos de cómo se produce la voz [1]. La teoría acústica analiza las leyes físicas que gobiernan la propagación del sonido en el tracto vocal, de acuerdo a esta teoría se debería considerar la propagación de onda en las tres dimensiones, la variación de la forma del tracto vocal en cada instante de tiempo, pérdidas por la fricción dependiendo la temperatura y viscosidad de la paredes del tracto vocal, radiación de sonido a través del los labios, acoplamiento nasal y excitación del sonido. Por ahora no existe un modelo que considere todas las variables, sin embargo, existen modelos que proveen una buena aproximación en la práctica, también involucran un buen entendimiento de la física que involucra [2].

## 2.3.2 Modelo de tubos concatenados sin pérdidas

Un modelo basado en el tracto vocal y muy usado para la producción de voz puede ser representado como una concatenación de tubos sin pérdida como muestra la figura 2.5 [2].

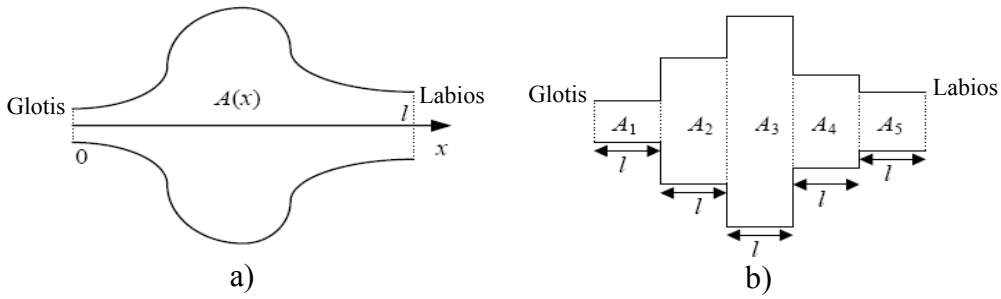



Figura 2.5. a) Esquema del tracto vocal; b) Aproximación del tubo como una concatenación variando el área.

El tracto vocal se puede modelar como un tubo no uniforme con sección transversal variante con el tiempo, para frecuencias correspondientes a la longitud de onda comparables a la dimensión del tracto vocal (menor a 4000Hz), es razonable asumir que el plano de propagación de la onda es a lo largo del eje del tubo. Además se asume que no hay pérdidas de conducción por la viscosidad o temperatura de la cantidad de fluido o entre las paredes del tubo.

## 2.3.3 Efecto de radiación de los labios

La onda de presión de la señal de voz está relacionada con la onda volumen-velocidad presente en los labios a través de una impedancia de radiación **R(z)**. Se considera que esta impedancia no varía según el sonido que se produce, por lo que el efecto de los labios se modela como un diferenciador fijo simple [5] como muestra la ecuación (2.1). Este modelo realiza también la función de filtro pasa alta con una frecuencia de corte aproximado de 3.8KHz y se conoce como pre-énfasis, definido en el dominio de la transformada Z como

$$S(z) = 1 - \alpha z^{-1} \quad (2.1)$$

Por otro lado el modelo de codificación de predicción lineal (LPC) es eficiente para modelar frecuencias bajas, pero pobre para frecuencias altas. Para prevenir este comportamiento la señal se pasa por el filtro de pre-énfasis que enfatiza las frecuencias altas antes de ser procesada. El valor típico para el valor de $\alpha$ está alrededor de 0.9, el que usualmente se usa es $\alpha = 15/16 = 0.9375$. En forma temporal donde **s(n)** representa la señal en forma discreta, el filtro de pre-énfasis se muestra en la ecuación (2.2) [17].

$$s'(n) = s(n) - 0.9375\, s(n-1) \quad (2.2)$$

Después del procesamiento en el sintetizador, la señal es pasada por un filtro de de-énfasis como muestra la ecuación (2.3) [17].

$$s'(n) = s(n) + 0.9375\, s(n-1) \quad (2.3)$$

### 2.3.4 Excitación del sonido en el tracto vocal

En esta sección se considera el mecanismo por el cual las ondas sonoras son generadas en el sistema de tracto vocal donde se identifican tres mecanismos de excitación principales. Estos mecanismos son [1]:

1. El flujo de aire es modulado por las vibraciones de las cuerdas vocales resultando una excitación parecida a un pulso cuasi-periódico.
2. El flujo del aire es turbulento dado que el aire pasa a través de una constricción en el tracto vocal, resultando en una excitación parecida al ruido.
3. El flujo de aire se genera por presión detrás de un punto en el tracto vocal que está totalmente cerrado. La rápida liberación de esta presión, por remover la constricción, causa una excitación transitoria.

El modelo de producción de voz toma en cuenta los dos primeros mecanismos de excitación los cuales se nombran *voceado* para el primero y *no voceado* para el segundo.

### 2.3.5 Modelo general de síntesis de voz

Existen limitaciones respecto al modelo simplificado de síntesis de voz representado en la figura 2.6 que afortunadamente no son impedimento para aplicarlo en muchos casos.

*Primero*, la variación de parámetros en sonidos continuos como las vocales, los cambios de parámetros es muy lento y el modelo trabaja adecuadamente. Con los sonidos transitorios el modelo no es tan bueno pero permanece con un comportamiento aceptable. Cabe destacar que el uso de la función de transferencia y función de respuesta de frecuencia implícitamente asume que puede representar la señal de voz en un "tiempo corto". Los parámetros del modelo deben

ser constantes sobre intervalos de tiempo cortos, típicamente entre 10 a 20 milisegundos, con ello se puede considerar la señal de voz con un comportamiento estacionario. La función de transferencia **H(z)** realmente sirve para definir la estructura del modelo cuyos parámetros varían lentamente en el tiempo.

Una *segunda* limitación es que carece de una provisión de ceros que son requeridos teóricamente para sonidos nasales y fricativos. Es definitivamente una limitación para los sonidos nasales pero no es severo para sonidos fricativos.

*Tercera*, la simple discriminación de voceado y no voceado es inadecuado para un sonido fricativo ya que está relacionado con los picos del flujo de aire glotal; una desventaja relativa menor de este modelo del la figura 2.6 es que requiere de pulsos glotales espaciados por un entero múltiplo del periodo de muestreo *T* por lo que el control del tono (*Pitch*) no es preciso [1] [6].

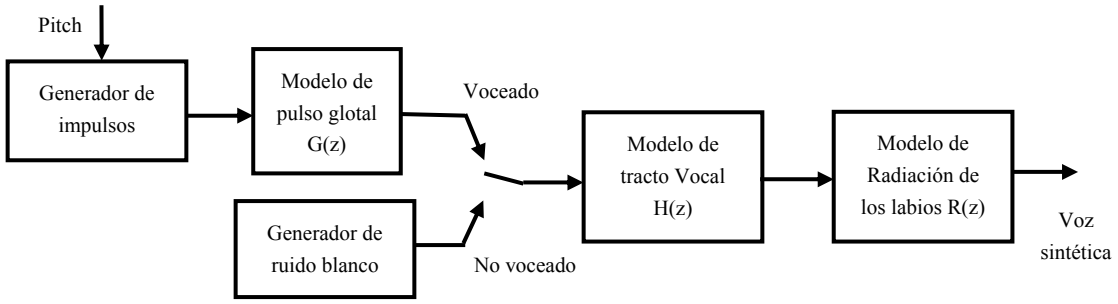



Figura 2.6. Modelo General de producción de voz.

En la figura 2.6 el generador de impulsos se necesita si la trama de voz es clasificado como voceado, entonces el tono (pitch que se explica a detalle más adelante) es representado por un tren de impulsos espaciados con el periodo encontrado en el tono. Si la trama de voz es clasificada como no voceado entonces la excitación es generada por ruido blanco.

La función de transferencia que permite modelar en un intervalo corto de tiempo el tracto vocal **H(z)** se muestra en la ecuación (2.4).

$$\mathbf{H(z) = \frac{G}{1 + \sum_{k=1}^{N} a_k z^{-k}} = \frac{G}{\prod_{k=1}^{N} (1 - p_k z^{-1})}} \qquad (2.4)$$

Donde **G** representa la ganancia global, $\mathbf{p_k}$ la ubicación de los polos complejos para el modelo de N-tubos (modelo acústico de producción de voz) y $\mathbf{a_k}$ el conjunto de coeficientes del filtro [6].
Cada ubicación de los pares de polos conjugados en el plano *Z*, de manera aproximada, corresponde a un formante en el espectro de **H(z)**. Para que **H(z)** sea estable, entonces todos los polos deberán estar ubicados dentro de la circunferencia unitaria en el dominio de la transformada *Z* [6].

En el modelo de la figura 2.6 final la señal reconstruida pasa por el Modelo de radiación de los labios **R(z)** o también se conoce como filtro de de-énfasis.

Para llevar a cabo la síntesis de voz es necesario realizar una análisis de la fuente de voz, con ello se obtienen los parámetros necesarios para poder reconstruir la señal de forma que sea lo más cercana a la original e inteligible.

### *2.3.6 Ventaneo de la señal de voz*

El método de predicción lineal es aplicable a señales estacionarias, señales cuyo comportamiento no cambia durante el tiempo, sin embargo, éste no es el caso de la señal de voz. Para poder aplicar el método de Predicción Lineal (LP) la señal es segmentada en pequeños bloques llamados *tramas* o *ventanas* que presentan un comportamiento cuasi-estacionario [17]. En la figura 2.7 se muestra en a) la señal y la aplicación de la ventana de Hamming en tres tramos diferentes y en b) el efecto que tiene en la señal al multiplicar la señal original y la ventana.

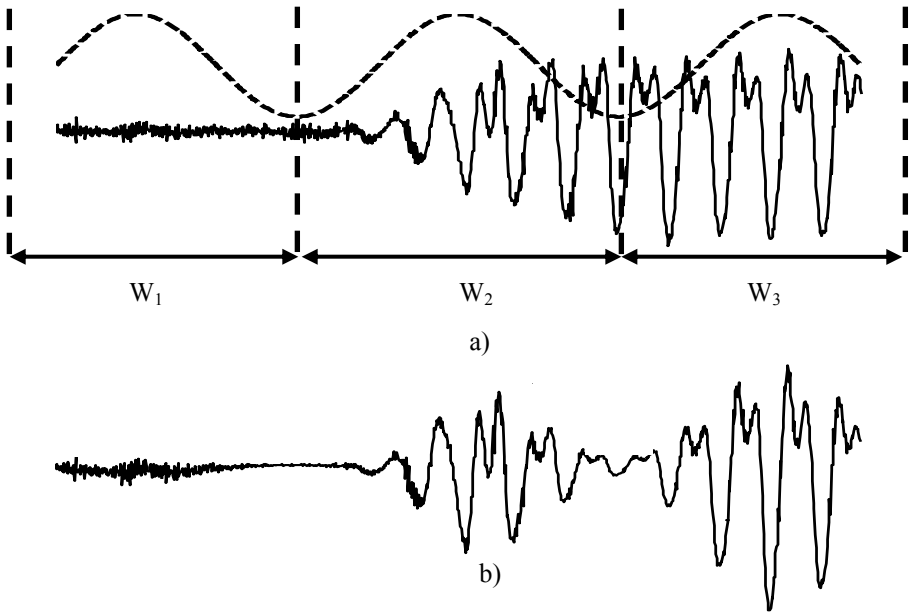



Figura 2.7. a) Señal original y ventana, b) Señal resultante al aplicarle una ventana.

Al multiplicar la ventana por la señal se produce el efecto de segmentación donde en los extremos de la ventana se tiene una atenuación de la señal evitando de esta manera cortes bruscos que generan componentes de alta frecuencia. Fuera de los intervalos de esta multiplicación la señal es cero. El ventaneo es necesario en el método de autocorrelación del análisis de LPC que se analizará en el siguiente capítulo. Se le conoce como $N$ al tamaño en muestras de la ventana y generalmente se escoge entre el intervalo de 20-40 milisegundos donde se considera que el bloque tiene un comportamiento ergódico [17].

Para realizar un análisis de la trama de la señal **s(n)**, esta se multiplica por una ventana de análisis de tamaño limitado **w(n)**, para extraer un segmento en particular en un tiempo definido, y con ello se logra que **s(n)** sea cero afuera del intervalo de interés, este proceso es llamado comúnmente *ventaneo*. El escoger la forma correcta de la ventana es muy importante porque esto permite que diferentes muestras puedan tener un peso diferente. En la figura 2.8 se muestra de forma temporal la gráfica de las ventanas más utilizadas.

A continuación se muestran las definiciones de las ventanas más comunes de longitud **N** [9]:

**Rectangular**

$$\mathbf{w(n)} = \begin{cases} 1 & \mathbf{0 \le n \le N-1} \\ 0 & \text{otro caso} \end{cases} \tag{2.5}$$

**Bartlett**

$$\mathbf{w(n)} = \begin{cases} \dfrac{\mathbf{2n}}{\mathbf{N-1}} & \mathbf{0 \le n \le \dfrac{N-1}{2}} \\ \mathbf{2 - \dfrac{2n}{N-1}} & \mathbf{\dfrac{N-1}{2} \le n \le N-1} \\ 0 & \text{otro caso} \end{cases} \tag{2.6}$$

**Hanning**

$$\mathbf{w(n)} = \begin{cases} \mathbf{0.5 - 0.5 \cos\left(2\pi \dfrac{n}{N-1}\right)} & \mathbf{0 \le n \le N-1} \\ 0 & \text{otro caso} \end{cases} \tag{2.7}$$

**Hamming**

$$\mathbf{w(n)} = \begin{cases} \mathbf{0.54 - 0.46 \cos\left(2\pi \dfrac{n}{N-1}\right)} & \mathbf{0 \le n \le N-1} \\ 0 & \text{otro caso} \end{cases} \tag{2.8}$$

**Blackman**

$$\mathbf{w(n)} = \begin{cases} \mathbf{0.42 - 0.5 \cos\left(2\pi \dfrac{n}{N-1}\right) + 0.08 \cos\left(2\pi \dfrac{2n}{N-1}\right)} & \mathbf{0 \le n \le N-1} \\ 0 & \text{otro caso} \end{cases} \tag{2.9}$$

**Kaiser**

$$w(n) = \begin{cases} \dfrac{I_0\left[\beta\sqrt{1-\left(\frac{2n}{N-1}-1\right)^2}\right]}{I_0(\beta)} & 0 \le n \le N-1 \\ 0 & \text{otro caso} \end{cases} \qquad (2.10)$$

Donde $I_0$ se puede calcular de la función modificada de Bessel de orden cero.

$$I_0(x) = 1 + \sum_{k=1}^{\infty}\left[\frac{\left(\frac{x}{2}\right)^k}{k!}\right]^2$$

$$\beta = \begin{cases} 0.1102(A_s - 8.7) & A_s > 50dB \\ 0.5842(A_s - 21)^{0.4} + 0.07886(A_s - 21) & 21dB \le A_s \le 50dB \\ 0.0 & A_s < 21dB \end{cases}$$

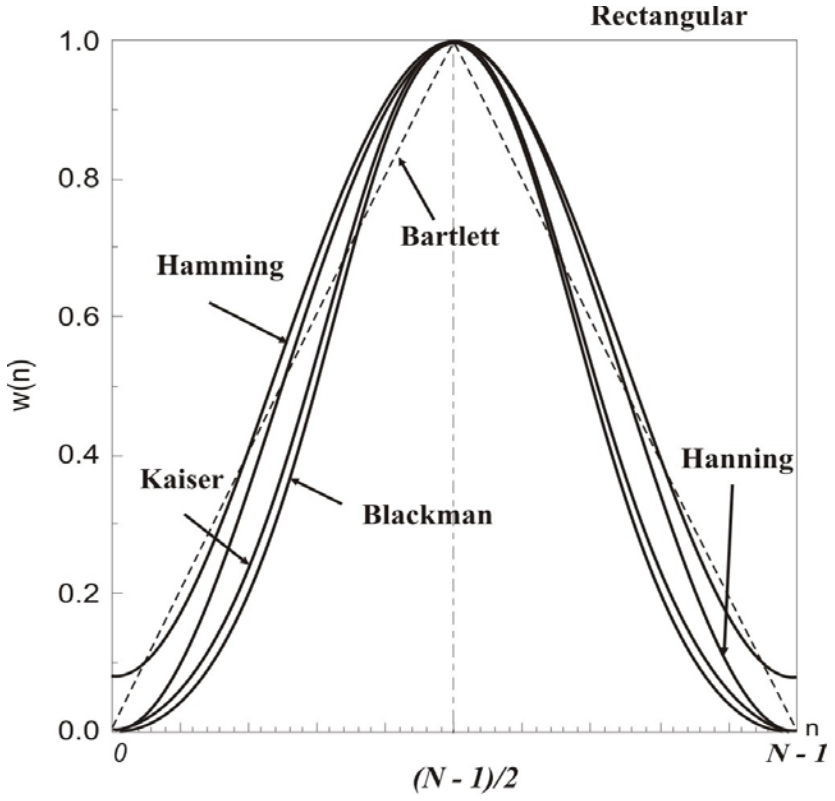



Figura 2.8 Gráfica en el tiempo de varias ventanas.

A continuación en la figura 2.9 se muestran los espectros de las ventanas más comunes donde se pueden observar las diferencias entre ellas como son el ancho del lóbulo principal, la atenuación del lóbulo lateral respecto al principal y cantidad de rizos. De forma resumida tenemos la tabla 2.4 donde nos muestra estas diferentes características de las ventanas. Cabe mencionar que las ventanas más utilizadas son la rectangular, Hanning y Hamming.

| Ventana | Ancho del lóbulo principal de W(ω) | Atenuación del Lóbulo principal a 2do. Lóbulo ($dB_s$) | $A_s$ mínima ($dB_s$) |
|---|---|---|---|
| Rectangular | 4π/N | -13 | -21 |
| Triangular | 8π/N | -26 | -25 |
| Hanning | 8π/N | -31 | -44 |
| Hamming | 8π/N | -41 | -53 |
| Blackman | 12π/N | -57 | -74 |

Tabla 2.4. Desempeño de ventanas [11].

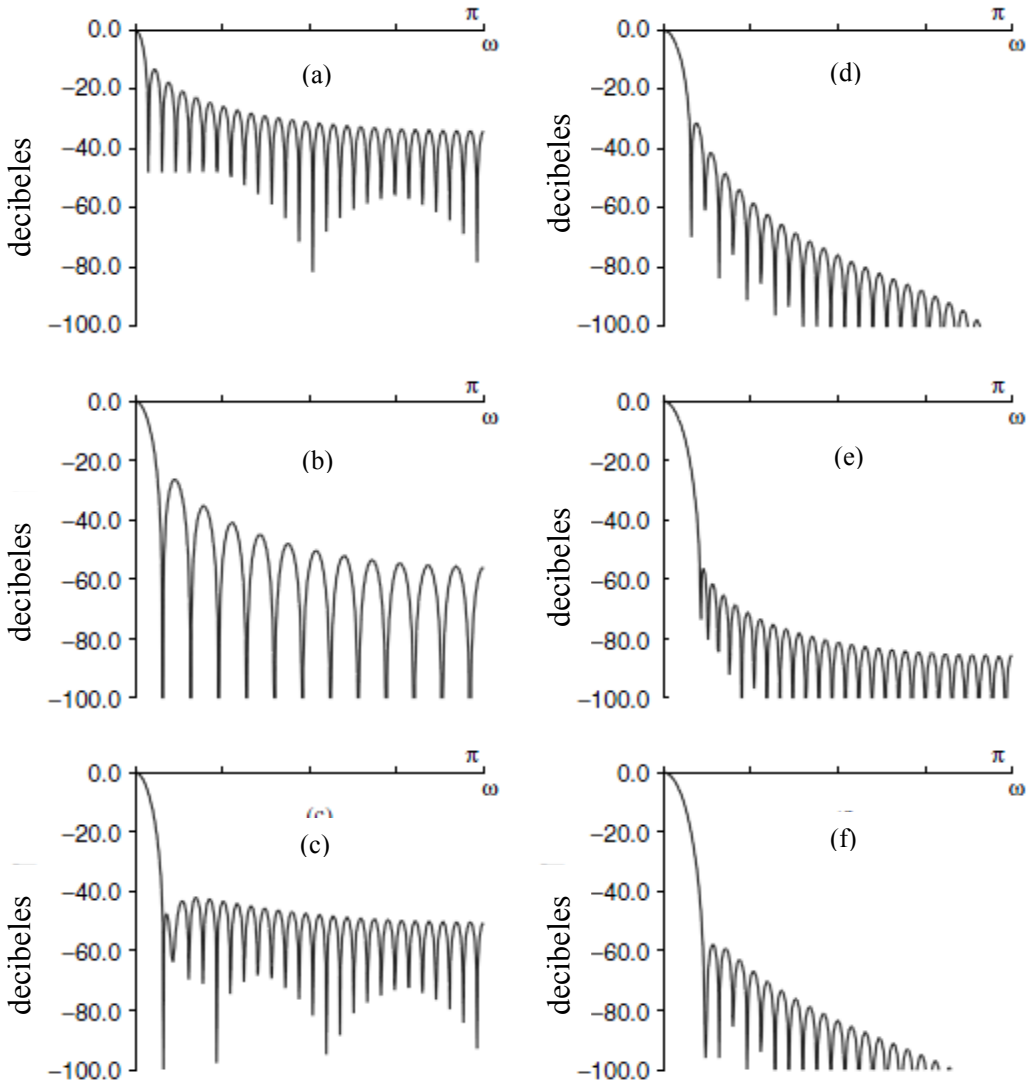



Figura 2.9 Espectro de las ventanas más comunes: (a) Rectangular, (b) Bartlett, (c) Hamming, (d) Hanning, (e) Kaiser β = 7.8, y (f) Blackman.

## 2.4 Representación de la señal de voz

Existen varias representaciones de la señal de voz que son útiles para la codificación, síntesis y reconocimiento. El tema central es la descomposición de la señal de voz como resultado de pasarlo por un filtro lineal variante en el tiempo, este filtro puede ser derivado de los modelos de producción de voz basado en la teoría acústica donde la fuente representa el flujo del aire en las cuerdas vocales, y el filtro representa la resonancia del tracto vocal el cual cambia a través del tiempo. El modelo de fuente-filtro ilustrado en la figura 2.10 se describe el método de estimar la fuente o excitación **e[n]** y el filtro **h[n]** a partir de la señal de voz **x[n]** [2].

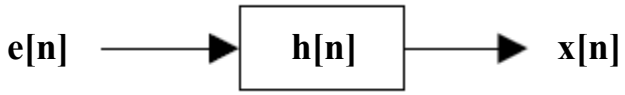



Figura 2.10. Modelo básico fuente-filtro para señales de voz.

### *2.4.1 Análisis de tiempo-corto de Fourier*

La representación de señales u otras funciones como sumas de senoidales o exponenciales complejos es conveniente para solucionar problemas, sobre todo da una visión más extensa del fenómeno físico y que es útil para encontrar más información de la señal [1].

Si tenemos una señal de voz **x[n]**, podemos definir una señal de tiempo-corto $\mathbf{x_m[n]}$ o trama **m** como muestra la ecuación (2.11) [2].

$$\mathbf{x_m[n] = x[n]\, w_m[n]} \tag{2.11}$$

Donde la ventana $\mathbf{w_m[n]}$ es cero para todos sus valores excepto en una región pequeña. Mientras la función de la ventana puede tomar diferentes valores para diferentes tramas **m**, por lo regular se opta que mantenga un valor constante para todas las tramas como se ilustra en la ecuación (2.12).

$$\mathbf{w_m[n] = w[m-n]} \tag{2.12}$$

Donde **w[n]** = 0 para **|n|>m/2**. En la práctica el tamaño de la ventana está en el orden de 20 a 30 ms. Por último la representación de tiempo-corto de Fourier para una trama **m** está definida como muestra la ecuación (2.13) [2].

$$\mathbf{X_m}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \mathbf{x_m[n]}\, e^{-j\omega n} = \sum_{n=-\infty}^{\infty} \mathbf{w[m-n]\, x[n]}\, e^{-j\omega n} \tag{2.13}$$

### 2.4.2 Energía del segmento de voz

Una forma de calcular la energía del segmento de voz en estudio es por medio de la ecuación (2.14).

$$\mathbf{E} = \sum_{\mathbf{i=0}}^{\mathbf{N-1}} |\,\mathbf{s(n)}\,|^{\mathbf{2}} \tag{2.14}$$

Donde **E** es la energía del segmento, **s(n)** es el segmento de voz y **N** es el número de muestras para cada segmento. La estimación de la energía de la señal se utiliza para decidir si el segmento es silencio. Para determinar el umbral para la detección de un segmento de silencio depende del ruido dónde se realice la adquisición de la voz. Cuando un segmento de voz se determina como silencio se puede evitar el proceso de análisis de la señal para determinación de sus parámetros [11].

## 2.5 Pitch

La determinación del *pitch* es una operación común en el procesamiento de señales, ya que proporciona información de la periodicidad del segmento analizado. El Pitch existe cuando se trata de un sonido de voz voceado, con ello se puede replicar al modelo de síntesis como se muestra en la figura 2.9. Desafortunadamente es difícil su estimación por lo que existen muchos algoritmos diferentes para determinarlo. Estos algoritmos se pueden clasificar en tres tipos, dependiendo del dominio donde se realiza el procesamiento: en el tiempo, de la frecuencia o bien cepstral [7].

Para la estimación del período de *pitch* es necesario determinar la naturaleza del segmento bajo análisis. Para tramas de sonido voceado se tendrá que estimar el período de *pitch*, mientras que para tramas con sonido no voceado se omitirá tal cálculo [6].

### 2.5.1 Métodos de determinación de pitch

Existen diversas estrategias para la estimación del *pitch*, como son: Método de recorte central, Método de Gold – Rabiner, Método Homomórfico, Método AMDF (Average Magnitude Difference Function), Algoritmo SIFT (Simplified Inverse Filter Tracking), entre otros [6]. Por su mínima carga computacional debido a que explota las propiedades de la función de autocorrelación y su fácil implementación, se utiliza el Método de Recorte Central [6].

### 2.5.2 Método de Recorte Central

La autocorrelación de tiempo corto nos brinda una representación conveniente para poder determinar el *periodo* o *pitch* de la señal en función del tiempo. Una de las mayores limitaciones de la representación de la autocorrelación es que contiene mucha información de la señal,

incluyendo ruido u otro tipo de información que puede interferir en la estimación del pitch y con ello tener un valor erróneo [1].

Para evitar este tipo de problemas es muy útil procesar la señal de voz, para que la periodicidad más predominante sea analizada, eliminando las demás que pudieran dar problemas en la estimación, con ello se logra obtener un detector de pitch más simple y confiable. La técnica aplicada en este tipo de operación se le conoce como "spectrum flattaners", cuyo objetivo es remover los efectos del tracto vocal de tal manera que iguala las armónicas al mismo nivel de amplitud como es el caso de un tren de impulsos periódico [1].

Existen diversas técnicas de "spectrum flatteners", pero la técnica llamada "*Center Clipping*" o "*Recorte Central*" presenta ventajas para nuestro interés en particular. En el esquema propuesto por Sondhi [1], el recorte central de la señal de voz se obtiene al aplicar una transformación no lineal **C[x(n)]** como se muestra en la figura 2.11.

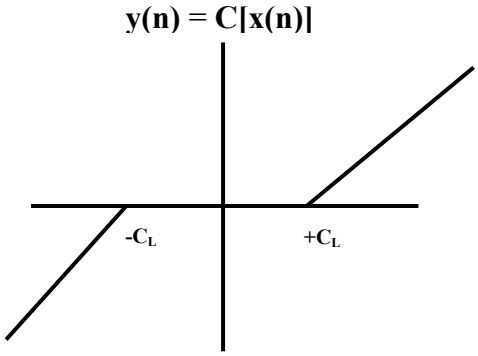



Figura 2.11. Función de recorte central.

El efecto que al aplicar la transformación de recorte central a una ventana de la señal de voz, se muestra en la figura 2.12, dando el resultado mostrado en la figura 2.13. En el recorte central se fijan dos umbrales $\mathbf{C_L}$ tanto para el lado negativo como positivo.

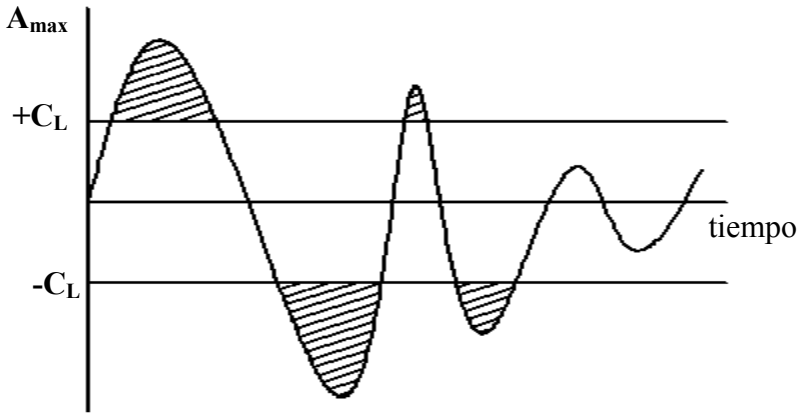



Figura 2.12. Señal original indicando los niveles de recorte.

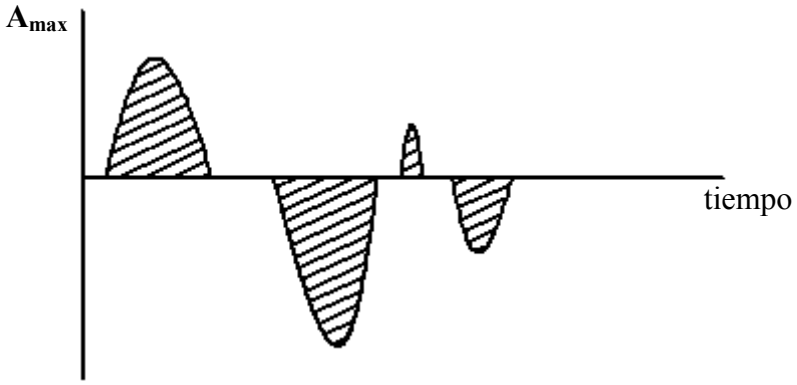



Figura 2.13. Señal resultante después del recorte central.

Al realizar la autocorrelación de la señal, se logra aislar las señales de más baja potencia o ruidosas, que nos resultaría en una autocorrelación más compleja y de mayor dificultad para determinar el *Pitch* de la señal. Pero aun así se puede simplificar aun más, tanto en picos no deseados, como en el cálculo de la autocorrelación. Esto se logra con una simple modificación a la función del recorte central sin degradar la detección del *pitch* de la señal, como se muestra en la figura 2.14.

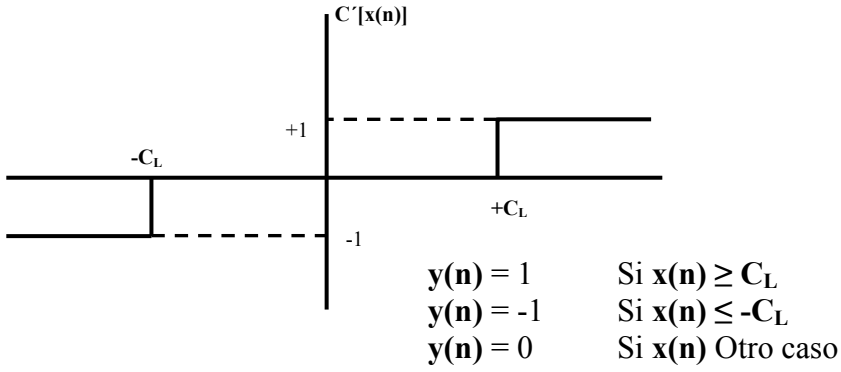



Figura 2.14. Función de recorte central simplificada.

El valor del umbral $\mathbf{C_L}$ es calculado para cada ventana, si dividimos a la ventana en tres segmentos y empleamos la siguiente estrategia:

- Encontrar las amplitudes máximas del primer y tercer subsegmento **($A_1$ y $A_3$)**.
- Calcular el umbral $\mathbf{C_L}$ como muestra la ecuación (2.15).

$$\mathbf{C_L = K\ min(A_1,A_3)} \tag{2.15}$$

Donde el operador **min()** calcula el valor mínimo entre $\mathbf{A_1}$ y $\mathbf{A_3}$ y **K** es un parámetro de calibración que se encuentra entre el valor de 0.6 y 0.8.

Teniendo la señal ya recortada, se aplica la autocorrelación a la señal resultante mostrada en la ecuación (2.16).

$$R(l) = \sum_{n=0}^{N-1} s(n)\, s(n-l) \qquad (2.16)$$

Donde **s(n)** es el segmento de voz recortado por la función de la figura 2.14, **N** es el número de muestras del segmento y "**l**" el índice temporal de retraso. En las figuras 2.15 y 2.16 se ejemplifican la aplicación de este método sobre una señal voceada.

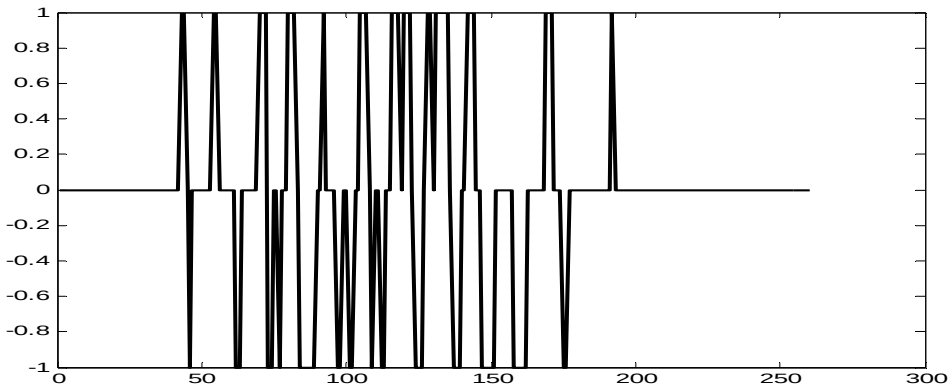



Figura 2.15 Señal después de la transformación del recorte central

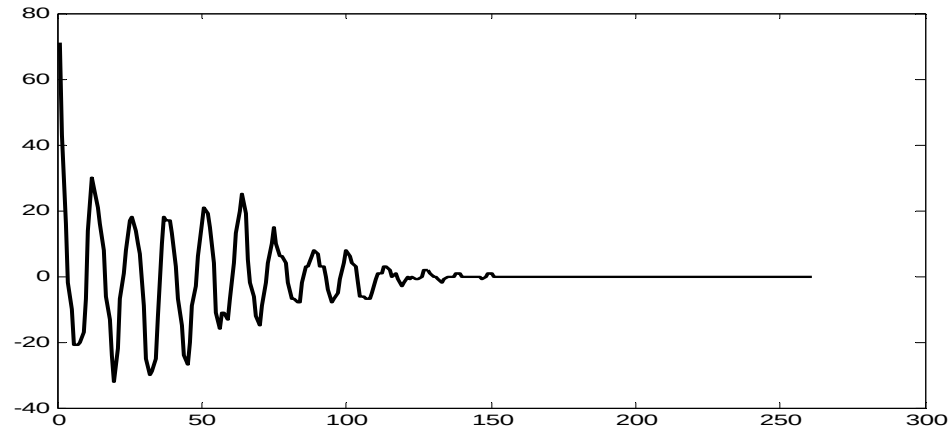



Figura 2.16. Autocorrelación de la señal afectada por el recorte central

En el caso de segmentos de sonidos voceados, como es el caso de la figura 2.15 y 2.16 existe una periodicidad de la forma de onda en la señal, no así para ventanas no voceadas. Por lo tanto, la función de autocorrelación tiene la propiedad de ser periódica si la ventana en estudio también es periódica. Una ventaja de usar la función de autocorrelación es que presenta con mayor realce la propiedad de periodicidad de las señales; en consecuencia, la propiedad de periodicidad puede ser empleada como criterio para la estimación de la naturaleza del segmento, también muestra información de la energía de la señal. Esta información adicional puede propiciar decisiones erróneas. Entonces para la determinación de la naturaleza de la señal se calcula el valor máximo

de la función de autocorrelación comprendida entre las muestras 20 a la 200, y este valor es comparado con un valor umbral igual a **0.3R(0)**. Si el valor máximo es mayor que el valor umbral se decide como un segmento voceado, si es menor, el segmento será no voceado [11].

## 2.6 Resumen

En este capítulo se explicó que para poder realizar el análisis de la voz y obtener sus parámetros, es necesario conocer cómo se genera de forma anatómica y cuáles son sus características físicas; con ello encontrar un modelo digital lo más simplificado posible, pero a su vez con la capacidad de poder reconstruir la señal de voz de forma aceptable. Además es necesario conocer las ventajas y desventajas del modelo para adecuar también la señal y obtener un mejor rendimiento; por ejemplo, pasar la señal a un filtro de preénfasis y acentuar sus componentes de alta frecuencia para calcular los coeficientes LPC, como se mostrará en el siguiente capítulo.

# CAPITULO TRES

# 3. Síntesis de voz

En el procesamiento digital de voz, uno de los objetivos es obtener una representación conveniente de la señal, que sea útil para su manejo en medios de información. La precisión de su representación es requerido de forma particular, por el medio de información donde requiere que la señal se recupere con cierta calidad y magnifique ciertas características o bien se suprima algunas otras. Por ejemplo el objetivo de un cierto sistema es determinar si se trata de una señal de voz o bien es silencio [1].

En este capítulo se aborda la síntesis de voz que consiste en reproducir una señal sintética lo más cercana posible a la señal de voz original, utilizando parámetros extraídos de esta última en una etapa de análisis de los coeficientes LCP, la energía de la señal, determinar si es voceada o no, y el *pitch* para caso en que lo sea.

## 3.1 Producción de voz sintética

El módulo de síntesis de voz, es el componente que genera la forma de onda de la señal de la voz sintética para poderla reproducir. Los sistemas de síntesis de voz pueden ser clasificados en tres tipos dependiendo del modelo usado para la generación de la voz [2].

- **Síntesis por articulación:** Usa modelos físicos para la producción de voz que incluye todas las articulaciones.

- **Síntesis por formantes:** Usa un modelo de fuente-filtro, donde el filtro es caracterizado por un cambio lento en formantes de frecuencia.

- **Síntesis por concatenación:** Genera la señal de voz por concatenación de segmentos previamente almacenados de voz.

**Atributos de la síntesis de voz**

El atributo más importante de la síntesis de voz, es la calidad de su salida dependiendo del tipo de sintetizador y las necesidades del sistema, además de la calidad también se pueden mencionar otras características [2]:

- *Tiempo de respuesta*. Es el tiempo que toma el sintetizador para empezar a hablar, es importante para aplicaciones interactivas que deberá responder en un tiempo menor a 200ms. Este tiempo está compuesto por la ejecución del algoritmo y por el modulo de sintetizador utilizado, así también la computación que involucra.

- *Memoria*. Los sintetizadores basados en reglas requiere de pocos recursos de almacenamiento, por lo que son ampliamente usados donde la memoria es limitada, en comparación a sistemas de concatenación que requiere de grandes cantidades.

- *CPU.* El desempeño del CPU es un factor determinante para operar en tiempo real, aunque en la actualidad los CPUs son suficiente rápidos para muchas aplicaciones.

- *Control del pitch.* Algunos sistemas hablantes requieren que la señal de voz de salida tenga un pitch específico. En el caso que se quiera generar voz para una canción, los sistemas de concatenación que no modifiquen la forma de onda de la señal no pueden realizar esta operación, por lo tanto, para superar esta limitación se graban un gran número de segmentos de voz con diferente pitch.

- *Características de la voz.* Otros sistemas hablantes necesitan una generación de voz específica, como una voz robótica que no puede ser grabada naturalmente, o también voz monótona que es muy complicado grabar. Para esto los sistemas basados en reglas son muy flexibles y son capaces de realizar muchas modificaciones.

## 3.2 Síntesis de voz por formantes

La síntesis por formante, se llama también *síntesis basado en reglas,* que se refiere a las reglas de cómo modificar el pitch, las frecuencias formantes y otros parámetros de un sonido a otro para mantener la continuidad del sistema igual a la generación de voz del sistema humano. Este sistema se puede describir con un diagrama a bloques como muestra la figura 3.1.

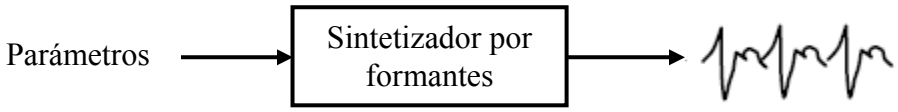



Figura 3.1. Diagrama a bloques de Síntesis por formantes.

En particular se analiza el sistema por formantes utilizando el método de Predicción Lineal (LP), donde parte de los parámetros serán los coeficientes extraídos de la predicción lineal que se estudiará más adelante.

En la figura 3.2 se muestra el aspecto más importante de la voz. La mayor información en la voz está codificada en el espectro de potencia de la onda de presión acústica. Una configuración diferente de la articulación da como resultado una señal con diferente espectro, especialmente con otras frecuencias de resonancia llamadas formantes, que es percibido como un sonido diferente [19].

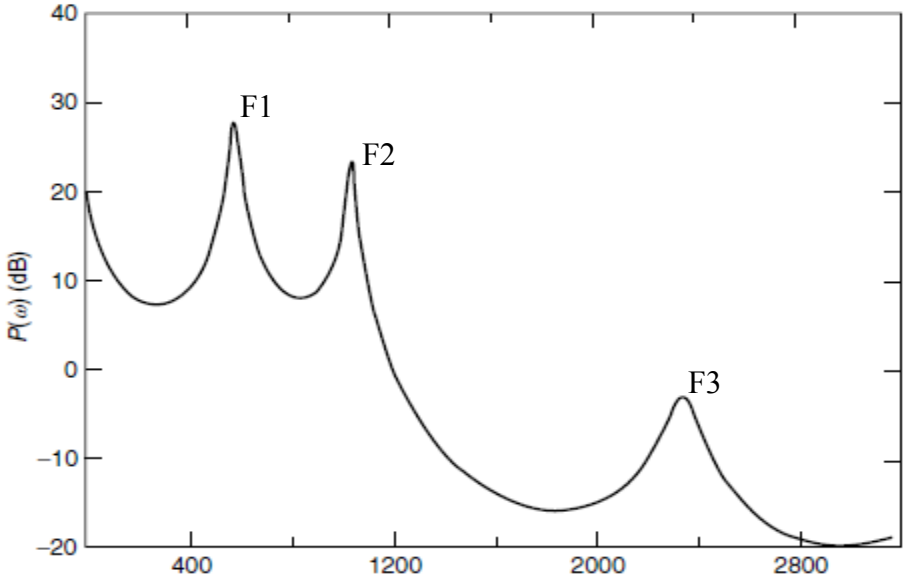



Figura 3.2. Espectro donde se muestran las formantes F1, F2 y F3.

## 3.3 Modelo Todo-Polo y Todo-Cero

Los modelos de Todo-Polo y Todo-Cero son filtros muy utilizados en la síntesis de voz, cuya función de transferencia se muestra en la ecuación (3.1) y en la ecuación (3.2) respectivamente [8].

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{i=1}^{N} h_i z^{-1}} \tag{3.1}$$

$$A(z) = 1 + \sum_{i=1}^{N} h_i z^{-1} \tag{3.2}$$

Dado que el modelo Todo-Cero tiene un comportamiento de un filtro de respuesta finita (FIR) dentro de sus características más importantes se pueden mencionar [11]:

- Son no recursivos.
- Contienen un polo múltiple en el origen de orden **N**.
- Se consideran sólo ceros.
- Siempre son estables.

- Son de memoria finita de longitud **N**, ya que no necesitan las entradas anteriores al tiempo discreto **N-1**.

El modelo Todo-Polo debido a su naturaleza de un filtro de respuesta infinita al impulso (IIR) tiene las siguientes ventajas y desventajas [11]:

- Debido a su recursividad, con pocos coeficientes pueden generar filtros de gran pendiente en la banda de transición.

- Los filtros IIR pueden generar respuestas con un buen grado de aproximación al comportamiento de un filtro analógico, esto es, emulando la respuesta al impulso a la respuesta en frecuencia, ya que cuando la frecuencia de muestreo es muy alta en comparación con las frecuencias de interés el comportamiento de un filtro IIR es similar a un filtro analógico.

- Los filtros IIR pueden ser inestables, por lo que hay que ser cuidadosos en el diseño. Los errores introducidos en el cálculo de la salida **y(n)** debido a la precisión finita de los cálculos matemáticos son difíciles de predecir y como consecuencia pueden hacer inestable al sistema al mover los polos fuera del círculo unitario.

- Como **h(n)** es infinita, no es posible tomar una suma de convolución desde un punto de vista práctico.

- No pueden ser de fase lineal como en los filtros FIR, pero esto se compensa al tener una mejor respuesta de la magnitud en frecuencia.

### *3.3.1 Forma Directa*

La realización de los filtros de forma directa en el dominio del tiempo corresponde al filtro Todo-Polo representado por la ecuación en diferencias (3.3) y para el filtro Todo-Cero por la ecuación (3.4). Así también en la figura 3.3 podemos observar diagrama de flujo de la señal. Cabe mencionar que la respuesta a impulso de un filtro Todo-Polo tiene un número infinito de muestras con valores diferentes de cero, dado que la salida **y[n]** está formada por la suma de una muestra de entrada **x[n]** y una versión de **N** retrasos de la salida que están afectadas en su escala por los $\mathbf{a_i}$. Esto es un comportamiento de un filtro de *respuesta infinito al impulso* (IIR). Por otro lado para el filtro Todo-Cero la respuesta al impulso sólo tiene **N+1** muestras diferentes de cero (el resto son ceros) y es un comportamiento de un filtro de *respuesta finita al impulso* (FIR).

$$\mathbf{y[n] = x[n] - \sum_{i=1}^{N} a_i\, y[n-i]} \tag{3.3}$$

$$\mathbf{y[n] = x[n] + \sum_{i=1}^{N} b_i\, x[n-i]} \tag{3.4}$$

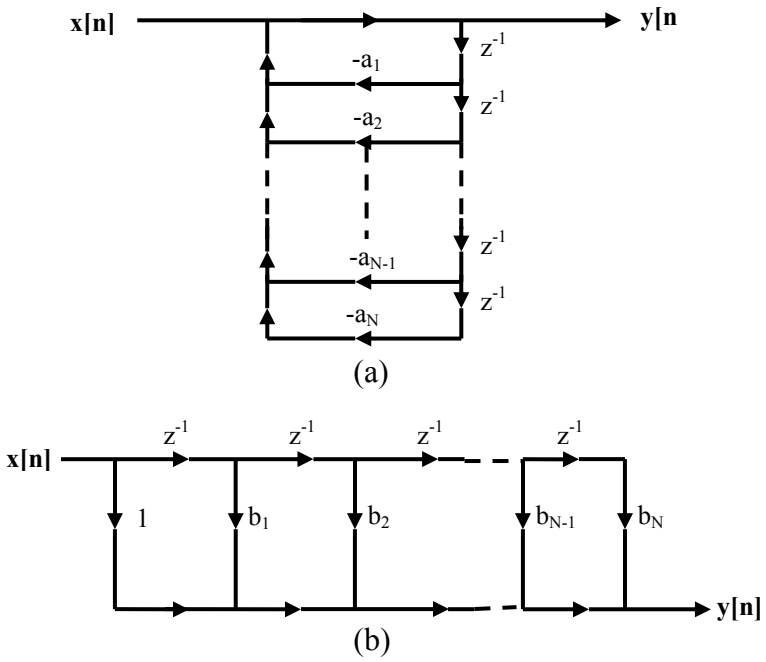



Figura 3.3. Graficas de flujo de señal de forma directa. (a) Todo-Polo y (b) Todo-Cero.

## 3.4 Predicción Lineal

La Predicción Lineal (LP) forma una parte integral de la mayoría de los algoritmos modernos de procesamiento de la voz, la idea fundamental es que una muestra de voz se puede calcular de forma aproximada como una combinación lineal de muestras pasadas. La mayor carga de cómputo de la combinación lineal se encuentra en la minimización del error cuadrático, medio de predicción dentro de una trama de la señal donde los *pesos* resultantes o coeficientes de predicción lineal, son usados para representar de forma particular una trama.

La predicción lineal también puede interpretarse como un procedimiento de estimación de espectro, en el proceso de análisis para Predicción Lineal nos permite calcular los parámetros del modelo *Auto-Regresivo* (AR) que definen la densidad espectral de la señal en estudio [8].

El proceso de producción de voz humana revela que la generación de cada fonema es caracterizado básicamente por dos factores: la fuente de excitación y la forma del tracto vocal. Para entender las características de producción de voz se asume que tanto la fuente como modelo del tracto vocal son independientes. El modelo del tracto vocal **H(z)**, que se puede observar en la figura 2.6, es excitado por una señal similar a una excitación glotal **u(n)** para producir la señal final de voz **s(n)**. Todas estas características se pueden representar por un filtro digital variante en lapsos de tiempo y de estado constante en sus parámetros, en una ventana de tiempo donde su función de transferencia está definida por la ecuación (3.5).

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{3.5}$$

Si se obtiene la transformada *Z* inversa de la ecuación (3.5) tendremos la ecuación en diferencias mostrada en la ecuación (3.6)

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{3.6}$$

La ecuación (3.6) es conocida como ecuación en diferencias de la Codificación por Predicción Lineal (LPC) [9].

### *3.4.1 El problema de Predicción Lineal*

La predicción lineal, puede ser descrita como un problema de un sistema de identificación, donde los parámetros del modelo Auto-Regresivo son estimados desde la misma señal [8]. La Codificación por Predicción Lineal toma su nombre del hecho de que predice la muestra actual, como una combinación lineal *p* muestras pasadas [2] como muestra la ecuación (3.7).

$$\hat{s}(n) = -\sum_{k=1}^{p} a_k s(n-k) \tag{3.7}$$

Donde **ŝ(n)** es la señal estimada, **p** es el orden de predicción, $\mathbf{a_k}$ son los coeficientes de predicción y por ultimo **s(n-k)** se trata de muestras pasadas con retraso **k**.

La secuencia **ŝ(n)**, es el predictor de **s(n)**, resultado de la suma de las **p** muestras pasadas de **s(n)**, donde cada muestra tiene un peso dado. La función asociada al sistema con orden **p** de predicción, es un filtro finito al impulso (FIR) con una longitud **p** dado por la ecuación (3.8) [10].

$$P(z) = \sum_{k=1}^{p} a_k z^{-k} \tag{3.8}$$

El error de predicción **e(n)**, está dado por la diferencia de la secuencia **s(n)** y de su predicción **ŝ(n)** tal como muestra la ecuación (3.9) y (3.10). El filtro se muestra en la figura 3.4.

$$e(n) = s(n) - \hat{s}(n) \tag{3.9}$$

$$e(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k) \tag{3.10}$$

Donde $\mathbf{a_0 = 1}$

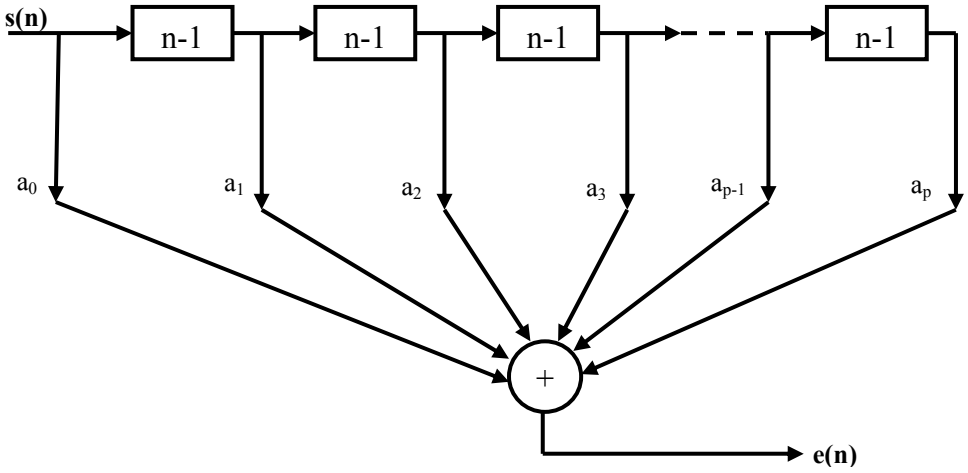



Figura 3.4 Filtro de Predicción.

La secuencia de error de predicción puede ser vista como la salida de un sistema, la cual tiene la función de transferencia mostrada en la ecuación (3.11), y sustituyendo **P(z)** definida en la ecuación (3.8) da como resultado la ecuación (3.12).

$$\mathbf{A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{3.11}$$

$$\mathbf{A(z) = 1 - P(z)} \tag{3.12}$$

El modelo de predicción lineal puede descomponerse en dos partes, análisis y síntesis como muestra la figura 3.5.

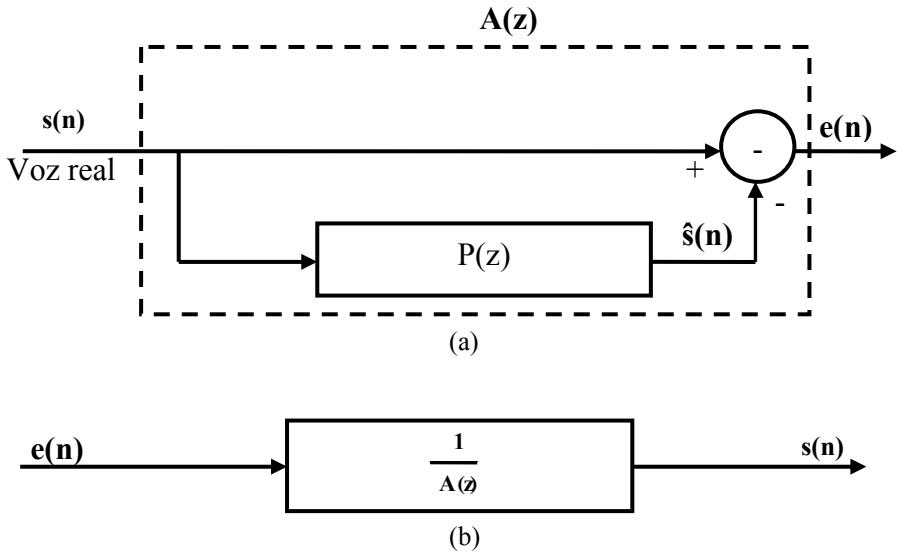



Figura 3.5. Vista del filtro de Predicción Lineal. (a) Análisis, filtro de error de predicción. (b) Síntesis, reconstrucción de **s(n)**.

## *3.4.2 Minimización de Error*

A partir del la ecuación (3.10) es necesario obtener el mínimo error **e(n)**, para que el modelo sea óptimo, para ello se cuenta con varios métodos numéricos de optimización de error, entre los cuales se encuentran:

- Mínimo error por corrección de tendencia.
- Mínima entropía de error estimado (MEEE).
- Mínimo error por mínimos cuadrados (MMSE).

El más comúnmente utilizado es la minimización de error de mínimos cuadrados (MMSE) el cual se desarrolla a continuación [11].

Si se aplica la esperanza al error cuadrático a la ecuación (3.10) tendremos:

$$\mathbf{E}\left\{\mathbf{e^2(n)}\right\} = \mathbf{E}\left\{\left(\mathbf{s(n)} + \sum_{k=1}^{p} \mathbf{a_k s(n-k)}\right)\left(\mathbf{s(n)} + \sum_{k=1}^{p} \mathbf{a_k s(n-k)}\right)\right\} \tag{3.13}$$

Desarrollando el binomio dado en la ecuación (3.13):

$$\mathbf{E}\left\{\mathbf{e^2(n)}\right\} = \mathbf{E}\left\{\mathbf{s(n)\,s(n)}\right\} + 2\,\mathbf{E}\left\{\sum_{k=1}^{p} \mathbf{a_k s(n-k)\,s(n)}\right\} + \mathbf{E}\left\{\left(\sum_{k=1}^{p} \mathbf{a_k s(n-k)}\right)\left(\sum_{j=1}^{p} \mathbf{a_j s(n-j)}\right)\right\} \tag{3.14}$$

Si representamos la ecuación (3.14) en forma de matrices:

$$E\{e^2(n)\} = r_p(0) + 2A_p^T r_p + A_p^T R_p A_p \tag{3.15}$$

Aplicando el criterio de optimización a la ecuación (3.15) resulta:

$$\frac{\partial E\{e^2(n)\}}{\partial A_p} = 2r_p + 2R_p A_p = 0 \tag{3.16}$$

$$R_p A_p = -r_p \tag{3.17}$$

La ecuación (3.17) en forma matricial [11]:

$$\begin{pmatrix} R_m[0] & R_m[1] & R_m[2] & \dots & R_m[p-1] \\ R_m[1] & R_m[0] & R_m[1] & \dots & R_m[p-2] \\ R_m[2] & R_m[1] & R_m[0] & \dots & R_m[p-3] \\ \dots & \dots & \dots & \dots & \dots \\ R_m[p-1] & R_m[p-2] & R_m[p-3] & \dots & R_m[0] \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_p \end{pmatrix} = \begin{pmatrix} R_m[1] \\ R_m[2] \\ R_m[3] \\ \dots \\ R_m[p] \end{pmatrix} \tag{3.18}$$

Donde $R_p$ se conoce como matriz de autocorrelación.

**Propiedades de la matriz de autocorrelación $R_p$**

- La matriz de autocorrelación $R_p$ es simétrica o hermitiana en el caso complejo, es decir, que $R_p = R_p^T$, entonces sus vectores característicos son ortogonales, si $Q$ es una matriz de vectores característicos de la matriz $R_p$, entonces $Q_m^T Q_n = 0$ [11].

- Si la matriz $R_p$ es real y simétrica, todos sus valores característicos deben ser reales o iguales a cero.

- La matriz $Q$ de vectores característicos puede ser normalizada, de tal forma que $QQ^T = I$.

- Para procesos estacionarios la matriz $R_p$ es del tipo Toeplitz, esto es, que todos los elementos de cada diagonal son iguales.

- La matriz $R_p$ es positivamente definida, implica que cumple con $X^T R_p X > 0$ para cualquier vector $X$.

- Cuando los elementos de un vector del proceso estacionario en observación son arreglados en forma “backward”, el efecto es equivalente a la transposición de la matriz de autocorrelación.

### 3.4.3 Ecuación normal y principio de ortogonalidad

La ecuación (3.17) es conocida como *ecuación normal* y consiste de un sistema de "**p**" ecuaciones simultáneas, cuyas incógnitas son los coeficientes $\mathbf{a_p}$ del modelo.

La ecuación (3.19) es conocida como el principio de ortogonalidad, y es un resultado directo de la minimización del error medio cuadrático. La condición de ortogonalidad establece que el producto escalar del error con los datos de la predicción es cero, es decir, que la secuencia del error no está correlacionada con los datos. Como se observa en la figura 3.6, una interpretación geométrica permite observar que la señal estimada **ŝ(n),** se obtiene al proyectar la señal deseada **s(n)**, en el espacio de los datos de **s(n-i)** como una combinación lineal de éstos con los coeficientes $\mathbf{a_k}$ [11].

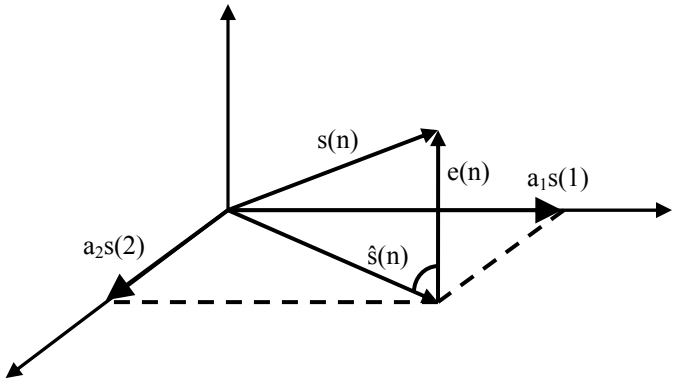



Figura 3.6. Condición de ortogonalidad.

### 3.4.4 Método de Autocorrelación

Este método ya fue introducido en la sección de minimización del error y nos permite obtener de forma eficiente y estable la solución de la ecuación normal, es decir, como la matriz de autocorrelación $\mathbf{R_p}$, es no singular, siempre es invertible por lo que todos los coeficientes $\mathbf{a_p}$ existen. El método de autocorrelación asume que las muestras fuera del intervalo [n – p, n + p] todas son cero y extiende el intervalo de predicción del error [10]. A partir de la ecuación normal (3.17) se puede despejar el vector de coeficientes $\mathbf{a_p}$ y con ello poder llegar a la solución del sistema, tal como muestra la ecuación (3.19).

$$\mathbf{a_p = - R_p^{-1}\ r_p} \tag{3.19}$$

Por las propiedades de $\mathbf{R_p}$ y especialmente por ser una matriz Toeplitz permite ecuaciones lineales que pueden ser resueltas por el algoritmo de Levison-Durbin, que será descrito más adelante, para la solución y garantizar la estacionalidad de la señal es necesario dividir en tramas la muestra entera, aplicando el ventaneo que se describió en la sección 2.3.3 para intervalos de 20 a 40 ms.

### 3.4.5 Algoritmo de Levison-Durbin

La ecuación normal dada en (3.17) puede ser calculada encontrando la matriz inversa de $\mathbf{R_p}$, donde la solución está dada por la ecuación (3.19). En general la demanda computacional para encontrar la inversa es considerable, ya que se necesitan alrededor de $\mathbf{O(p^3)}$ operaciones. Afortunadamente existen algoritmos para la solución de la ecuación normal, que toman ventaja de la estructura especial de la matriz de correlación. Uno de ellos es el algoritmo de Levison-Durbin, que es muy apropiado para la implementación en términos prácticos en el análisis de la Predicción Lineal y que necesita $\mathbf{O(p^2)}$ operaciones, ya que explota la ventaja de tener una matriz Toeplitz. Consideremos la ecuación normal aumentada de la forma mostrada en la ecuación (3.20) [8],[11],[14].

$$\begin{pmatrix} \mathbf{R[0]} & \mathbf{R[1]} & \mathbf{...} & \mathbf{R[M]} \\ \mathbf{R[1]} & \mathbf{R[0]} & \mathbf{...} & \mathbf{R[M-1]} \\ \mathbf{...} & \mathbf{...} & \mathbf{...} & \mathbf{...} \\ \mathbf{R[M]} & \mathbf{R[M-1]} & \mathbf{...} & \mathbf{R[0]} \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \mathbf{a_1} \\ \mathbf{...} \\ \mathbf{a_M} \end{pmatrix} = \begin{pmatrix} \mathbf{J} \\ \mathbf{0} \\ \mathbf{...} \\ \mathbf{0} \end{pmatrix} \tag{3.20}$$

Con el objetivo de comenzar con la solución de los coeficientes LPCs, $\mathbf{a_i}$ , $\mathbf{i = 1,\ldots, M}$, dados los valores de la autocorrelación $\mathbf{R_p[l], l = 0, 1, \ldots, M}$. **J** representa el mínimo del error medio cuadrático de predicción. En una situación práctica, los valores de la autocorrelación son estimados desde las muestras de la señal y **J** es usualmente desconocida; sin embargo el algoritmo de Levison-Durvin se fórmula para encontrar esta cantidad.

El algoritmo de Levinson-Durvin se enfoca a encontrar la solución del predictor de orden **M** desde el predictor de orden **(M-1)**. Es un proceso recursivo e iterativo, donde la solución del predictor de orden cero es el primero que se encuentra, el cual es usado para encontrar la solución del predictor de orden uno; este proceso se repite hasta que se encuentra el predictor de orden **M**. El algoritmo depende de dos propiedades claves de la matriz de autocorrelación [8]:

- La matriz de autocorrelación de un tamaño dado contiene sub-bloques de matrices de correlación de orden inferior.

- La matriz de autocorrelación no sufre cambio alguno cuando se transpone, esta propiedad es una consecuencia directa de ser una matriz Toeplitz. Se dice que una matriz es Toeplitz si todos los elementos que contiene en su diagonal principal son iguales y si los elementos de cualquier otra diagonal paralela a la diagonal principal son también iguales.

$$\begin{pmatrix} \mathbf{R[0]} & \mathbf{R[1]} & \mathbf{...} & \mathbf{R[M]} \\ \mathbf{R[1]} & \mathbf{R[0]} & \mathbf{...} & \mathbf{R[M-1]} \\ \mathbf{...} & \mathbf{...} & \mathbf{...} & \mathbf{...} \\ \mathbf{R[M]} & \mathbf{R[M-1]} & \mathbf{...} & \mathbf{R[0]} \end{pmatrix} \begin{pmatrix} \mathbf{a_0} \\ \mathbf{a_1} \\ \mathbf{...} \\ \mathbf{a_M} \end{pmatrix} = \begin{pmatrix} \mathbf{b_0} \\ \mathbf{b_1} \\ \mathbf{...} \\ \mathbf{b_M} \end{pmatrix} \tag{3.21}$$

$$\begin{pmatrix} R[0] & R[1] & \dots & R[M] \\ R[1] & R[0] & \dots & R[M-1] \\ \dots & \dots & \dots & \dots \\ R[M] & R[M-1] & \dots & R[0] \end{pmatrix} \begin{pmatrix} a_M \\ a_{M-1} \\ \dots \\ a_0 \end{pmatrix} = \begin{pmatrix} b_M \\ b_{M-1} \\ \dots \\ b_0 \end{pmatrix} \tag{3.22}$$

La solución de la ecuación normal aumentada comienza desde el predictor de orden cero. Se muestra a continuación como se obtiene la solución de un predictor de cierto orden a partir de un predictor de orden menor.

**Predictor de orden cero**

En este caso se considera la ecuación (3.23) que está actualmente resuelta. El estado de esta relación es básicamente el mínimo error medio cuadrático que está contenido en el predictor de orden cero y está dado por la autocorrelación de la señal con un retrasó cero, o la diferencia de la señal en sí misma. Para el orden de predicción cero el error de predicción es igual a la señal en sí misma [8].

$$R[0] = J_0 \tag{3.23}$$

Expandiendo la ecuación (3.23) a la siguiente dimensión se tiene la ecuación (3.24) la cual es de dos dimensiones con $a_1 = 0$. Mientras $a_1 = 0$, en general la condición optima no puede lograrse por tanto se introduce el término $\Delta_0$ para balancear la ecuación. Esta cantidad corresponde al valor de **R[1]** como muestra la ecuación (3.25)

$$\begin{pmatrix} R[0] & R[1] \\ R[1] & R[0] \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} J_0 \\ \Delta_0 \end{pmatrix} \tag{3.24}$$

$$\Delta_0 = R[1] \tag{3.25}$$

Por la propiedad de la matriz de correlación Toeplitz la ecuación (3.24) es equivalente a la ecuación (3.26) que será usada en el siguiente paso.

$$\begin{pmatrix} R[0] & R[1] \\ R[1] & R[0] \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \Delta_0 \\ J_0 \end{pmatrix} \tag{3.26}$$

**Predictor de orden uno**

Entonces se tiene que encontrar la solución de la ecuación (3.27) [8].

$$\begin{pmatrix} R[0] & R[1] \\ R[1] & R[0] \end{pmatrix} \begin{pmatrix} 1 \\ a_1^{(1)} \end{pmatrix} = \begin{pmatrix} J_1 \\ 0 \end{pmatrix} \tag{3.27}$$

Donde $a_1^{(1)}$ es el primer coeficiente de predicción lineal del predictor; el superíndice denota el orden de predicción de uno. $J_1$ representa el mínimo error medio cuadrático de predicción logrado usando el predictor de primer orden. Como consecuencia tendremos dos incógnitas que

son $a_1^{(1)}$ y $J_1$. Consideremos una solución de la forma mostrada en la ecuación (3.28).

$$\begin{pmatrix} 1 \\ a_1^{(1)} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} - k_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3.28}$$

Con $k_1$ que será una constante. Multiplicando ambos lados por la matriz de autocorrelación se obtiene la ecuación (3.29).

$$\begin{pmatrix} R[0] & R[1] \\ R[1] & R[0] \end{pmatrix} \begin{pmatrix} 1 \\ a_1^{(1)} \end{pmatrix} = \begin{pmatrix} R[0] & R[1] \\ R[1] & R[0] \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} - k_1 \begin{pmatrix} R[0] & R[1] \\ R[1] & R[0] \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3.29}$$

Sustituyendo (3.24), (3.26) y (3.27) nos da la ecuación (3.30).

$$\begin{pmatrix} J_1 \\ 0 \end{pmatrix} = \begin{pmatrix} J_0 \\ \Delta_0 \end{pmatrix} - k_1 \begin{pmatrix} \Delta_0 \\ J_0 \end{pmatrix} \tag{3.30}$$

Usando la ecuación (3.25) tendremos la ecuación (3.31).

$$k_1 = \frac{\Delta_0}{J_0} = \frac{R[1]}{J_0} \tag{3.31}$$

El coeficiente de predicción lineal de este predictor es fácilmente encontrado de la ecuación (3.28) dándonos la ecuación (3.32).

$$a_1^{(1)} = -k_1 \tag{3.32}$$

Usando la ecuación (3.30) y (3.31) se encuentra la ecuación (3.33).

$$J_1 = J_0 \left(1 - k_1^2\right) \tag{3.33}$$

De esta manera el predictor de primer orden está completamente definido. El parámetro $k_1$ es conocido como el coeficiente de reflexión (RC), representando una alternativa a los coeficientes de predicción lineal. Hay que notar que $k_1$ (y por lo tanto $a_1^{(1)}$ y $J_1$) son derivados de resultados previos dados por las ecuaciones (3.31), (3.32) y (3.33).

De manera similar, el siguiente paso del predictor de orden uno se puede expandir a tres dimensiones como muestra la ecuación (3.34) ó (3.35).

$$\begin{pmatrix} R[0] & R[1] & R[2] \\ R[1] & R[0] & R[1] \\ R[2] & R[1] & R[0] \end{pmatrix} \begin{pmatrix} 1 \\ a_1^{(1)} \\ 0 \end{pmatrix} = \begin{pmatrix} J_1 \\ 0 \\ \Delta_1 \end{pmatrix} \tag{3.34}$$

$$\begin{pmatrix} R[0] & R[1] & R[2] \\ R[1] & R[0] & R[1] \\ R[2] & R[1] & R[0] \end{pmatrix} \begin{pmatrix} 0 \\ a_1^{(1)} \\ 1 \end{pmatrix} = \begin{pmatrix} \Delta_1 \\ 0 \\ J_1 \end{pmatrix} \tag{3.35}$$

Donde $\Delta_1$ representa el término adicional necesario para balancear la ecuación cuando el predictor de primer orden es usado y $R[2] \neq 0$. Esta cantidad es resuelta como muestra la ecuación (3.36).

$$\Delta_1 = R[2] + a_1^{(1)} R[1] \tag{3.36}$$

**Predictor de orden dos**

Resolviendo un paso más adelante tenemos la ecuación (3.37) [8].

$$\begin{pmatrix} R[0] & R[1] & R[2] \\ R[1] & R[0] & R[1] \\ R[2] & R[1] & R[0] \end{pmatrix} \begin{pmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \end{pmatrix} = \begin{pmatrix} J_2 \\ 0 \\ 0 \end{pmatrix} \tag{3.37}$$

En este caso las incógnitas son los coeficientes LPCs $a_1^{(2)}$, $a_2^{(2)}$ y el mínimo error medio cuadrático de predicción $J_2$. Considerando la solución de la forma como muestra la ecuación (3.38).

$$\begin{pmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \end{pmatrix} = \begin{pmatrix} 1 \\ a_1^{(1)} \\ 0 \end{pmatrix} - k_2 \begin{pmatrix} 0 \\ a_1^{(1)} \\ 1 \end{pmatrix} \tag{3.38}$$

Con $k_2$ como RC. Multiplicando ambos lados por la matriz de correlación y usando las ecuaciones (3.34), (3.35) y (3.37) nos queda como la ecuación (3.39)

$$\begin{pmatrix} J_2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} J_1 \\ 0 \\ \Delta_1 \end{pmatrix} - k_2 \begin{pmatrix} \Delta_1 \\ 0 \\ J_1 \end{pmatrix} \tag{3.39}$$

La constante RC $k_2$ se puede encontrar de la ecuación (3.39) y usando (3.36) para $\Delta_1$ resultando la ecuación (3.40).

$$k_2 = \frac{1}{J_1} \left( R[2] + a_1^{(1)} R[1] \right) \tag{3.40}$$

De la ecuación (3.38) se pueden deducir las ecuaciones (3.41) y (3.42).

$$a_2^{(2)} = -k_2 \tag{3.41}$$

$$a_1^{(2)} = a_1^{(1)} - k_2 a_1^{(1)} \tag{3.42}$$

Finalmente $\mathbf{J_2}$ es encontrado a partir de la ecuación (3.39) y (3.40) como muestra la ecuación (3.43).

$$\mathbf{J_2 = J_1 \left(1 - k_2^2\right)} \tag{3.43}$$

Para el siguiente paso el predictor de orden dos se expande a cuatro dimensiones como muestra la ecuación (3.44) y (3.45).

$$\begin{pmatrix} \mathbf{R[0]} & \mathbf{R[1]} & \mathbf{R[2]} & \mathbf{R[3]} \\ \mathbf{R[1]} & \mathbf{R[0]} & \mathbf{R[1]} & \mathbf{R[2]} \\ \mathbf{R[2]} & \mathbf{R[1]} & \mathbf{R[0]} & \mathbf{R[1]} \\ \mathbf{R[3]} & \mathbf{R[2]} & \mathbf{R[1]} & \mathbf{R[0]} \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \mathbf{a_1^{(2)}} \\ \mathbf{a_2^{(2)}} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{J_2} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{\Delta_2} \end{pmatrix} \tag{3.44}$$

$$\begin{pmatrix} \mathbf{R[0]} & \mathbf{R[1]} & \mathbf{R[2]} & \mathbf{R[3]} \\ \mathbf{R[1]} & \mathbf{R[0]} & \mathbf{R[1]} & \mathbf{R[2]} \\ \mathbf{R[2]} & \mathbf{R[1]} & \mathbf{R[0]} & \mathbf{R[1]} \\ \mathbf{R[3]} & \mathbf{R[2]} & \mathbf{R[1]} & \mathbf{R[0]} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{a_2^{(2)}} \\ \mathbf{a_1^{(2)}} \\ \mathbf{1} \end{pmatrix} = \begin{pmatrix} \mathbf{\Delta_2} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{J_2} \end{pmatrix} \tag{3.45}$$

Con lo que $\mathbf{\Delta_2}$ se muestra en la ecuación (3.46).

$$\mathbf{\Delta_2 = R[3] + a_1^{(2)} R[2] + a_2^{(2)} R[1]} \tag{3.46}$$

**Predictor de orden tres**

En este caso, la solución considerada está dada por la ecuación (3.47) [8].

$$\begin{pmatrix} \mathbf{1} \\ \mathbf{a_1^{(3)}} \\ \mathbf{a_2^{(3)}} \\ \mathbf{a_3^{(3)}} \end{pmatrix} = \begin{pmatrix} \mathbf{1} \\ \mathbf{a_1^{(2)}} \\ \mathbf{a_2^{(2)}} \\ \mathbf{0} \end{pmatrix} - \mathbf{k_3} \begin{pmatrix} \mathbf{0} \\ \mathbf{a_2^{(2)}} \\ \mathbf{a_1^{(2)}} \\ \mathbf{1} \end{pmatrix} \tag{3.47}$$

Procediendo de manera similar se pueden llegar a la solución mostrada por las ecuaciones (3.48), (3.49), (3.50), (3.51) y (3.52).

$$\mathbf{k_3 = \frac{1}{J_2} \left(R[3] + a_1^{(2)} R[2] + a_2^{(2)} R[1]\right)} \tag{3.48}$$

$$\mathbf{a_3^{(3)} = -k_3} \tag{3.49}$$

$$\mathbf{a_2^{(3)} = a_2^{(2)} - k_3 a_1^{(2)}} \tag{3.50}$$

$$\mathbf{a_1^{(3)} = a_1^{(2)} - k_3 a_2^{(2)}} \tag{3.51}$$

$$J_3 = J_2 \left(1 - k_3^2\right) \tag{3.52}$$

El procedimiento continúa hasta que el orden "**p**" de predicción necesario es alcanzado.

Entonces el algoritmo de Levinson-Durbin se resume como sigue [8]:

---

**Iniciación: l=0**, fijar $J_0 = R[0]$.

---

**Recursivo:** para **l = 1, 2, …, p** (orden de predicción **l**)

- **Paso 1**. Calcular los coeficientes RC de orden **l** como muestra la ecuación (3.53).

$$k_l = \frac{1}{J_{l-1}} \left( R[l] + \sum_{i=1}^{l-1} a_i^{(l-1)} R[l-i] \right) \tag{3.53}$$

- **Paso 2**. Calcular los coeficientes LPCs del predictor de orden **l**

$$a_l^{(l)} = -k_l \tag{3.54}$$

$$a_i^{(l)} = a_i^{(l-1)} - k_l a_{(l-i)}^{(l-1)} \quad ; i = 1,2,\ldots,l-1 \tag{3.55}$$

  Se detiene si **l=M**.

- **Paso 3**. Calcular el mínimo error medio cuadrático asociado con la solución de orden **l**.

$$J_l = J_{l-1} \left(1 - k_l^2\right) \tag{3.56}$$

  Se establece $l \leftarrow l + 1$ y regresa al paso 1.

---

**Finalización:** Los coeficientes LPC finales son

$$a_i = a_i^{(p)} \quad ; i = 1,2,\ldots,p$$

---

Cabe mencionar que el proceso calcula los coeficientes LPCs y también los coeficientes RCs ($k_i$, **i=1, 2, …,p**) son encontrados.

## 3.5 Codificador decodificador de voz LPC (Vocoder)

El LPC vocoder consta de la unión de dos bloques, que es el codificador de LPC y decodificador de LPC o también conocido como *Encoder* y *Decoder* respectivamente. En la figura 3.6 se

muestra el *Encoder* o análisis de la señal para obtener los parámetros de predicción lineal; mientras que en la figura 3.7 se muestra el *Decoder* o síntesis de voz utilizando los parámetros de predicción lineal.

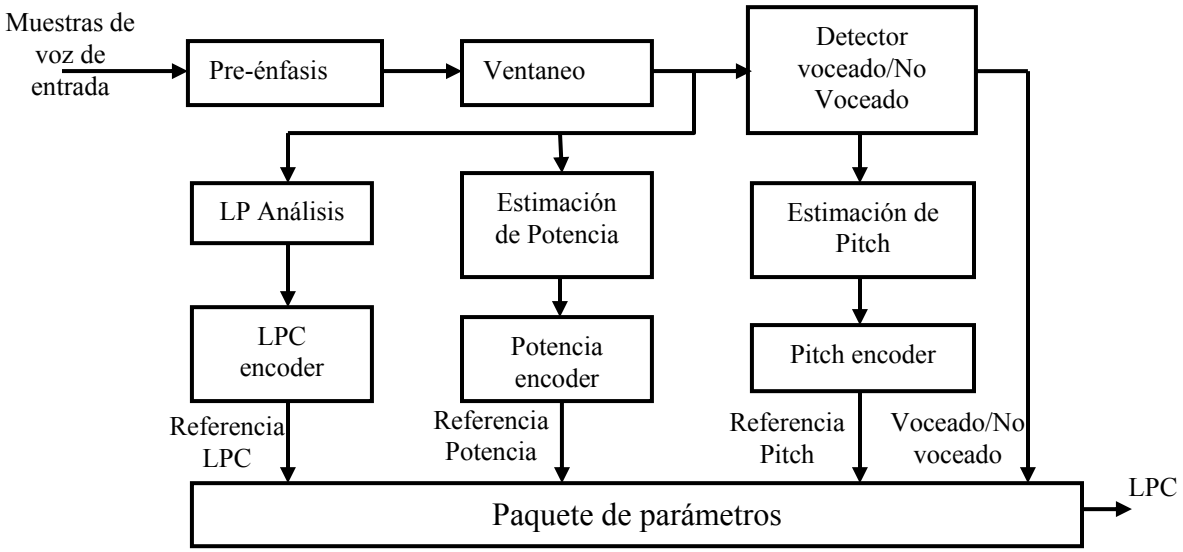



Figura 3.6. Diagrama de bloques del LPC encoder.

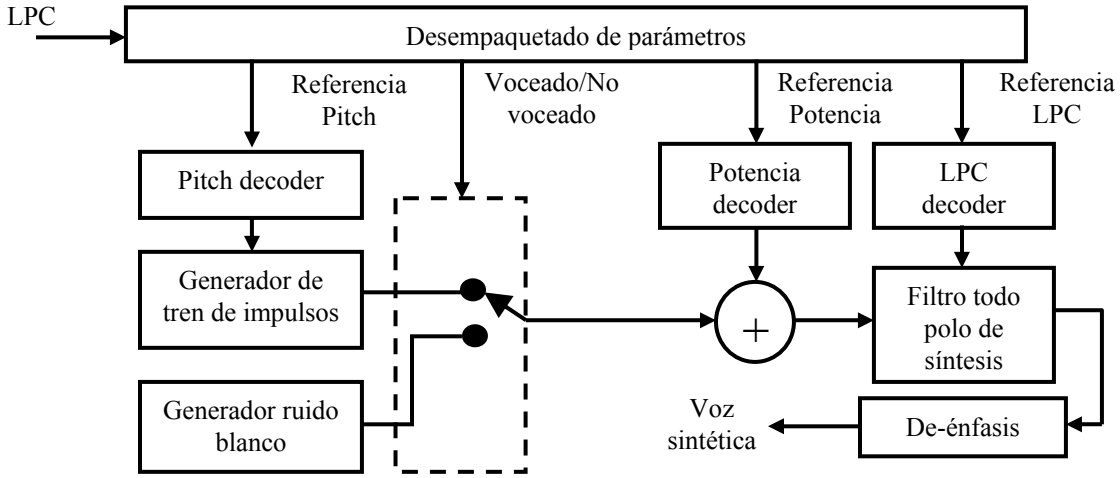



Figura 3.7. Diagrama de bloques del LPC decoder

Para el proceso de síntesis de voz es necesario tener una etapa anterior, que es el análisis de voz, donde se obtienen los parámetros necesarios para su posterior reproducción, a continuación enumeraremos los principales puntos de cada uno de ellos basándonos en las figuras 3.6 y 3.7.

**Análisis de la señal de voz**

El principal objetivo es extraer de la señal de voz original, parámetros necesarios para el decoder del LPC. Por lo que es necesario realizar los siguientes pasos.

- Se determina que la señal no sea silencio.
- La señal de voz se pasa por un filtro de pre-énfasis.
- La señal de voz se divide en tramas y se ventanea utilizando Hamming.
- Se calcula el Pitch de la señal en caso de ser voceada, de lo contrario el pitch es cero.
- Se determina la Energía de la trama de voz
- Se calcula sus coeficientes de predicción lineal (LPCs).
- Se hace un arreglo de estos parámetros para su almacenamiento (Encoder).

**Síntesis de voz**

La principal función de la síntesis, es reproducir la señal lo más parecido posible a la señal de voz original, utilizando los parámetros previamente calculados realizando los siguientes pasos:

- Se determina si el segmento es de silencio o no.
- A partir del arreglo de parámetros previamente calculados, si el segmento de voz es voceado, se extrae el Pitch generando un tren de pulsos con el periodo indicado, de lo contrario se genera ruido blanco.
- Se recupera la potencia del segmento de voz, con él se escala la energía de salida.
- Se recuperan los coeficientes LPCs definiendo el comportamiento del filtro todo polo de síntesis.
- Por último se pasa un filtro de de-énfasis, dando como resultado la voz sintética.

## 3.6 Resumen

En este capítulo se mostró como obtener los parámetros necesarios para la producción de voz sintética, la calidad de la señal original es importante para contar con parámetros confiables, de igual manera, la cantidad de coeficientes LPCs que se ocuparán y una buena determinación del pitch determinan inteligibilidad del la señal resultante. Para trabajar en tiempo real es necesario seleccionar con cuidado las técnicas y algoritmos a utilizar ya que con ello se definirá el tiempo de respuesta del sistema.

# CAPITULO CUATRO

# 4. Comunicación Telefónica

Las telecomunicaciones han tenido una gran importancia desde sus inicios, y la telefonía a sufrido avances las cuales permiten en la vida moderna implementar diversos sistemas de interconexión; como son, el internet y el envío de fax por mencionar algunas. A través del tiempo, las líneas telefónicas han conservado características como son el ancho de banda capaz de transmitir la voz humana.

La señal de voz no es lo único que se transmite por la línea telefónica, también se transmiten el tono de marcado, tonos de las teclas para el marcado, tono de ocupado, y tono de "ring". Estas señales son de control para la conexión o indicaciones del estatus de la llamada que pueden ser la señal de tonos (analógico) o señales "*on-off*" (digitales).

En este capítulo se estudian los principios y configuración del teléfono, mostrando sus principales procesos, características y de igual manera limitaciones que interviene de forma directa en el desarrollo del presente trabajo, puesto que se trata de un protocolo de comunicación establecido. El sistema en el DSP debe ser capar de entender las diferentes señales presentes en la línea telefónica y de igual manera el sistema debe de generar la respuesta acorde a las características de la línea telefónica para ser exitosamente enviada.

También se da una introducción de tonos DTMF (Dual Tone Multi-Frequency) aunque se estudiará a detalle en el siguiente capítulo.

## 4.1 Principios de telefonía

Un aparato telefónico se usa para originar y recibir llamadas telefónicas, el aparato es simple en su apariencia, pero realiza una cantidad sorprendente de funciones, las más importantes son las siguientes [12]:

- Solicita el uso del sistema telefónico al levantar el auricular.
- Indica que el sistema está disponible para el uso al recibir un tono, llamado tono de discar.
- Envía al sistema el número telefónico a llamar. Este número se inicia por la persona que llama al marcar el número por medio del teclado o al girar el disco.
- Indica el estado de la llamada en ejecución al recibir tonos que indican este estado (llamando, ocupado, etc.)
- Indica una llamada entrante al teléfono llamado por medio de una campanilla o de otros tonos audibles*.
- Transforma el lenguaje de una persona que llama en señales eléctricas para su transmisión a otro usuario a través del sistema*.

- Transforma las señales eléctricas recibidas de un usuario distante en audio para la persona llamada*.
- Ajusta automáticamente los cambios en la fuente de alimentación que recibe*.
- Señala al sistema cuando una llamada ha terminado al colgar la persona que llama el auricular*.

Para que un teléfono sea útil debe estar conectado a otro teléfono o a otro sistema telefónico, en los primeros días de la existencia del teléfono, los teléfonos estaban conectados uno al otro, sin conmutadores, a medida que la cantidad de teléfonos aumentó, esto resultó poco práctico y se estableció una central u oficina de conmutación para atender las conmutaciones y otras funciones.

### *4.1.1 Circuito telefónico*

El teléfono de cada usuario está conectado a una central que contiene equipos de conmutación, equipos de señalización y baterías que suministran corriente continua para hacer funcionar el teléfono como vemos en la figura 4.1. Cada teléfono está conectado a la central por medio de un lazo local de dos conductores, denominados un par. Uno de los conductores se llama *T* (del inglés tip) y el otro se llama *R* (del inglés ring), términos que se refieren a las partes de punta (tip) y anillo (ring) del conector (plug) usado en los tableros de conmutación antiguos.

Los interruptores en la central responden a los pulsos del discado o los tonos del teléfono que llama para conectar el mismo al teléfono llamado. Cuando se haya establecido la conexión, ambos teléfonos se comunican por medio de lazos acoplados por transformadores, utilizando la corriente suministrada por las baterías de la central [12].

Los teléfonos antiguos mandan el número de teléfono por medio de pulsos, mientras que los teléfonos modernos este número es enviado por medio de tonos audibles (figura 4.1) llamados tonos de marcado. Los teléfonos de pulsos contaban con un disco rotatorio que cerraba y abría el circuito telefónico en un cierto lapso de tiempo, el número de pulsos resultante de abrir y cerrar el circuito era determinado por que tanto se giraba el disco de marcado antes de soltarlo. Aunque todavía se conserva la compatibilidad de este método de marcado hoy en día prácticamente ha desaparecido siendo predominante la marcación por tonos.

* El sistema implementado en este trabajo debe reaccionar a las funciones telefónicas que son de llamado (“ring”), realizar el descolgado y el envió recepción de señales a través de la línea telefónica así como la decodificaciones de señales DTMF.

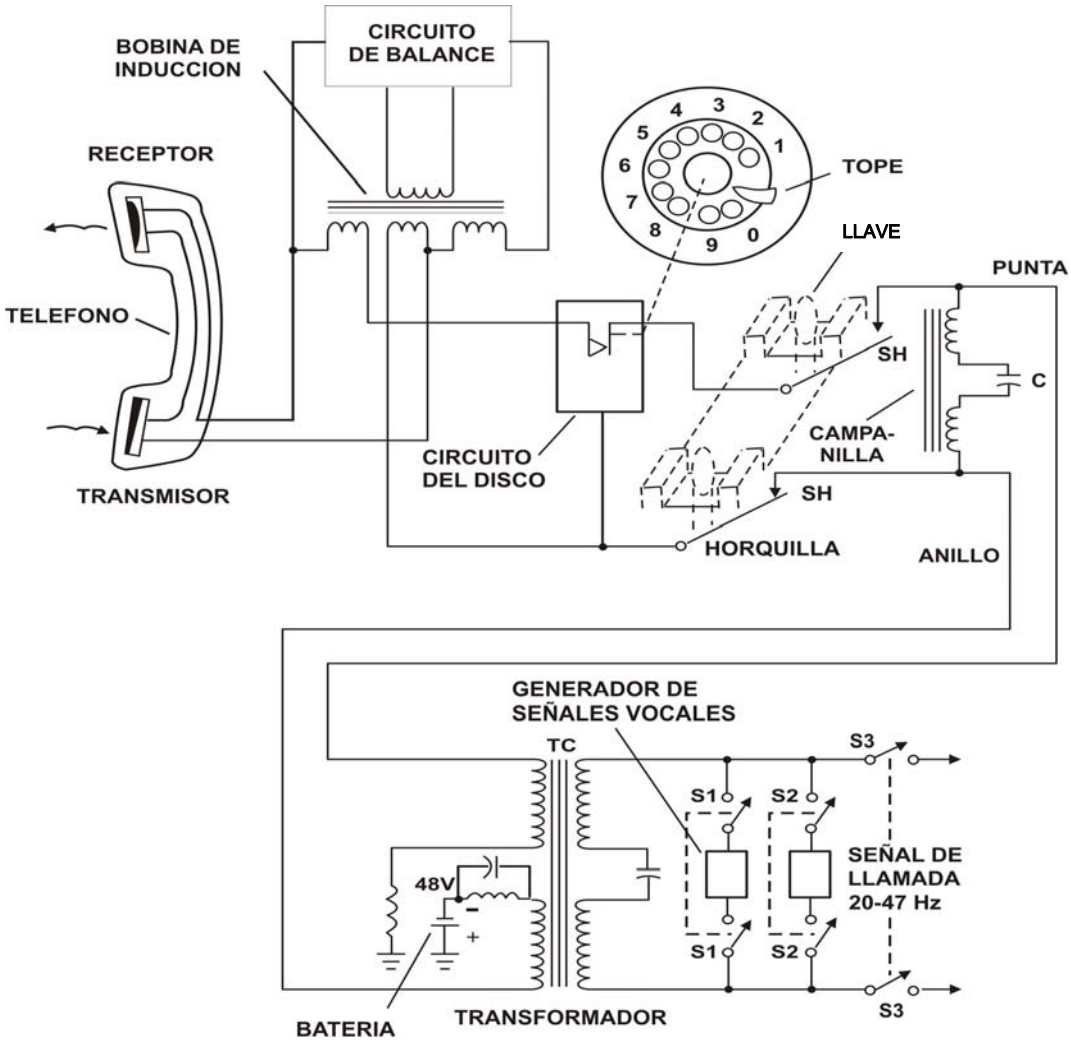



Figura 4.1 Circuito simplificado del aparato telefónico y central.

## *4.1.2 Realizando una Llamada*

Cuando el auricular del teléfono descansa en la horquilla, el peso del auricular aprieta los botones de la llave del mismo hacia abajo y los contactos (SH) están abiertos. Esta es la posición de *colgado*. El circuito entre el auricular y la central está abierto; sin embargo el circuito de *llamada* (campana) del teléfono está siempre conectado a la central, como vemos en la figura 4.1. El capacitor C, bloquea la circulación de la corriente continua de la batería, pero deja pasar la señal de la campana de corriente alterna. El circuito de la campana ofrece una impedancia elevada para las señales de voz de tal manera que no tiene ningún efecto sobre ellos.

Cuando se retira el auricular de su lugar, los botones provistos de resortes se levantan y los contactos (SH) se cierran. Esto completa el circuito a la central y la corriente circula en el circuito, ésta es la condición de *descolgado*. Los términos de *colgado*, *descolgado* y *colgar* provienen de las primeras épocas del teléfono cuando el receptor estaba separado y se *colgaba* de

un gancho cuando no estaba en uso. Esto explica también por qué mucha gente se refiere aún hoy como tubo al auricular actual.

La señal de *descolgado* informa a la central que alguien quiere hacer una llamada. La central devuelve un tono de discar al teléfono llamado para comunicar a la persona que llama, y la central está dispuesta a aceptar un número telefónico. El número telefónico puede ser referido también como una dirección.

La parte del teléfono con la que una persona habla, se denomina trasmisor, éste convierte la voz (energía acústica) en variaciones de corriente eléctrica (energía eléctrica) que se pueden transmitir a través de sistemas de trasmisión hasta el receptor del teléfono llamado. El transmisor telefónico más común que se usa actualmente es en principio igual al que inventó hace unos cien años Thomas A. Edison [12].

Tal como se observa en la figura 4.2, el transmisor consiste en una cápsula pequeña de dos piezas, llena con miles de gránulos de carbón. El frente y la parte posterior son conductores metálicos que se encuentran aislados entre sí. Un lado de la cápsula se mantiene fijo por medio de un soporte que es parte del gabinete del auricular.

El otro lado está unido a un diafragma que vibra en respuesta a las variaciones de presión del aire producido por la voz que recibe. Si los gránulos son obligados a acercarse más apretadamente, la resistencia de la cápsula disminuye. En cambio, si la presión sobre los gránulos es reducida, se alejan más y la resistencia aumenta. La **corriente** que circula a través de la cápsula del transmisor varía debido a las variaciones de la resistencia y de esta manera, la presión variable del aire que representa el habla, se convierte en una señal eléctrica variable, apta para ser transmitida al usuario que llama. Otros transmisores de carbón pueden tener diferencias en su construcción, pero su funcionamiento es semejante [12].

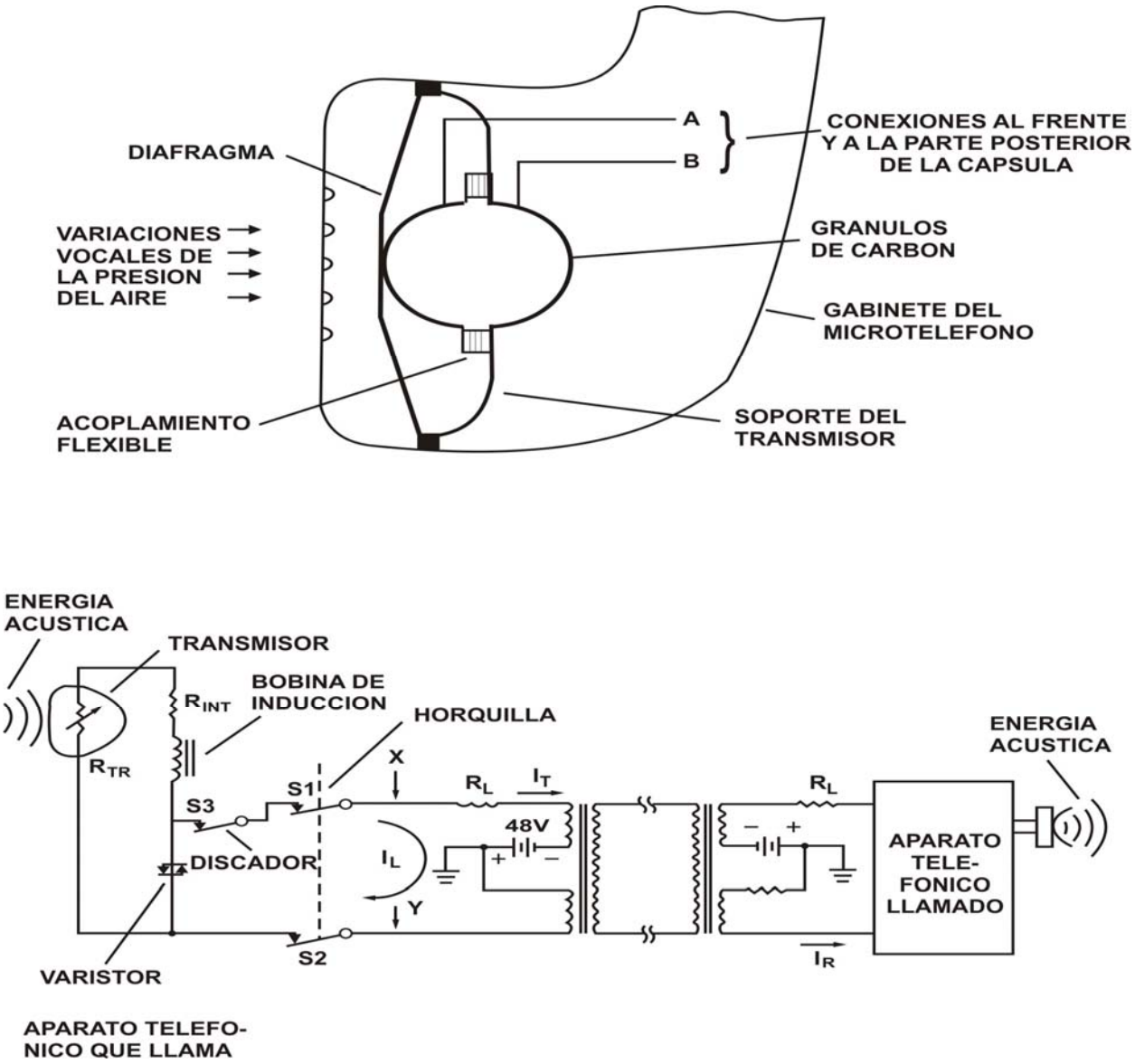



Figura 4.2. Transmisor Telefónico

**La Conexión de los Teléfonos**

La central posee varios conmutadores y relés, o su equivalente funcional en tecnología de circuitos integrados, que conectan automáticamente los teléfonos del que llama con el teléfono llamado, es decir, que la conexión ha sido realizada.

Si el tubo del teléfono llamado está descolgado cuando se intenta hacer la conexión, la central genera un tono de "ocupado" y lo envía al teléfono que llama. En el caso contrario, una señal de llamada es enviada al teléfono llamado para advertir al usuario que le espera una llamada. Al mismo tiempo, una señal de retorno es enviada al teléfono que llama para indicar que el teléfono llamado está sonando [12].

**Marcando al Teléfono**

Los circuitos telefónicos primitivos eran de punto a punto (sin conmutación), y el llamador obtenía la atención de la otra parte levantando el tubo y gritando ciertas frases. Esto no era muy satisfactorio y pronto se inventaron distintos dispositivos para la señalización automática. Uno que está aún en uso hoy es el *llamador polarizado* o *campana* que fue patentado en 1878 por Thomas A. Watson (el asistente de Graham Bell). Otros dispositivos electrónicos de llamada han reemplazado rápidamente los llamadores polarizados en nuevos diseños de teléfonos [12].

**Contestando la Llamada**

Cuando el usuario llamado descuelga el auricular en respuesta a la campana, el circuito hacia este teléfono se completa al cerrar los contactos (SH) en el aparato y la corriente del circuito circula por el teléfono llamado. Entonces la central retira la señal de llamada y el tono de retorno del circuito.

**La Conversación**

La parte del teléfono en la cual una persona habla se denomina *transmisor*, éste convierte la energía acústica de la voz en variaciones de una corriente eléctrica por medio de la variación de la corriente del lazo de acuerdo con la conversación de la persona que habla.

La parte del teléfono que convierte las variaciones de la corriente eléctrica en sonido que una persona puede escuchar se llama *receptor*. La señal producida por el transmisor es llevada por las variaciones de la corriente del lazo al receptor de la persona llamada. También, una pequeña parte de la señal del transmisor es realimentada al receptor de la persona que habla. Esto se llama *tono lateral* o *ruido local*. El *tono lateral* es necesario para que la persona que habla pueda escuchar su propia voz del receptor para poder determinar cuán fuerte está hablando. El *tono lateral* debe tener un nivel adecuado, porque un tono lateral muy fuerte puede causar que la persona hable demasiado despacio para tener una buena recepción del otro lado. A la inversa, un tono lateral muy bajo causará una voz demasiado fuerte que puede parecer un grito del otro lado del receptor [12].

**Terminación de la Llamada**

La llamada es terminada cuando cualquiera de las partes cuelga el auricular. La señal de *colgado* indica a la central liberar las conexiones de la línea. En algunas centrales, la línea queda liberada cuando cualquiera de las partes cuelga. En otras, la conexión es liberada solo cuando el abonado que llamó, cuelga.

### *4.1.3 Medios de transmisión*

El canal de comunicación provee la conexión entre el transmisor y el receptor, un canal físico puede ser un par de cables que transmiten señales eléctricas, cable de fibra óptica que transporta la información en forma de un rayo de luz modulado, o un canal submarino donde la información

es transmitida de forma acústica [13].

La red telefónica usa de forma extensiva líneas cableadas para la transmisión de señales de voz, pero también puede ser por medio de red celular, en el cual el canal de transmisión es electromagnético. Las señales transmitidas a través de estos canales sufren distorsión tanto en la amplitud como en la fase y además corrompida por ruido aditivo. El par de cable trenzado que se utiliza para conectar los teléfonos a la oficina central puede sufrir interferencia de los demás canales trenzados adyacentes produciendo interferencia [13].

### *4.1.4 Marcación por Multi-frecuencia de doble tono*

Algunos teléfonos en especial los antiguos envían sus números telefónicos por medio de pulsos de discado, mientras que los teléfonos modernos lo hacen por medio de tonos de audio.

Los teléfonos que usaban el discado por pulsos, poseían un disco rotativo manejado por un resorte, de 10 agujeros espaciados en forma equidistante como se observa en las figuras 4.1 y 4.3, que abrían y cerraban el circuito local en un ritmo predeterminado. La cantidad de pulsos de discado que resultan de una operación del disco está determinada por el giro del disco antes de soltarlo. Los pulsos de discado fueron concebidos originariamente para operar sistemas de conmutación electromecánicos. La inercia mecánica asociada con tales sistemas fijó el límite superior en el ritmo de funcionamiento de unas diez operaciones por segundo. De esta manera, los discos rotativos mecánicos de los teléfonos fueron diseñados para producir una tasa nominal de diez pulsos por segundo.

Si bien todas las facilidades de las redes telefónicas son actualmente compatibles con los teléfonos con discado por pulsos, las normas presentes prevén el uso generalizado del discado por tonos donde se sustituyen el numero de pulsos por un dígito representado por una señal audible como muestra la figura 4.3.

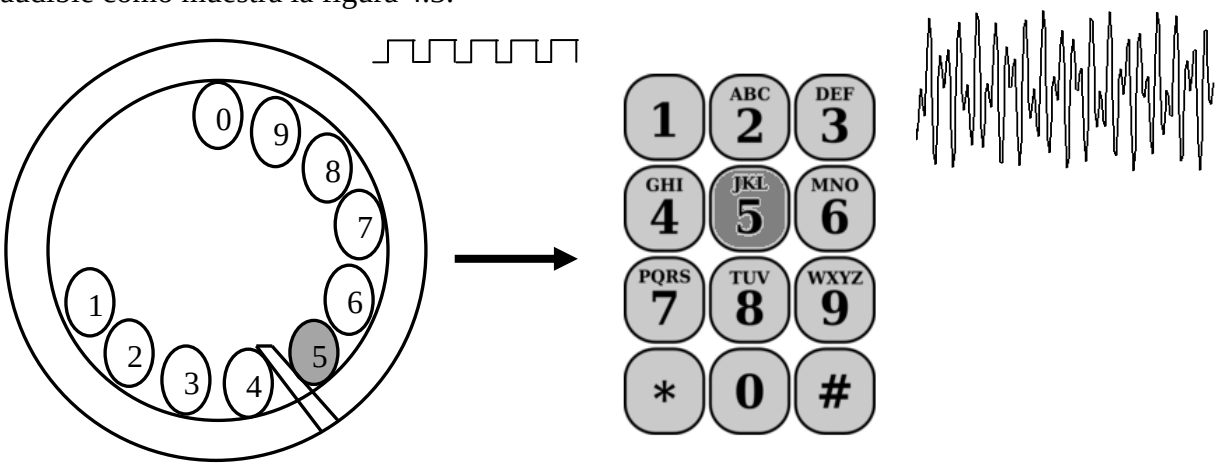



Figura 4.3. Equivalencia de marcación por pulsos a tonos.

La mayoría de los teléfonos modernos emplean el método más nuevo de usar tonos de audio para enviar el número telefónico, esto sólo se puede usar si la central está equipada para procesar los tonos, que en la actualidad es el estándar telefónico. En lugar del disco rotativo, estos teléfonos tienen un teclado con 12 teclas para los números del 0 al 9 y los símbolos * (asterisco) y # (número). Al apretar una de las teclas, un circuito electrónico genera dos tonos de salida que representan el número [12].

Para el presente trabajo la codificación DTMF forma parte esencial, ya que con base en ello se puede establecer una comunicación en ambos sentidos con el sistema, de tal forma que el usuario interactué al enviar dígitos codificados para cambiar el flujo de respuesta del sistema. El manejo de este tipo de codificación se explica más a detalle en el capítulo 5.

## 4.2 Transmisión de voz

Una señal analógica es continua y puede tener variaciones ya sea de amplitud o frecuencia como por ejemplo la señal se voz. La figura 4.4 muestra la relación de energía y frecuencia en una señal de voz. Se muestra que la frecuencia de la voz que contribuye al habla puede extenderse desde debajo de 100 hertz (Hz) hasta arriba de 6000 Hz. Sin embargo, se ha encontrado que la energía necesaria para que la voz sea inteligible se concentra en la banda de frecuencias entre 200 y 4000 Hz, después de este intervalo la energía disminuye lo suficiente como para no afectar el entendimiento del habla.

### *4.2.1 Ancho de banda*

El ancho de banda es la representación o comportamiento de un canal o señal en el dominio de la frecuencia, es una práctica común clasificar señales en términos de su contenido en frecuencia [14], como por ejemplo señales de baja frecuencia, de alta frecuencia o bien canales con comportamiento de paso bajas o paso de banda como es el caso del canal telefónico.

El circuito del teléfono está diseñado para limitar la banda de paso. Esto permite la transmisión de la frecuencia de la voz y limita frecuencias no deseadas de ruido.

Para eliminar señales no deseadas (ruido) que puede interferir con la conversación o causar errores en señales de control, el circuito de transmisión de señales telefónicas sólo permite el paso de ciertas frecuencias. El canal de voz cuenta con un intervalo de 0 a 4000 Hz pero no todo el ancho del canal está permitido por lo que el intervalo de frecuencias se limita de 300 a 3000 Hz coincidiendo con el intervalo donde la energía de voz es más alta y necesaria para ser inteligible como se muestra en la figura 4.4 [12].

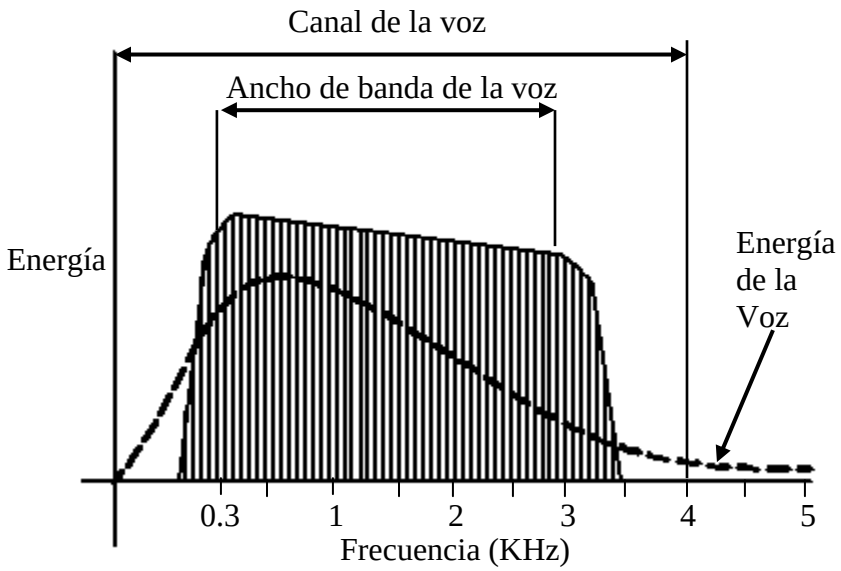



Figura 4.4 Ancho de banda de la voz

Dado que el ancho de banda es limitado para voz se necesitan la utilización de técnicas de compresión para hacer más eficiente la transmisión de datos por esta vía como por ejemplo la predicción lineal.

## 4.3 Resumen

La línea telefónica es el canal de comunicación muy utilizado desde su aparición soportando marcación por pulsos y tonos dando la pauta de cambiar la tecnología de marcado sin mayor contratiempo. En el presente trabajo se utiliza este canal de comunicación para la interacción de un usuario con nuestro sistema por medio de envío de voz sintética y envío de tonos de DTMF por lo que es importante tener en cuenta sus características y sus limitaciones para tener una comunicación exitosa.

# CAPITULO CINCO

# 5. Codificación en Multi-frecuencia de doble tono

La codificación en multi-frecuencia de doble tono o *DTMF* (Dual Tone Multi-Frequency), es un sistema de señalización usado para la marcación por tonos en los teléfonos, sustituyendo al antiguo modo de marcación por pulsos. Consiste en el envío de tonos dobles audibles que corresponden a un número del teclado telefónico, y al tener establecida la comunicación además del envío de señales de voz, se pueden mandar datos por medio de la codificación DTMF que pueden ser usados para el control de algún sistema.

En este capítulo se abordan diferentes métodos para la generación de tonos que constituyen la codificación DTMF, así como las técnicas para su decodificación de tal manera que se detecte el número presionado por un usuario. Se considera también que la implementación se debe realizar para que trabaje en tiempo real y sea lo más inmune posible al ruido para evitar tonos falsos.

## 5.1 Definición y generación de tonos DTMF

La codificación DTMF consiste en representar un número del teclado telefónico, con una señal dentro del ancho de banda de voz formada por un par de ondas senoidales de diferente frecuencia. Esta frecuencias están clasificadas en tono alto y un tono bajo como se muestra en la figura 5.1.

En general los teléfonos actualmente cuentan con un teclado con doce teclas, las cuales van desde el cero hasta la nueve y los símbolos * y #. Adicionalmente se incluyen cuatro teclas más en algunos teléfonos para funciones especiales dando un total de 16 teclas [12].

Al presionar una tecla causa que un circuito electrónico genere dos tonos dentro del ancho de banda de voz: un tono de baja frecuencia para cada renglón y un tono de alta frecuencia para cada columna, como se especifica en la figura 5.1 [12]. Por ejemplo al presionar el dígito 5 genera un tono de 770Hz y otro tono de 1336Hz.

Las frecuencias y el diseño del teclado es un estándar internacional, pero las tolerancias aceptables en las frecuencias individuales pueden variar en diferentes ciudades [12]. En Norte América es ±1.5% para la generación de los tonos mientras que para la recepción es de ±2%.

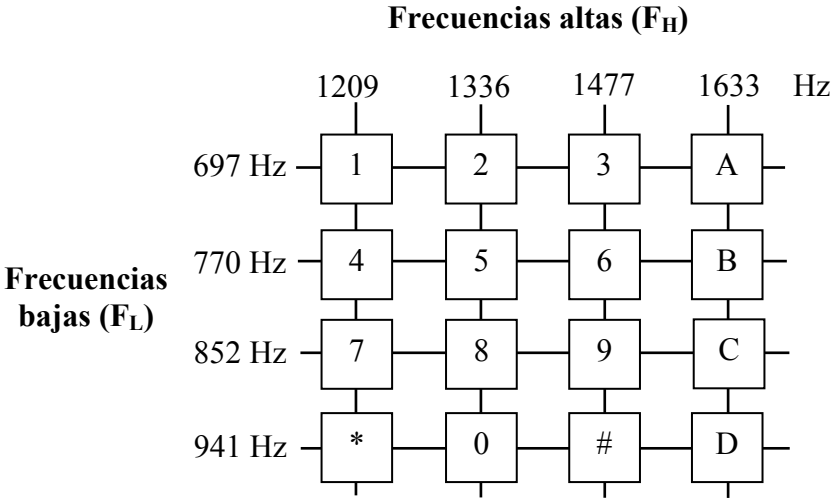



Figura 5.1. Matriz de frecuencias de un teclado telefónico

DTMF tiene un uso generalizado en diversas aplicaciones, como son, sistemas de correo electrónico y sistemas de servicios telefónicos de automatización, en donde el usuario puede seleccionar opciones desde un menú por medio de enviar señales DTMF desde el teléfono [15]. En la figura 5.2 se muestra el sistema de transmisión del tono DTMF desde el teléfono al sistema controlado.

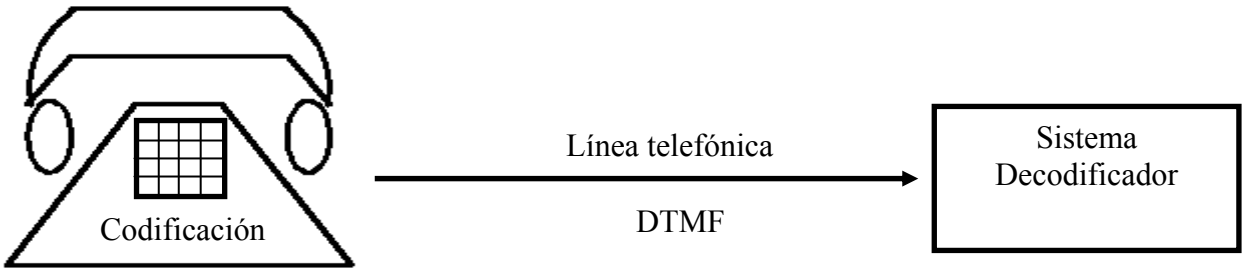



Figura 5.2. Sistema de transmisión telefónica de señales DTMF

Las frecuencias son seleccionadas de tal manera que eviten los armónicos y la intermodulación de los tonos, generando una señal fiable por lo que las frecuencias tienen las siguientes características:

- Ninguna frecuencia es múltiplo de otra.
- La diferencia entre dos frecuencias nunca es igual a cualquiera de las otras frecuencias.
- La suma de dos frecuencias no es igual a cualquiera de las otras frecuencias.

En el estándar de codificación DTMF cada tecla presionada en el teléfono genera la suma de dos tonos, expresado como muestra la ecuación (5.1), el diagrama a bloques se observa en la figura 5.3.

$$x(n) = \cos(2\pi f_L nT) + \cos(2\pi f_H nT) \tag{5.1}$$

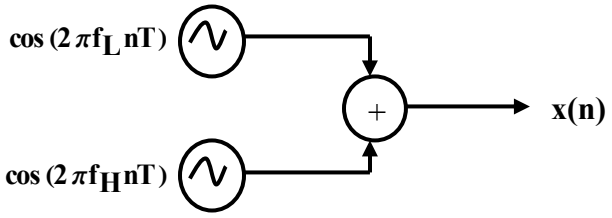



Figura 5.3. Diagrama a bloques generador DTMF.

Donde **T** es periodo de muestreo y las dos frecuencias $\mathbf{f_L}$ y $\mathbf{f_H}$, identifican de forma única la tecla que fue presionada de las mostradas en la figura 5.1. En la figura 5.4 se muestra la forma de onda que se genera al presionar una tecla, en este caso la tecla 8. En la Figura 5.5 se muestra el espectro de la señal donde se observa dos picos que corresponden a las frecuencias de 852 Hz y 1336 Hz que representan al dígito 8.

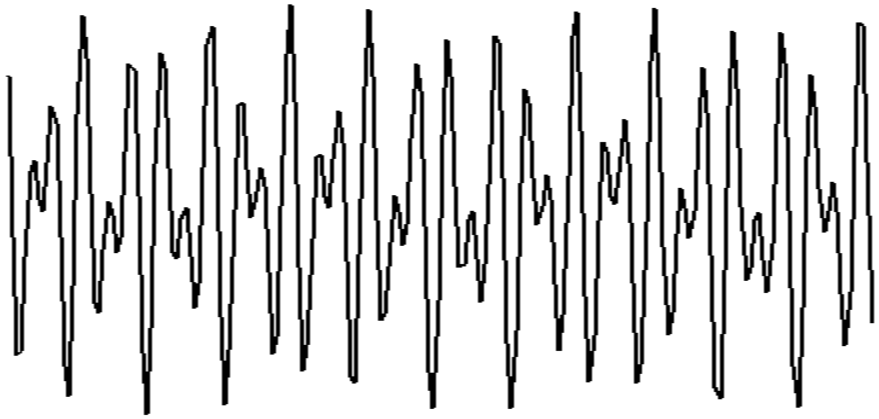


Figura 5.4. Forma de onda resultante al presionar una tecla.

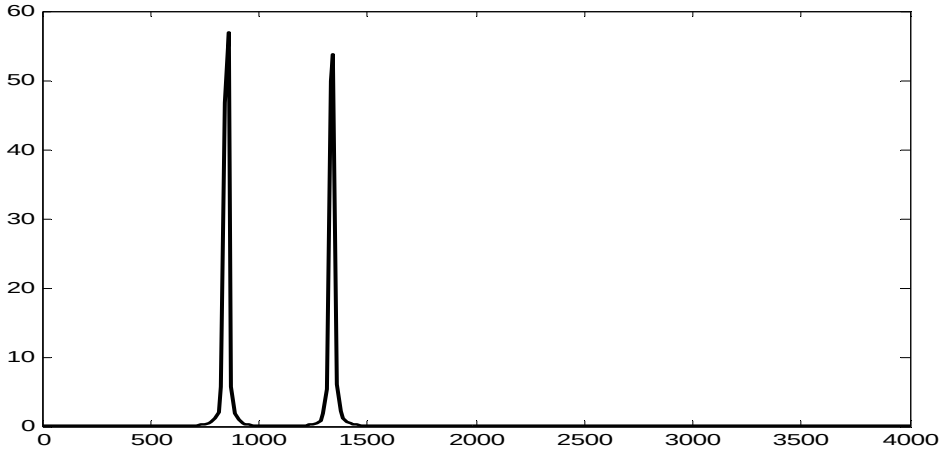



Figura 5.5. Espectro correspondiente al digito 8.

La señal DTMF debe cumplir con los requerimientos de tiempos de duración y espaciamiento de los tonos de los dígitos que son:

- Los dígitos son requeridos para ser transmitidos a una tasa menor de 10 por segundo [15].
- El espacio mínimo entre tonos es de 50 ms.
- Los tonos deben estar presentes por un mínimo de 40 ms.

La generación de dos tonos puede ser implementado usando dos generadores de señales senoidales conectados en paralelo, cada generador senoidal puede ser realizado utilizando diferentes métodos como:

1. La técnica de aproximación por polinomios [15].
2. Por medio de un oscilador recursivo.
3. El método de búsqueda de tablas.

### *5.1.1 Generación de tonos por aproximación de polinomios*

Para muchas aplicaciones en Procesamiento Digital de Señales (PDS), es necesario utilizar funciones matemáticas que los procesadores no pueden ejecutar de forma nativa, entonces esas funciones se descomponen en series donde se realizan operaciones básicas; como son: multiplicaciones, restas, sumas y divisiones. Cabe mencionar que se debe tener cuidado, ya que existen otras consideraciones, como son el desbordamiento, que es la condición en que el resultado de una operación aritmética excede la capacidad del registro usado para almacenar dicho resultado, por el efecto de multiplicación acumulación [15].

En general un tono se puede generar a través de las funciones coseno y seno, y estas funciones pueden ser expresadas como una expansión de serie infinita de Maclaurin, que se presenta en la ecuación (5.2), que es el caso especial cuando el punto de interés es igual a cero en la serie de Taylor. Las funciones coseno y seno se representan como muestran las ecuaciones (5.3) y (5.4) [24] [27].

$$f(x) = f(0) + f'(0)\,x + \frac{f''(0)}{2!}x^2 + \ldots + \frac{f^{(n)}(0)}{n!}x^n + \ldots \tag{5.2}$$

$$\cos(\theta) = 1 - \frac{1}{2!}\theta^2 + \frac{1}{4!}\theta^4 - \frac{1}{6!}\theta^6 + \ldots + (-1)^n \frac{\theta^{2n}}{(2n)!} + \ldots \tag{5.3}$$

$$\text{sen}(\theta) = \theta - \frac{1}{3!}\theta^3 + \frac{1}{5!}\theta^5 - \frac{1}{7!}\theta^7 + \ldots + (-1)^n \frac{\theta^{2n+1}}{(2n+1)!} + \ldots \tag{5.4}$$

Donde $\theta$ está en radianes, el símbolo "**!**" representa la operación factorial y $n$ es el n-ésimo término de la serie. La exactitud de la aproximación depende de cuántos términos son usados en la serie. Usualmente para valores grandes de $\theta$ es necesario proveer más términos, para tener una aproximación razonable. Sin embargo, en aplicaciones de PDS en tiempo real, sólo un número limitado de términos pueden ser utilizados. Usando aproximación Chebyshev las funciones coseno y seno pueden ser aproximadas como muestra las ecuaciones (5.5) y (5.6) [15].

$$\cos(\theta) = 1 - 0.001922\,\theta - 4.9001474\,\theta^2 - 0.264892\,\theta^3 + 5.04541\,\theta^4 + 1.800293\,\theta^5 \tag{5.5}$$

$$\text{sen}(\theta) = 3.140625\,\theta + 0.02026367\,\theta^2 - 5.325196\,\theta^3 + 0.5446788\,\theta^4 + 1.800293\,\theta^5 \tag{5.6}$$

Donde el valor de $\theta$ está definido en el primer cuadrante del plano cartesiano. Esto es $0 \le \theta < \pi/2$. Para $\theta$ en los otros cuadrantes las propiedades mostradas en las ecuaciones (5.7) y (5.8) pueden ser usadas para trasladar el valor al primer cuadrante.

$$\text{sen}(180° - \theta) = \text{sen}(\theta), \qquad \cos(180° - \theta) = -\cos(\theta) \tag{5.7}$$

$$\text{sen}(-180° + \theta) = -\text{sen}(\theta), \qquad \cos(-180° + \theta) = -\cos(\theta) \tag{5.8}$$

### 5.1.2 Generación de tonos con un oscilador digital recursivo

Considerando un resonador simple de segundo orden, cuya respuesta en frecuencia es determinada por un pico de frecuencia en $\omega_0$. Para establecer $\omega = \omega_0$ se posiciona el par de polos conjugados complejos como muestra la ecuación (5.9).

$$p_i = r_p\, e^{\pm j\omega_0} \tag{5.9}$$

Los polos se pueden ubicar en el plano *Z* como muestra la figura 5.6 [28].

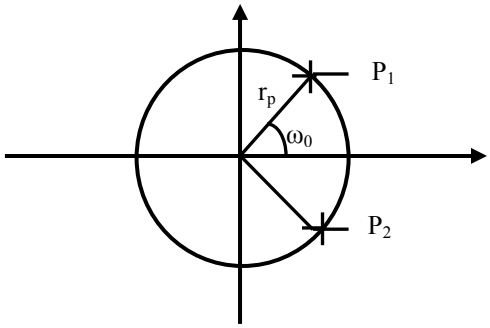



Figura 5.6. Polos del resonador en el plano *Z*.

La función de transferencia a partir del plano *Z* puede ser expresada como indica la ecuación (5.10).

$$H(z) = \frac{1}{(1 - P_1 z^{-1})(1 - P_2 z^{-1})} \tag{5.10}$$

Donde $P_1 = r_p e^{j\omega 0}$ y $P_2 = P_1^* = r_p e^{-j\omega 0}$ entonces resulta la ecuación (5.11)

$$H(z) = \frac{A}{(1 - r_p e^{j\omega 0} z^{-1})(1 - r_p e^{-j\omega 0} z^{-1})} = \frac{A}{1 - 2 r_p \cos(\omega_0) z^{-1} + r_p^2 z^{-2}} = \frac{A}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{5.11}$$

Donde **A** es la magnitud usada para normalizar el filtro a la unidad en $\omega_0$, esto es $|H(\omega_0)|=1$.

Donde $a_1 = -2r_p \cos\omega_0$ y $a_2 = r_p^2$.

La respuesta en magnitud normalizada está dada por la ecuación (5.12) y la condición puede resolverse como indica la ecuación (5.13).

$$|H(\omega_0)|_{z=e^{j\omega 0}} = \frac{A}{\left|(1 - r_p e^{j\omega 0} e^{-j\omega 0})(1 - r_p e^{-j\omega 0} e^{-j\omega 0})\right|} = 1 \tag{5.12}$$

$$A = \left|(1 - r_p)(1 - r_p e^{-2 j\omega 0})\right| = (1 - r_p)\sqrt{1 - 2 r_p \cos(2\omega_0) + r_p^2} \tag{5.13}$$

Como $H(z) = \frac{Y(z)}{X(z)} \longrightarrow Y(z)=X(z)H(z)$ y aplicando transformada ***Z*** inversa se obtiene ***y(n)*** mostrada en la ecuación (5.14) cuya implementación en forma directa se muestra en la figura 5.7.

$$y(n) = Ax(n) - a_1 y(n-1) - a_2 y(n-2) \tag{5.14}$$

Donde $r_p = 1$

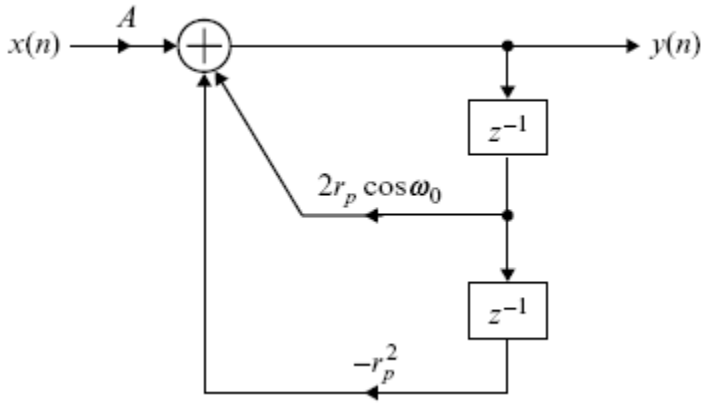



Figura 5.7 Filtro resonador de segundo orden.

A partir de la generalización de un sistema de segundo orden solo polo antes expuesto, se obtiene el oscilador recursivo que es muy útil para generar formas de onda senoidales. El método consiste en usar un resonador cuasi-estable de dos polos, donde los polos complejos conjugados que se encuentran sobre el círculo unitario ($\mathbf{r_p} = 1$) [15] y con una respuesta al impuso senoidal.

Si consideramos el sistema que se muestra en la figura 5.8, donde tenemos a **x(n)** como entrada **h(n)** como función de transferencia y por último **y(n)** como la salida, la salida debe ser **Asen($\omega_0$n) u(n)** y su respectiva transformada *Z* (TZ).

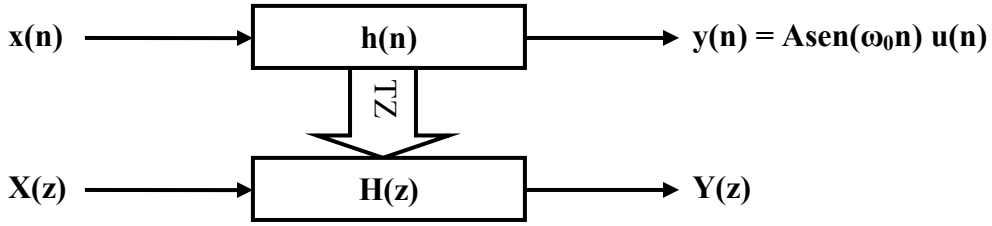



Figura 5.8. Sistema del oscilador digital.

Si se considera que se la entrada es un impulso **δ(n)** entonces:

$$\mathbf{Y(z)=X(z)H(z) = H(z) \implies h(n) = Asen(\omega_0 n)\, u(n)} \qquad (5.15)$$

(1 sobre X(z))

Aplicando la Transformada *Z* a **h(n) = sen($\omega_0$n) u(n)** se obtiene la ecuación (5.16).

$$\mathbf{H_S(z) = \frac{sen(\omega_0) z^{-1}}{1 - 2\cos(\omega_0) z^{-1} + z^{-2}}} \qquad (5.16)$$

Entonces:

$$\frac{Y(z)}{X(z)} = H(z) = \frac{\text{sen}(\omega_0) z^{-1}}{1 - 2\cos(\omega_0) z^{-1} + z^{-2}} \quad \Longrightarrow \quad Y(z) = \frac{X(z)\,\text{sen}(\omega_0) z^{-1}}{1 - 2\cos(\omega_0) z^{-1} + z^{-2}} \tag{5.17}$$

Desarrollando:

$$Y(z) - 2Y(z)\cos(\omega_0) z^{-1} + Y(z) z^{-2} = X(z)\,\text{sen}(\omega_0) z^{-1} \tag{5.18}$$

$$Y(z) = X(z)\,\text{sen}(\omega_0) z^{-1} + 2Y(z)\cos(\omega_0) z^{-1} - Y(z) z^{-2} \tag{5.19}$$

Aplicando Transformada *Z* inversa:

$$y(n) = \text{sen}(\omega_0)\, x(n-1) + 2\cos(\omega_0)\, y(n-1) - y(n-2) \tag{5.20}$$

Donde $\mathbf{x(n)} = \boldsymbol{\delta(n)} = \{\, \mathbf{1, 0, 0, 0, 0, 0, \ldots} \,\}$

Cabe mencionar que los polos son complejos conjugados, y están ubicados sobre el círculo unitario. Para una entrada impulso el sistema es cuasi-estable que es una característica importante para el oscilador. La ecuación con condiciones iníciales se muestra en (5.21).

$$y(0) = 0; \quad y(1) = \text{sen}(\omega_0); \quad y(n) = 2\cos(\omega_0)\, y(n-1) - y(n-2) \tag{5.21}$$

para **n** =2, 3,…

Una vez que el filtro recursivo es establecido con las condiciones iníciales, éste seguirá oscilando por siempre. Hay que notar que se necesita un resonador para cada tono que se genere y la frecuencia correspondiente para los dígitos necesarios de DTMF, deben ser generados y correctamente sumados.

### *5.1.3 Generación por búsqueda en tabla*

El método de búsqueda por tablas (Look up table: LUT) o generador de onda por tablas, es un método conceptualmente donde es fácil de generar una onda senoidal. La técnica implica simplemente en leer una serie de datos guardados, que representan valores discretos de muestras de la forma de onda a generar. Una señal periódica es generada al repetir cíclicamente la lectura de los datos guardados en memoria usando un apuntador circular [15].

En una tabla que contenga la representación de una onda seno, los valores son igualmente espaciados sobre un período de la forma de onda. Para **N**-puntos en la tabla de la onda seno puede ser calculado evaluando la ecuación (5.22).

$$x(n) = \operatorname{sen}\left(\frac{2\pi n}{N}\right), \quad n=0, 1, \ldots, N-1 \tag{5.22}$$

La generación de la forma de onda deseada al leer los valores guardados en la tabla a una tasa constante del paso **Δ**, se realiza por toda la tabla hasta el final y repitiéndose sin que el apuntador exceda el valor de **N-1**. La frecuencia de la forma de onda del seno generado depende del período **T**, del tamaño de la tabla *N* y del incremento **Δ** como muestra la ecuación (5.23).

$$f = \frac{\Delta}{NT} \text{ Hz} \tag{5.23}$$

Con la tabla diseñada para la forma de onda del seno de longitud **N**, la frecuencia **f** con una tasa de muestreo $\mathbf{f_s}$ puede ser generado usando un incremento del apuntador como muestra la ecuación (5.24).

$$\Delta = \frac{Nf}{f_S}, \quad \Delta \leq \frac{N}{2} \tag{5.24}$$

Para generar **L** muestras de la onda senoidal **x(l)**, **l=0**, **1, …, L-1**, se usa un apuntador circular **k** que cumple con la ecuación (5.25).

$$k = (m + l\Delta)_{\bmod N} \tag{5.25}$$

Donde **m** determina la fase inicial de la onda senoidal. Es importante notar que el paso **Δ** dado el la ecuación (5.24) puede ser un valor no entero, en consecuencia **(m+lΔ)** es un número real, es decir que consiste en un número compuesto por una parte entera y una fracción. Cuando el valor de la fracción de **Δ** es usado, entonces la muestra entre los puntos de la tabla debe ser estimada usando los valores de la tabla. La solución más fácil es redondear el número real al entero más cercano, sin embargo la mejor solución pero más compleja en su implementación es interpolar los dos valores de los puntos adyacentes [15].

Cabe señalar que con una tabla única de la señal seno, no es posible generar todas las frecuencias que intervienen en la generación DTMF, por características mencionadas en la sección 5.1, por lo que es necesario tener más tablas para su generación, por tanto no es un método adecuado para esta aplicación.

## 5.2 Decodificación de tonos DTMF

La correcta detección de un dígito requiere de un par de tonos válidos y de intervalos correctos de tiempo. En algunas aplicaciones es necesario detectar señales DTMF en presencia de voz, es importante que la forma de onda de la voz no sea interpretada como una señal válida de tono DTMF [15].

El decodificador DTMF es un bloque principal del presente trabajo y entra en acción cuando se

requiere interactuar con el usuario, es decir, después de que el sistema da una instrucción con voz sintética el sistema entra en espera a recibir un tono DTMF válido y dar una respuesta en función de ello.

Es necesario, de igual manera, poder descartar otras señales que se presenten en la línea telefónica como son voz, o interferencia, y también cuando se considera que no existe señal que decodificar, para ello se tienen los siguientes pasos:

- **Detección de silencio**: Nos sirve para descartar esas muestras antes del proceso de decodificación.
- **Cálculo de Energía**: Que permite considerar si se trata de ruido o señal no deseada y se descarta.
- **Adquisición de una ventana de datos**: Se adquiere toda la muestra para su evaluación.
- **Obtener espectro**: Se aplica un algoritmo para obtener el espectro del tono DTMF.
- **Se descarta DTMF no válido**: Se encuentran los tonos, la posición y la diferencia entre ellos, si no son congruentes con la especificación DTMF se descartan.
- **Se Decodifica**: Por último se encuentra el dígito correspondiente a partir de los tonos encontrados.

### 5.2.1 Especificaciones

La implementación de un receptor DTMF consiste en la detección de los tonos que componen la señal, una correcta validación del par de tonos y de los tiempos para determinar qué digito está presente y por último del correcto espaciamiento entre tonos. Adicionalmente es necesario realizar pruebas para medir el rendimiento del decodificador en presencia de la voz [15].

El receptor de DTMF necesita detectar frecuencias con tolerancias de ±1.5 por ciento como un tono válido. Los tonos que están afuera de ±3.5 por ciento o mayor no deben ser detectados como tonos válidos. Este requerimiento es necesario para prevenir que el detector interprete como señales válidas de DTMF la voz u otras señales. También es requerido que el receptor de DTMF funcione en un ambiente, donde la relación de señal a radio (SNR), se encuentre en 15dB hasta un intervalo dinámico de 26dB [15].

Otro de los requerimientos del receptor, es la habilidad de detectar señales DTMF cuando dos tonos son recibidos con diferente nivel. El tono de alta frecuencia puede ser recibido a un nivel menor, que el tono de baja frecuencia, debido a la respuesta en magnitud del canal de comunicación. A esta diferencia entre los niveles se le conoce como *twist hacia adelante*. En el caso en que el tono de baja frecuencia es recibida con el nivel más bajo que el tono de alta frecuencia, se le conoce como *twist hacia atrás*. El detector DTMF debe funcionar con un

máximo de 8dB de twist hacia adelante y 4dB con twist hacia atrás. Al final el requerimiento es que el receptor funcione en presencia de voz sin que identifique erróneamente la señal de voz como un digito DTMF válido [15].

### 5.2.2 Algoritmo de detección de tonos

El principio de la detección DTMF es examinar la energía de la señal recibida en las frecuencias de DTMF, para determinar que el par de tonos que han sido recibidos son válidos o no.

El algoritmo de detección DTMF puede ser implementado usando el algoritmo de la Transformada Rápida de Fourier (FFT), o una implementación de banco de filtros en paralelo pasa banda, centrados en las frecuencias que conforman los tonos DTMF. La FFT puede ser utilizada para calcular las energías de *N* frecuencias espaciadas uniformemente. Para llevar a cabo la detección requerida para las ocho frecuencias DTMF con un porcentaje de desviación de ±1.5 y con 256 puntos de FFT, con una frecuencia de muestreo de 8KHz, resulta más eficiente una implementación de un banco de filtros dado que se requiere un número reducido de tonos a ser detectados [15].

De acuerdo al estándar telefónico de la figura 5.1, sólo ocho frecuencias son las que interesan ser detectadas, es más eficiente usar directamente la transformada discreta de Fourier (DFT), mostrada en la ecuación (5.26) para los ocho diferentes valores de **k** que corresponden a las frecuencias definidas de la codificación DTMF. Los coeficientes DFT pueden ser calculados de una manera más eficiente utilizando el algoritmo de Goertzel, el cual puede ser interpretado como un banco de filtros resonadores para cada frecuencia **k** como se muestra en la figura 5.9. En esta figura se muestra **x(n)** como la señal de entrada del sistema, $\mathbf{H_k(z)}$ es la función de transferencia del filtro a la frecuencia **k** y **X(k)** es la correspondiente salida del filtro [15].

$$\mathbf{X(k)} = \sum_{\mathbf{n=0}}^{\mathbf{N-1}} \mathbf{x(n)}\, \mathbf{W_N}^{\mathbf{kn}} \tag{5.26}$$

Donde $\mathbf{W_N}^{\mathbf{kn}} = e^{-\mathbf{j}\left(\frac{\mathbf{2\pi}}{\mathbf{N}}\right)\mathbf{kn}}$

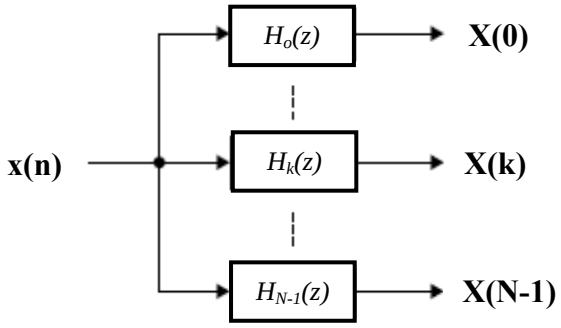



Figura 5.9.Banco de filtros resonadores de Goertzel.

Considerando el factor [14],[28].

$$W_N^{-kN} = e^{j\left(\frac{2\pi}{N}\right)kN} = e^{j2\pi k} = 1 \tag{5.27}$$

Multiplicando a la definición de la DFT ecuación (5.26) por (5.27).

$$X(k) = W_N^{-kN} \sum_{m=0}^{N-1} x(m)\, W_N^{km} = \sum_{m=0}^{N-1} x(m)\, W_N^{-k(N-m)} \tag{5.28}$$

Cambiando variables y si **N = n** entonces se define la secuencia:

$$X(k) = y_k(n) = \sum_{m=0}^{N-1} x(m)\, W_N^{-k(n-m)} \tag{5.29}$$

La ecuación (5.29) puede ser interpretada como una convolución de la secuencia **x(n)** de duración finita, **0≤n≤N-1** con la secuencia $W_N^{-kn}u(n)$. En consecuencia $y_k(n)$ puede ser vista como la salida de un filtro con respuesta al impulso $W_N^{-kn}u(n)$, es decir

$$h_k(n) = W_N^{-kn}\, u(n) \tag{5.30}$$

Debido a la entrada **x(n)** de longitud finita, la ecuación (5.29) puede ser expresada como

$$y_k(n) = x(n) * W_N^{-kn}\, u(n) \tag{5.31}$$

De la ecuación (5.28) y (5.29) y del hecho que **x(n)=0** para **n<0 y n > N**, se muestra que

$$X(k) = y_k(n)\,|_{n=N} \tag{5.32}$$

Esto es $X(k)=y_k(n)$ es la salida del filtro $H_k(z)$ en el tiempo **n=N**.

Aplicando la transformada ***Z*** a la ecuación (5.31) y por el teorema de convolución

$$Y_k(z) = X(z)\, \frac{1}{1 - W_N^{-k} z^{-1}} \tag{5.33}$$

La función de transferencia para el filtro **k** de Goertzel está definido como

$$H_k(z) = \frac{Y_k(z)}{X(z)} = \frac{1}{1 - W_N^{-k} z^{-1}} \tag{5.34}$$

Este filtro contiene polos ubicados sobre el círculo unitario a la frecuencia de $\omega_k = 2\pi k/N$. Así la DFT puede ser calculada en su totalidad filtrando los bloques de entrada usando un banco paralelo de **N** filtros definidos por la ecuación (5.34), donde cada filtro tiene un polo en la frecuencia correspondiente de la DFT. Como el algoritmo de Goertzel calcula *N* coeficientes de la DFT, el parámetro *N* se debe seleccionar para asegurar que **X(k)** esté cerca de la frecuencia $f_k$ de la DTMF. Esto puede ser realizado si se escogen **N** tal que

$$\frac{f_k}{f_s} = \frac{k}{N} \tag{5.35}$$

Donde la frecuencia de muestreo $f_s$ = 8KHz es utilizada por la mayoría de sistemas de comunicaciones de voz.

El diagrama de la función de transferencia $H_k(z)$ se muestra en la figura 5.10. Ya que los coeficientes $W_N^{-k}$ son valores complejos, el cálculo de cada nuevo valor de $y_k(n)$ requiere de operaciones complejas. Todos los valores intermedios $y_k(0)$, $y_k(1)$, …, $y_k(N)$ deben ser calculados en orden para obtener la salida final $y_k(N) = X(k)$. Por lo tanto el cálculo del algoritmo requiere de **4N** multiplicaciones y sumas complejas para calcular **X(k)** para cada frecuencia con índice **k**.

Multiplicando tanto el numerador como el denominador de $H_k(z)$ en la ecuación (5.34) por el factor $(1-W_N^k z^{-1})$ se tiene

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{(1 - W_N^{-k} z^{-1})(1 - W_N^k z^{-1})} = \frac{1 - e^{\frac{-j2\pi k}{N}} z^{-1}}{1 - 2\cos\left(\frac{2\pi k}{N}\right) z^{-1} + z^{-2}} \tag{5.36}$$

Si dividimos $H_k(z)$ en dos factores tendríamos

$$H_k(z) = \left\{\frac{1 - W_N^k z^{-1}}{1}\right\}\left\{\frac{1}{1 - 2\cos\left(\frac{2\pi k}{N}\right) z^{-1} + z^{-2}}\right\} \tag{5.37}$$

Sería equivalente a la expresión

$$H_k(z) = (H_1)(H_2) = \left\{\frac{Y(z)}{V_k(z)}\right\}\left\{\frac{V_k(z)}{X(z)}\right\} \tag{5.38}$$

Desarrollando $H_1$ se tiene

$$H_1 = \left\{\frac{Y(z)}{V_k(z)}\right\} = 1 - W_N^k z^{-1} \quad \Rightarrow \quad Y_k(z) = V_k(z) - W_N^k z^{-1} V_k(z) \tag{5.39}$$

Aplicando TZI (Transformada Zeta Inversa)

$$y_k(n) = v_k(n) - W_N^k v_k(n-1) \tag{5.40}$$

El cálculo para **n = N** de la parte no recursiva de $y_k(N)$ es expresado como

$$X(k) = y_k(N) = v_k(N) - e^{\frac{-j2\pi f_k}{f_s}} v_k(N-1) \tag{5.41}$$

Desarrollando $H_2$, se tiene

$$H_2 = \left\{ \frac{V_k(z)}{X(z)} \right\} = \frac{1}{1 - 2\cos\left(\frac{2\pi k}{N}\right) z^{-1} + z^{-2}} \tag{5.42}$$

Despejando $V_k(z)$

$$V_k(z) = X(z) + 2\cos\left(\frac{2\pi k}{N}\right) z^{-1} V_k(z) - z^{-2} V_k(z) \tag{5.43}$$

Aplicando TZI nos da la parte recursiva como

$$v_k(n) = x(n) + 2\cos\left(\frac{2\pi k}{N}\right) v_k(n-1) - v_k(n-2) \tag{5.44}$$

Para **n = 0,1,2,…, N**. donde cada iteración requiere de una multiplicación real y dos sumas reales.

El diagrama de la función de transferencia de la ecuación (5.36) se muestra en la figura 5.10 usando la implantación de forma directa tanto la parte recursiva como la no recursiva. La parte recursiva se muestra en la parte izquierda de los elementos de retrasos y la parte no recursiva en la parte derecha. Donde la salida $y_k(n)$ se requiere sólo el tiempo **N**, entonces se necesita calcular la parte no recursiva para el filtro en la iteración **N**.

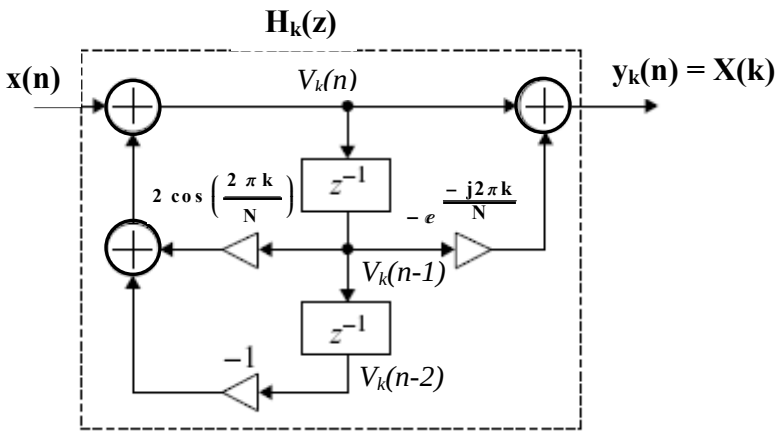



Figura 5.10. Algoritmo Goetzel

Una simplificación adicional que se puede hacer al algoritmo, es obtener la magnitud cuadrada de **X(k)** que es lo que se necesita para la detección del tono. De la ecuación (5.41) la magnitud cuadrada de **X(k)** es calculada como [15]

$$|X(k)|^2 = V_k^2(N) - 2\cos\left(\frac{2\pi k}{N}\right) V_k(N) V_k(N-1) + V_k^2(N-1) \qquad (5.45)$$

Por lo tanto ya no es necesaria la complejidad aritmética de la ecuación (5.41) y la ecuación (5.45) sólo requiere un coeficiente **$2\cos(2\pi k/N)$** para cada **$|X(k)|^2$** a ser evaluado. Ya que se tienen ocho posibles tonos a ser detectados se necesitan ocho filtros descritos por las ecuaciones (5.44) y (5.45). El algoritmo opera como si cada filtro es sintonizado a una de las ocho frecuencias definidas para el detector DTMF [15]. En la tabla 5.1 se muestran las constantes **k** para **N** = 205.

| $k_i$ | Frecuencia ($f_k$) | $K = Nf_k/f_s$ | $2\cos(2\pi K/N)$ |
|---|---|---|---|
| 1 | 697 | 18 | 1.7033 |
| 2 | 770 | 20 | 1.6359 |
| 3 | 852 | 22 | 1.5623 |
| 4 | 941 | 24 | 1.4829 |
| 5 | 1209 | 31 | 1.1631 |
| 6 | 1336 | 34 | 1.0088 |
| 7 | 1477 | 38 | 0.7901 |
| 8 | 1633 | 42 | 0.5594 |

Tabla 5.1. Valores de las constantes utilizadas en el algoritmo de Goertzel

## 5.3 Consideraciones de implementación

En la figura 5.11 se muestra el diagrama de flujo del algoritmo de detección, al inicio de cada trama de longitud **N** , el estado de las variables **x(n)**, **vk(n)**, **vk(n-1)**, **vk(n-2)** e **yk(n)** para cada uno de los ocho filtros de Goertzel y la energía se establece a 0. Para cada muestra la parte recursiva de cada filtro definida en la ecuación (5.44), es ejecutada considerando que previamente se calculó **k** y **2cos(2πk)**. El final de cada trama donde **n=N** la magnitud cuadrada $|\mathbf{X(k)}|^2$ para cada frecuencia DTMF es calculada con base en la ecuación (5.45) para cada **k**. Para determinar que una muestra se trata de un dígito DTMF es necesario realizar diferentes pruebas con ello se descartan muestras que no representan una codificación DTMF [15].

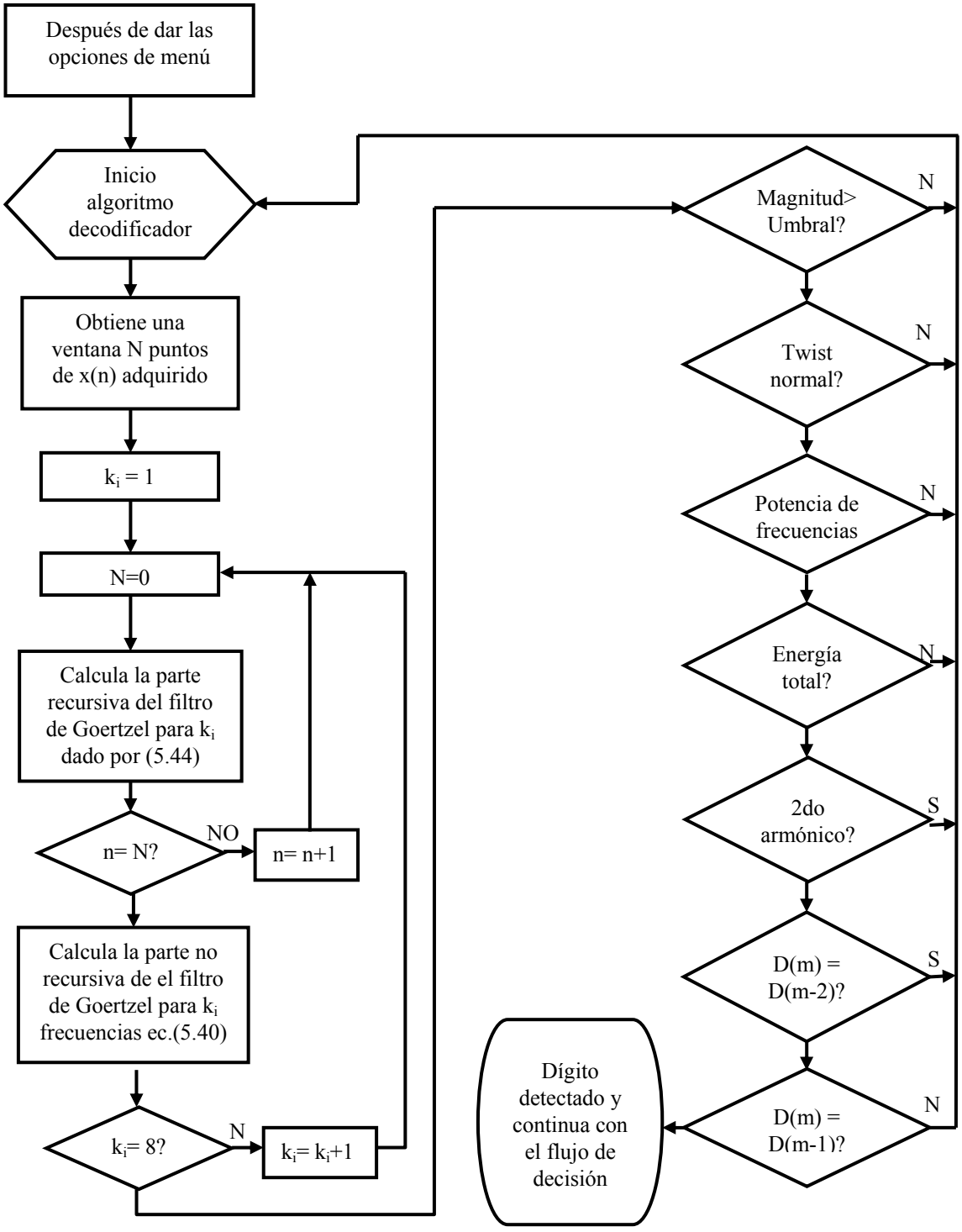



Figura 5.11. Implementación del detector de DTMF

### 5.3.1 Prueba de Magnitud

Generalmente el receptor de DTMF se espera que opere en un intervalo promedio de -29dBm a 1dBm de SNR, en un caso extremo, pero que podría pasar. Por lo tanto debajo de los -29dBm el decodificador DTMF no debe decodificar el dígito [15].

Para la prueba de magnitud se utiliza la ecuación (5.45) que nos da la magnitud al cuadrado $|\mathbf{X(k)}|^2$ y se calcula para cada frecuencia del DTMF.

### 5.3.2 Prueba de Twist

Los tonos pueden ser atenuados debido a la respuesta en frecuencia del sistema telefónico. Por consecuencia no se espera que tanto las frecuencias altas como las frecuencias bajas sean recibidas con la misma amplitud en el receptor, aunque se transmitan con la misma intensidad. Twist es la diferencia, en decibeles, entre los niveles de la frecuencia alta y la frecuencia baja. Generalmente los dígitos DTMF son generados con algún twist hacia adelante, es decir, amplitud de la frecuencia alta es un poco mayor que la frecuencia baja, esto para compensar las pérdidas en la alta frecuencia a lo largo de la línea telefónica. Las tolerancias de twist son generalmente definidas según la administración o país del que se trate. Por ejemplo en Australia se permite una diferencia de 10dB, para Japón sólo 5dB, y AT&T recomienda no más de 4dB hacia adelante o bien 8dB hacia atrás [15]. La prueba consiste en determinar la diferencia de amplitud entre las frecuencias y verificar que caigan en una tolerancia válida según el país donde se realice.

### 5.3.3 Prueba de potencia de las frecuencias

Esta prueba es realizada para prevenir que se detecte el ruido como dígitos válidos. Si efectivamente existen tonos DTMF válidos, el nivel de potencia en estas dos frecuencias debería ser mucho mayor que el nivel de potencia de otras frecuencias. Esta prueba consiste en comparar grupos de frecuencias con otros grupos de otras frecuencias, la diferencia debe ser predominantemente mayor que el umbral que se tenga definido en cada grupo [15].

### 5.3.4 Prueba de energía total

Similar a la prueba anterior, esta prueba se realiza para rechazar ruido que pueda estar presente en la línea telefónica (como voz) y con ello hacer más robusto el decodificador. Para realizar la prueba se necesitan tres diferentes constantes, $\mathbf{c_1}$, $\mathbf{c_2}$ y $\mathbf{c_3}$. La energía de tono detectado en el grupo de baja frecuencia se establece en *c1*, la energía detectada en el tono del grupo de alta frecuencia se toma en $\mathbf{c_2}$ y la suma de las dos energías se guarda en $\mathbf{c_3}$. Cada uno de estos términos debe ser mayor que la suma de la energía de la salida de los ocho filtros. La energía total se calcula como

$$E = \sum_{k=1}^{8} |X(k)|^2 \tag{5.46}$$

### 5.3.5 Prueba del segundo armónico

El objetivo de esta prueba es rechazar la voz que contenga una armónica cerca de $\mathbf{f_k}$ que pudiera ser detectada como un tono DTMF. Ya que los tonos DTMF son señales senoidales puras, contienen una energía del segundo armónico muy tenue, por el contrario, la energía de la voz similar a este segundo armónico es considerable. Para probar el nivel del segundo armónico el decodificador debe evaluar el segundo armónico de las frecuencias de los ocho tonos DTMF. Estas frecuencias de segundo armónico pueden ser evaluadas también por el algoritmo de Goertzel (1394Hz, 1540Hz, 1704Hz, 1882Hz, 2418Hz, 2672Hz, 2954Hz y 3266Hz) [15].

### 5.3.6 Decodificador de dígito

Finalmente, si las pruebas realizadas anteriormente se verificaron correctamente, el par de tonos son decodificados como un entero entre 1 y 16. Así el decodificador del dígito es implementado como:

$$\mathbf{D(m)=C+4(R-1)} \tag{5.47}$$

Donde **D(m)** es detector de dígito por trama **m, m=0, 1, 2, …** es el índice de la trama, **C** es el índice de columnas que representa la frecuencia alta que ha sido detectada y **R** es el índice del renglón que representa a la frecuencia baja que ha sido detectada. Por lo regular en las tramas donde no es detectado algún tono DTMF dado que alguna prueba falló, entonces se puede representar el dígito como "-1" que indica que no hay un dígito presente.

Para que un nuevo dígito sea válido, **D(m)** debe ser el mismo para dos tramas sucesivas **D(m-2) = D(m-1)**. Si el dígito es válido para más de dos tramas sucesivas el detector manda como resultado el digito previamente validado [15].

Existen dos razones para verificar tres dígitos sucesivos:

1. La verificación elimina la necesidad de decodificar el dígito cada vez que el tono esté presente. Como el tono está presente, puede ser ignorado hasta que este cambie.

2. Comparando el dígito **D(m-2)**, **D(m-1)** y **D(m)** se mejora la inmunidad al ruido y a la voz.

## 5.4 Resumen

Existen diferentes formas de generar y decodificar el tono DTMF y estos métodos determinan el tiempo de respuesta del sistema pero hay que tomar en cuenta también el número de validaciones que se llevan a cabo para determinar si el dígito es válido o no. Otro punto a tomar en cuenta es si se requiere la decodificación en todo momento, o bien en un intervalo de tiempo determinado donde no se tenga presencia de voz humana y considerando que la decodificación se realiza en tiempo real.

El algoritmo de Goertzel resulta una buena herramienta para detección de tonos en el decodificador DTMF ya que ahorra tiempo de procesamiento y su implementación no requiere mayor complejidad. También es necesario tener en cuenta los requerimientos mínimos de tiempo para llevar a cabo de forma exitosa la decodificación. Además en este trabajo se evaluaron las pruebas que se pueden realizar a la señal para evitar tonos falsos y que se interpretaron como valores válidos de DTMF.

# CAPITULO SEIS

# 6. Diseño y desarrollo del sistema

El Procesamiento Digital de Señales (PDS) se distingue de otras áreas de la ciencia de la computación por manejar señales que generalmente son originadas por sensores del mundo real, como pueden ser vibraciones sísmicas, voz, imágenes, ondas, sonido, etc. El PDS está constituido por el análisis matemático, algoritmos y técnicas usadas para la manipulación de estas señales después de haber sido convertidas a una forma digital [20].

El rápido avance en la tecnología digital en los años recientes ha llevado a la implementación de sofisticados algoritmos de PDS que hacen posible la implementación de tareas en tiempo real; de igual forma, ha llevado a la investigación desarrollar algoritmos más eficientes para aplicaciones, donde puede resultar difícil o imposible de implementar en forma analógica.

Existen muchas ventajas al usar técnicas digitales para el procesamiento de señales sobre los tradicionales dispositivos analógicos. Algunas de estas ventajas son: flexibilidad, reproducibilidad, confiabilidad y complejidad [15].

Para realizar aplicaciones del PDS en tiempo real, se necesita efectuar una gran cantidad de operaciones matemáticas en un tiempo limitado, para ello se desarrollaron los Procesadores Digitales de Señales (DSP), circuitos integrados mejorados, que permiten la implementación en sistemas de tiempo real.

En este capítulo se muestra el hardware y software utilizado para llevar a cabo la decodificación de tonos DTMF, como también la generación de voz sintética, para ello se implementan diferentes algoritmos de PDS, para el acondicionamiento y selección de señales provenientes de la línea telefónica para llevar a cabo nuestros objetivos.

El sistema diseñado consiste en un autómata conectado a la línea telefónica, cuando se marca al número telefónico correspondiente, el sistema detecta la llamada realizando las función de descolgado, posteriormente pone en la línea una señal de voz sintética dando un menú de opciones y a continuación se pone en espera de un tono DTMF, que al detectarlo toma la decisión correspondiente y responde con voz sintética sobre la línea. A continuación se describe el funcionamiento.

# 6.1 Descripción General del sistema

El sistema a diseñar consiste en una terminal autónoma interactiva conectada a la línea telefónica; donde por un lado tenemos a un usuario que interactúa con el sistema a través de una terminal telefónica y por el otro, conectado por medio del RJ-11 a la línea, el DSP con sus dispositivos necesarios para el acoplamiento. El usuario manda instrucciones por medio de tonos DTMF al presionar dígitos, el sistema responde con voz sintética. En la figura 6.1 se muestra un diagrama a bloques general del sistema, y la figura 6.2 un diagrama de flujo de su funcionamiento.

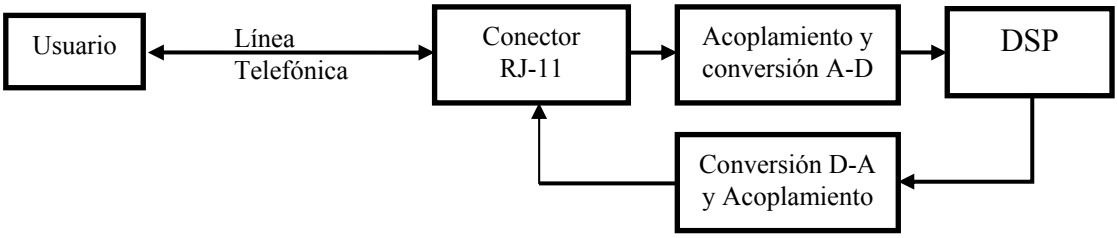



Figura 6.1. Diagrama a bloques del sistema.

Para realizar la síntesis de voz es necesario contar con los parámetros necesarios para su reproducción, estos parámetros se obtienen a partir de una muestra de voz real la cual es procesada y analizada fuera de línea. Contando con estos parámetros es posible realizar la síntesis de voz en tiempo real (en línea).

Para demostrar la funcionalidad del sistema se diseñó un menú general con una profundidad de tres niveles, representando una pizzería, cuando se da el menú al usuario por medio de voz sintética, el sistema se queda en espera de un tono DTMF, debido a esta característica el sistema no decodifica tonos DTMF en presencia de voz sintética.

En la operación, las selecciones que realiza el usuario se almacenan en un arreglo que tiene una longitud de 100 llamadas, pudiendo almacenar las 3 opciones realizadas por el usuario en una llamada, a esto le podemos llamar base de datos de almacenamiento de selección, el cual puede ser explotado en futuros trabajos que extiendan la funcionalidad presentada.

El sistema puede operar con una relación de señal a ruido (SNR), suficiente para considerarse un sistema confiable y evitar en lo posible la detección de tonos DTMF falsos teniendo como resultado datos confiables.

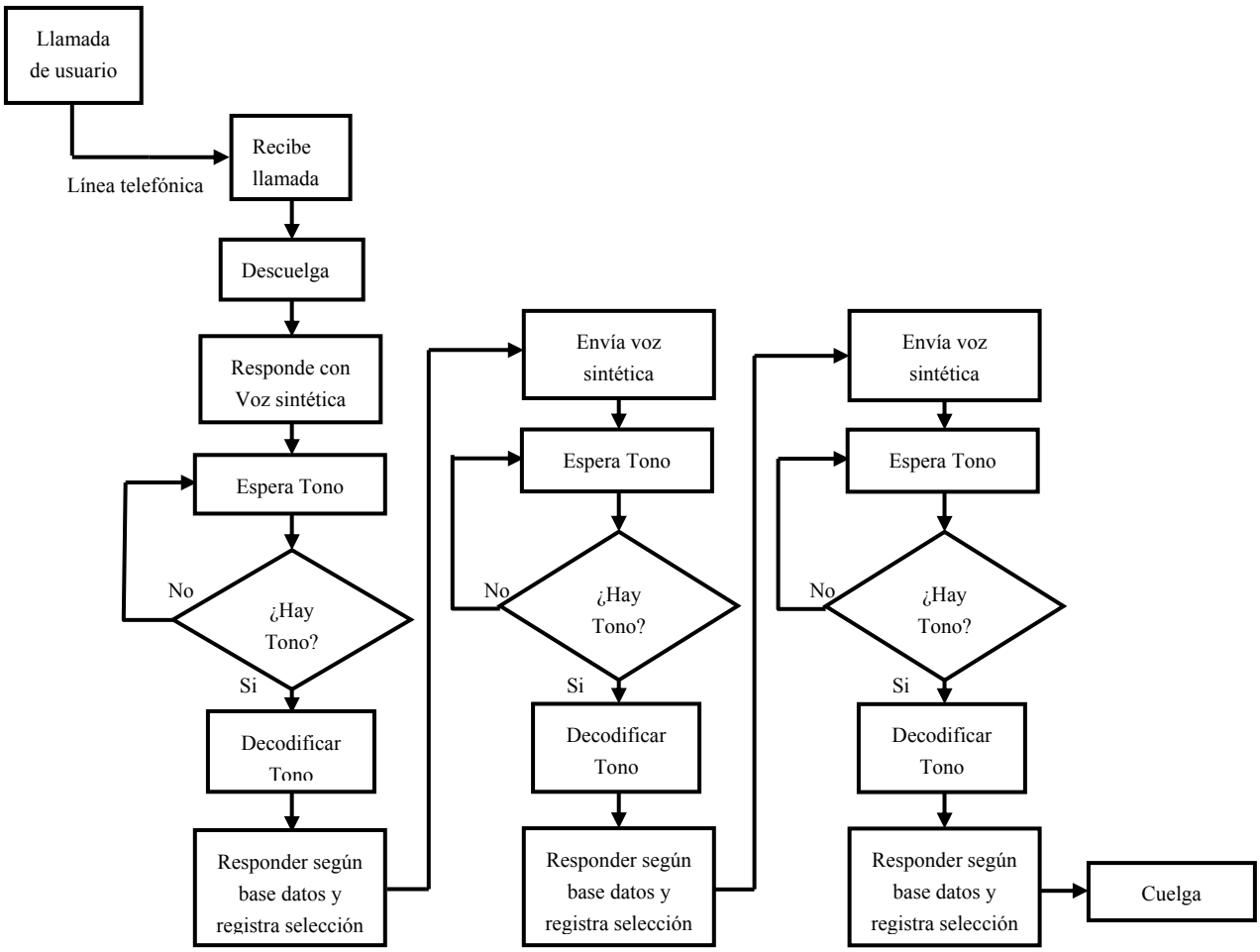



Figura 6.2. Diagrama de flujo del funcionamiento.

## 6.2 Elementos básicos de un sistema de PDS en tiempo real

En el Procesamiento Digital de Señales (PDS) en general existen dos formas de realizar los procesos sobre señales que son:

- **Señales almacenadas:** La señal a procesar se encuentra en algún medio de almacenamiento; donde el resultado puede o no representar una acción inmediata y la necesidad de una respuesta en tiempo real no es prioridad.

- **Señales adquiridas en tiempo real:** estas señales son adquiridas y procesadas en un lapso de tiempo específico, el procesamiento en tiempo real demanda en forma estricta el diseño de software y hardware del PDS para completar la tarea predefinida en un lapso de tiempo que satisfaga la necesidad requerida [15].

Un sistema básico de PDS en tiempo real se muestra en la figura 6.3, donde la señal analógica del mundo real se acondiciona y pasa por un filtro anti-aliasing, para eliminar señales no deseadas (como ruido), posteriormente es convertido a señal digital por medio del módulo Convertidor Analógico Digital (ADC), después es procesado por el hardware del DSP en forma digital y de nuevo se convierte en forma analógica por medio del Convertidor Digital Analógico (DAC), pasando por último por un filtro de reconstrucción y acondicionando la respuesta para el mundo real.

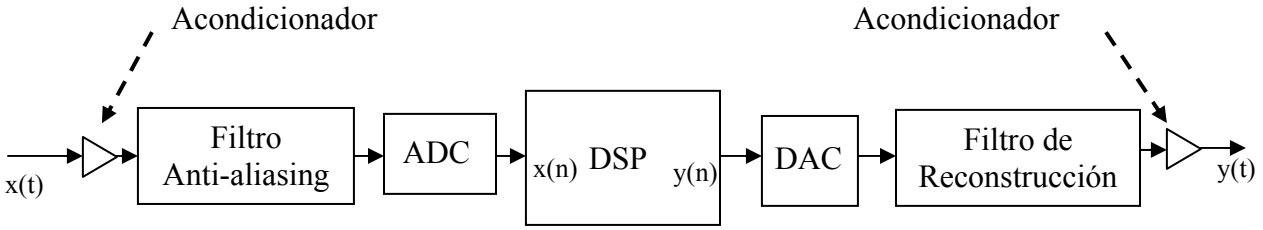



Figura 6.3. Sistema básico de PDS en tiempo real.

## 6.2.1 Acondicionamiento de señal y muestreo

Generalmente los sensores electrónicos convierten la presión, temperatura, o sonido en señales eléctricas; sin embargo, estas señales no se envían de forma directa al convertidor analógico digital, ya que los niveles de operación por lo general son muy diferentes, es decir, mientras que los sensores pueden operar en un intervalo de milivolts, los ADC van en intervalos de volts, por lo que es necesario tener una etapa de amplificación y también de acoplamiento de impedancias, lo mismo sucede con la salida.

La mayoría de las señales discretas son el resultado del muestreo de las señales continuas, como son señales de voz y audio o algún otro tipo de señal eléctrica. El proceso de convenir estas señales en una forma digital se le llama Conversión Analógico Digital (ADC) y el proceso que se encarga de reconstruir la señal continua a partir de sus muestras se le llama Conversión Digital Analógico (DAC) [25].

Típicamente el muestreo se desarrolla de forma periódica donde $\mathbf{T_s}$ es el período de muestreo, y $\mathbf{f_s = 1/ T_s}$ es la frecuencia de muestreo dando las muestras por segundo [25].

Una forma de realizar el proceso de muestreo, es multiplicando una señal continua por una secuencia de impulsos periódicos como muestra la ecuación (6.1).

$$\mathbf{x_s(t) = x_a(t)s_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT_s)\delta(t - nT_s)} \qquad (6.1)$$

Si **x(t)** es una señal con un ancho de banda limitada (BW: Band Width) con $\mathbf{X(\omega)} = 0$ para $|\omega| > \omega_M$. Entonces **x(t)** es determinado por sus muestras **x(nT)**, $\mathbf{n} = 0, \pm1, \pm2, \ldots$ si

$$\omega_s > 2\omega_M$$

donde

$\omega_s = 2\pi/T_s$

$\omega_s$ : frecuencia de Nyquist

$\omega_M$: Ancho de banda de la señal

Dadas las muestras, **x(t)** se puede reconstruir generando un tren de impulsos periódicos ponderados por el valor de las muestras correspondientes. Este tren de impulsos es procesado a través de un filtro pasa bajas ideal con una ganancia de **T** y una frecuencia de corte mucho mayor a $\omega_M$ y menor a $(\omega_s - \omega_M)$. El resultado es la señal de salida será exactamente igual a **x(t)** [26]. En la figura 6.4 se muestra de forma gráfica el espectro de una señal discreta, donde por el teorema del muestreo se evita el traslape espectral.

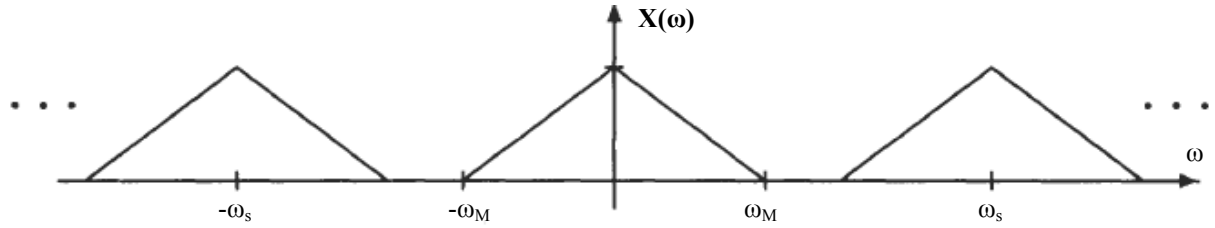



Figura 6.4. Forma gráfica del espectro del muestreo para $\omega_s > 2\omega_M$

Para nuestro caso la señal tiene un ancho de banda de voz que se considera de 4000Hz por lo que se muestrea a una frecuencia de Nyquist de 8000Hz .

## 6.2.2 Opciones de hardware

Como se muestra en la figura 6.3, **x(t)** representa una variable física real obtenida por medio de algún transductor en forma eléctrica, esta variable, por lo regular, necesita de un *Acondicionador* que es el componente que permite obtener una señal en los niveles y acoplamiento adecuados para las siguientes etapas. El filtro Anti-aliasing nos permite eliminar las frecuencias no deseadas de la señal de entrada con ello la señal se pasa por un ADC que genera una señal discreta **x(n)**, ésta se introduce al bloque donde se encuentra el hardware del DSP teniendo como salida **y(n)**, donde posteriormente pasa por el bloque DAC reconstruyendo y adecuando la salida **y(t)**. A pesar de que es posible implementar algoritmos de PDS en cualquier computadora digital, el desempeño (tiempo de proceso) lo determina la plataforma de hardware a utilizar. Existen seis plataformas de hardware para PDS que son ampliamente usadas:

1. Microprocesadores y microcontroladores de propósito general (GPP).

2. Procesador digital de señales (DSP chips).

3. Módulos digitales (DBB) como multiplicadores y sumadores.

4. Dispositivo Lógico Programable Complejo (CPLD)

5. Dispositivo Programable de Arreglo de Compuetas (FPGA)

6. Dispositivos dedicados como son circuitos integrados de aplicación específica (ASIC).

En la tabla 6.1 se muestran las características de las plataformas mencionadas.

| | **ASIC** | **FPGA** | **CPLD** | **DBB** | **GPP** | **DSP chips** |
|---|---|---|---|---|---|---|
| Cantidad chip | 1 | 1 | >1 | >1 | 1 | 1 |
| Flexibilidad | Ninguna | Programable | Programable | Limitada | Programable | Programable |
| Tiempo de diseño | Largo | Largo | Medio | Medio | Corto | Corto |
| Consumo potencia | Baja | Baja | Media | Media-Alta | Media | Baja-Media |
| Velocidad procesamiento | Alta | Alta | Alta | Alta | Baja-Media | Alta |
| Confiabilidad | Alta | Alta | Media | Baja-Media | Alta | Alta |
| Costo de desarrollo | Alta | Alta | Media | Media | Baja | Baja |
| Costo de producción | Baja | Baja-Media | Alta | Alta | Baja-Media | Baja-Media |

Tabla 6.1 Comparación de plataformas.

En relación con la tabla 6.1 podemos observar que la utilización de un DSP para sistemas PDS resultan ser una buena elección dado que tiene todas las ventajas para realizar el desarrollo e implementación de un sistema en tiempo real.

### *6.2.3 Dispositivos de punto fijo y punto flotante*

Una diferencia básica entre las arquitecturas de DSP, es si el chip es de aritmética de punto fijo o de punto flotante, un procesador de punto fijo típico de 16 bits, como la familia de DSPs TMS320C5xxx de Texas Instruments (TI), almacena los números en un formato entero donde los coeficientes y las muestras de la señal son guardados en una precisión de 16 bits, los valores de operaciones intermedias pueden ser guardados en una precisión de 32 y 40 bits contenido en los acumuladores internos con la idea de reducir el error acumulativo. Los dispositivos DSP de punto fijo son usualmente más baratos, más rápidos y de menor consumo de potencia que los de punto flotante ya que tienen menos bloques internamente y menos pines externos [15].

Un procesador de punto flotante típico de 32 bits, como el TMS320C3x de TI, almacena 24 bits de mantisa y 8 bits de exponente. El formato de 32 bits de punto flotante da un intervalo dinámico muy amplio, sin embargo, la resolución continúa siendo de sólo 24 bits. Las limitaciones del intervalo dinámico pueden ser virtualmente ignoradas cuando se diseña con un dispositivo DSP de punto flotante, esto contrasta en un diseño de punto fijo, donde el diseño debe contemplar factores de escala para prevenir desbordamientos, lo cual dificulta en gran medida el desarrollo, además de que utiliza mayor tiempo de procesamiento. Los dispositivos de punto flotante se pueden necesitar en aplicaciones donde los coeficientes son variables con el

tiempo, las señales y los coeficientes están en un intervalo dinámico muy amplio o donde se necesitan estructuras muy grandes de memoria como el procesamiento de imágenes. Otro caso donde los dispositivos de punto flotante pueden ser justificados es cuando el costo de desarrollo es alto y se quiere bajos volúmenes de producción. Estos dispositivos son eficientes para usarlos con compiladores "C" de alto nivel y reducen la necesidad de identificar el intervalo dinámico del sistema [15].

## 6.3 DSP TMS320C5402

El DSP que se utiliza para el desarrollo de este proyecto es el TMS320C5402, que pertenece a la familia C54x de Texas Instruments, es un procesador digital de señales de punto fijo diseñado para aplicaciones en tiempo real. Está construido bajo una arquitectura tipo Harvard modificada con alto grado de paralelismo y bajo consumo de potencia con modos de direccionamiento versátiles y un conjunto de instrucciones que mejoran su desempeño [6],[29].

### *6.3.1 Características generales*

A continuación se listan las características generales más importantes del DSP TMS320C5402.

- Arquitectura de multibus avanzado con tres buses separados para memoria de datos (16 bits) y un bus para memoria programa.
- Unidad Aritmética Lógica (ALU) de 40 bits, incluyendo dos acumuladores independientes de 40 bits y un bloque de corrimiento de 40 bits.
- Multiplicador en paralelo de 17x17 bits acoplado a un Sumador dedicado de 40 bits para la operación de Multiplicación – Acumulación (MAC) en un sólo ciclo de instrucción.
- Unidad de Comparación, Selección y Almacenamiento (CSSU) para el algoritmo Viterbi.
- Codificador de Exponente para el cálculo del exponente de un valor de 40 bits en el acumulador en un solo ciclo.
- Dos generadores de dirección con ocho registros auxiliares y dos unidades aritméticas de registros auxiliares (ARAU).
- Buses de datos con características de retención ("Holding").
- Modo extendido de direccionamiento para direccionar hasta 1M x 16 bits.
- 4K x 16 bits en ROM interna.
- 16K x 16 bits en DARAM.
- Instrucciones:

- Para operaciones de repetición y bloques de repetición de código programa.
- Para el movimiento de bloques de memoria y para un manejo eficiente de datos y programa.
- Para operaciones con palabras de 32 bits.
- Para lectura de dos o tres operandos.
- Aritméticas con almacenamiento y carga en paralelo.
- Almacenamiento condicional.
- De retornos rápidos desde llamadas a interrupciones.

- Periféricos internos:

  - Generador de estados de espera programados por software y un banco de interrupción programable.
  - Generador de reloj con oscilador interno o reloj externo.
  - Circuito de malla de fase amarrada (PLL) programada por software.
  - 2 puertos seriales bufereados multicanal (McBSPs).
  - Interfaz de puerto paralelo de tipo Huésped de 8 bits mejorado (HPI8).
  - 2 Temporizadores de 16 bits con preescalador de 4 bits.
  - Controlador de 6 canales de acceso directo a memoria (DMA).

- Control de bajo consumo de potencia con instrucciones IDLE1, IDLE2 e IDLE3.

- Emula el estándar 1149.1 (JTAG) de IEEE.

- Velocidad a 100Mhz (100MIPS) y periodo de ciclo de reloj de 10ns.

- Maneja seis niveles de pipeline: pre-búsqueda, búsqueda, decodificación, acceso, lectura y ejecución.

- Voltaje de operación a 3.3/1.8 V.

### 6.3.2 Sistema de Desarrollo TMS320C5402 DSK

El sistema TMS320C5402 DSK es una tarjeta Texas Instruments de desarrollo de bajo costo que permite evaluar y desarrollar aplicaciones para el C54x DSP y también como referencia de diseño de hardware.

El sistema incluye:

- Un DSP TMS320C5402 a 100 MHz.

- 64K palabras de 16 bits en memoria externa SRAM.

- 256K palabras de 16 bits en memoria FLASH.
- Interfaz JTAG de prueba para la emulación del controlador de bus y un puerto host conectado a un PC anfitrión del puerto paralelo.
- Interfaz telefónica (DAA).
- Interfaz para audio y micrófono.
- Interfaz RS-232 asíncrono.
- Conector  para expansión de tarjetas externas.

La tarjeta DSK es capaz de trabajar de forma independiente y requiere de 5 Vdc, dados por una fuente externa, de forma interna, cuenta con reguladores lineales que proveen de 1.8 Vdc, que alimentan al núcleo del chip DSP y de 3.3 VCD para la operación digital y 5 Vdc para los voltajes analógicos.

El diagrama a bloques del la tarjeta TMS320C5402 DSK se muestra en la figura 6.5.

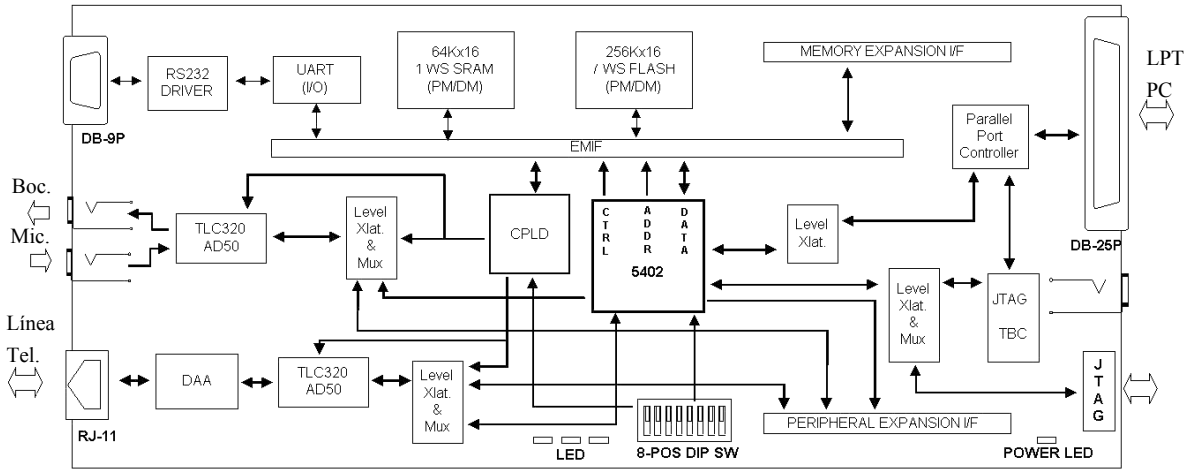



Figura 6.5. Sistema de desarrollo TMS320C5402 DSK.

Esta tarjeta cuenta con el conector RJ-11 para la conexión directa con el cable telefónico, con la interfaz DAA CPC5604, que se encarga de manejar con aislamiento óptico los niveles del TIP y RING a -48 volts. En la figura 6.6 se muestra su diagrama típico de conexión de la interfaz DAA.

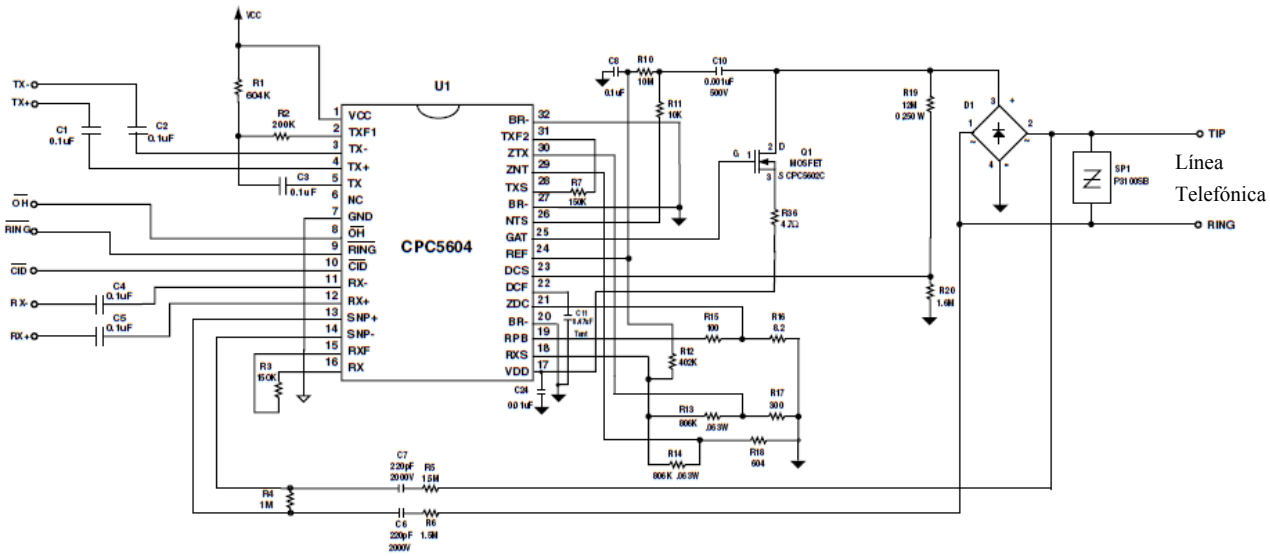



Figura 6.6. Arreglo de acceso a datos óptico (DAA).

Con el convertidor Analógico Digital (ADC) y Digital Analógico (DAC) del TLC320AD50, se tienen cubiertas dos funciones importantes para la comunicación telefónica al conectar la señal de entrada analógica a la parte digital y de la salida digital pasarlo a analógica. La tasa de conversión es programable tanto para el ADC como el DAC con una tasa máxima de 22.05KHz [21]. En la figura 6.7 se muestra la conexión del TLC320AD50. Este códec está conectado directamente al puerto serie buffereado multicanal (McBSP) cero del DSP.

La línea telefónica se conecta al circuito CPC5604, donde la señal que va desde y hacia la línea de teléfono se lleva a cabo en las conexiones de “Ring” y “Tip”, la señal se recibe a través de dos cables de la línea telefónica, la cual es convertida en luz infrarroja y recibida por un fotodiodo. La intensidad de la luz infrarroja es modulada dependiendo de la señal recibida y ésta es transferida a través de un aislamiento óptico al fotodiodo generando una corriente eléctrica que es una representación lineal de la señal. Esta corriente posteriormente se amplifica y es convertida a un voltaje diferencial representada por RX+ y RX-, los cuales son conectados al ADC. De igual manera la transmisión de una señal a la línea telefónica es realizada por una entrada diferencial dada por TX+ y TX- cuyo máximo voltaje no debe exceder de 2.18Vpp, la cual es convertida en una sola señal, y de forma similar está acoplado al transmisor con medio de un fotodiodo.

Otras dos funciones que nos da el CPC5604, es la detección del ring y la presencia del identificador (Caller ID) por medio de CID, las cuales puede ser sensadas por medio de banderas proporcionadas por el API de CCS.

El CPC5604 se encuentra conectado de forma directa al códec TLC320AD50, quien se encarga de realizar las conversiones ADC y DAC. El TLC320AD50 se encuentra conectado por medio de un puerto serial bufereado Multicanal (McBSP) al TMS320C5402.

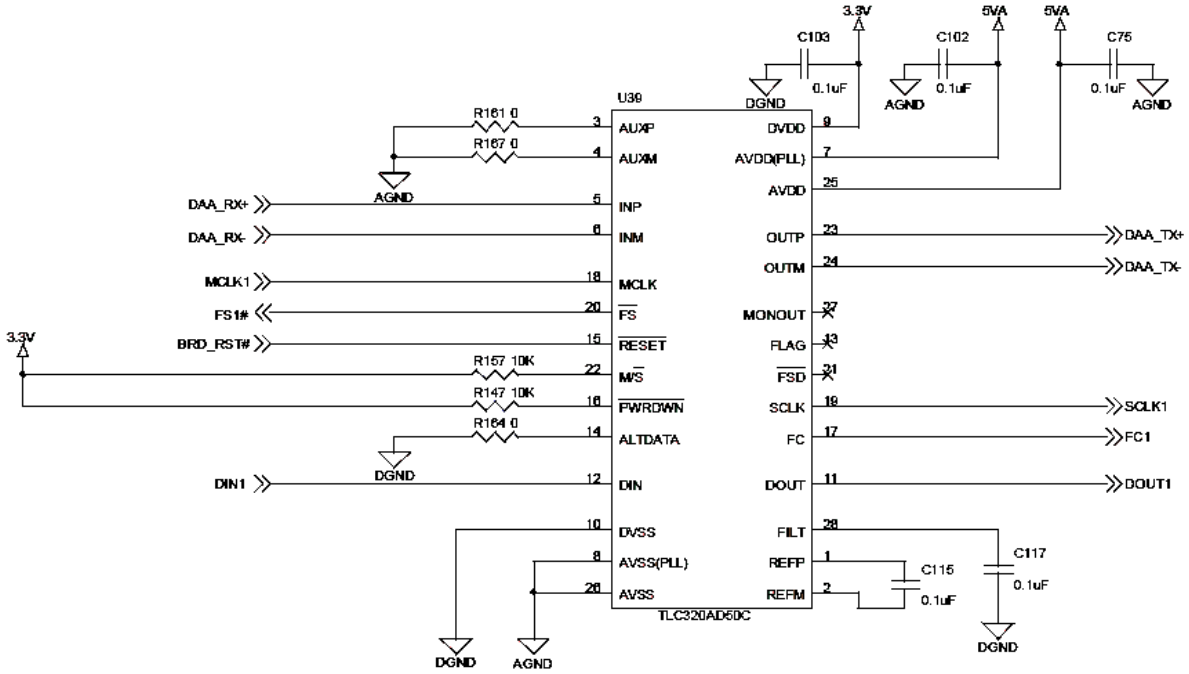



Figura 6.7. Conexión del TLC320AD50.

EL DSP se conecta tanto al DAA como al AD50 para su control y flujo de datos, estas líneas de control se manejan posteriormente por medio de software que se especificará a continuación.

# 6.4 Software

El software es una parte esencial del proyecto ya que se encarga del control de los diferentes periféricos y de realizar el procesamiento digital de las señales utilizando el DSP.

En el proyecto el software desarrollado se divide en dos partes:

- Análisis Fuera de Línea: es el que se encarga de calcular los parámetros necesarios a partir de las muestras de voz real y que serán alimentados al sintetizador de voz para su reproducción en tiempo real.

- Procesamiento en Línea o Tiempo Real: es el que se encarga de decodificar la señal DTMF y de generar la voz sintética.

## *6.4.1 Análisis fuera de línea*

En el procesamiento de fuera de línea, se realiza el análisis para el cálculo de los parámetros necesarios para posteriormente reproducir la voz sintética. Las herramientas utilizadas para este propósito son:

- Cool Edit. Es un programa de adquisición y edición de audio.

- Visual Studio de Microsoft versión 2008 programando en el lenguaje de C#. Este lenguaje es orientado a objetos para la plataforma .NET en un ambiente Windows.

- MatLab que facilita el desarrollo de programas con diagnóstico de errores y seguimiento de código, un eficiente manejo de matrices que implementa funciones intrínsecas y además un ambiente gráfico para facilitar el análisis.

Para el análisis fuera de línea se tienen las siguientes consideraciones:

1. Contar con la muestra de voz en un archivo en forma de vector columna con las muestras correspondientes en formato entero el cual lo llamaremos archivo.dat.

2. Esta muestra de voz es editada para eliminar los silencios, al inicio y fin donde el algoritmo de detección de silencio sólo es necesario para los pequeños silencios presentes dentro de la palabra.

3. Con base a la teoría, tener definido el tamaño de ventana y el número de coeficientes a calcular.

4. Como resultado se generan los archivos .h que corresponden a los diferentes parámetros necesarios para su reproductor en el sintetizador.

Los parámetros que se obtienen como resultado corresponden a:

- ***Coeficientes de Predicción Lineal***. Son los coeficientes utilizados por el algoritmo del sintetizador de voz (ver sección 3.4).

- ***Pitch***. Si se trata de un sonido *voceado*, el pitch representa la frecuencia fundamental, la cual se reproducirá en el algoritmo del sintetizador por medio de un tren de impulsos. En el caso en que el pitch no exista (sea igual a cero) entonces se tratará de un sonido *no voceado*, lo que implica que en lugar de la frecuencia fundamental el sintetizador será alimentado con ruido blanco como se puede observar en la figura 3.7 (ver sección 2.5).

- ***Ganancia de segmento***. La ganancia del segmento corresponde a la raíz cuadrada del elemento $\mathbf{R_m(0)}$ de la matriz de autocorrelación.

En la figura 6.8 se muestra el diagrama del proceso que consiste en cargar el archivo.dat de la muestra de voz a procesar en memoria, dado que las muestras pueden no estar normalizada se pasan por un algoritmo de normalización, posteriormente por un filtro de preénfasis, se divide en segmentos y se ventanea por medio de una ventana *Hamming*. Para cada ventana se obtiene el pitch, los coeficientes *LPC* y la energía de cada segmento que corresponde al elemento $\mathbf{R_m(0)}$ de la matriz de autocorrelación. Cada resultado es guardado en un archivo con el formato y sintaxis necesaria con extensión .h listo para el compilador "C" de Code Composer.

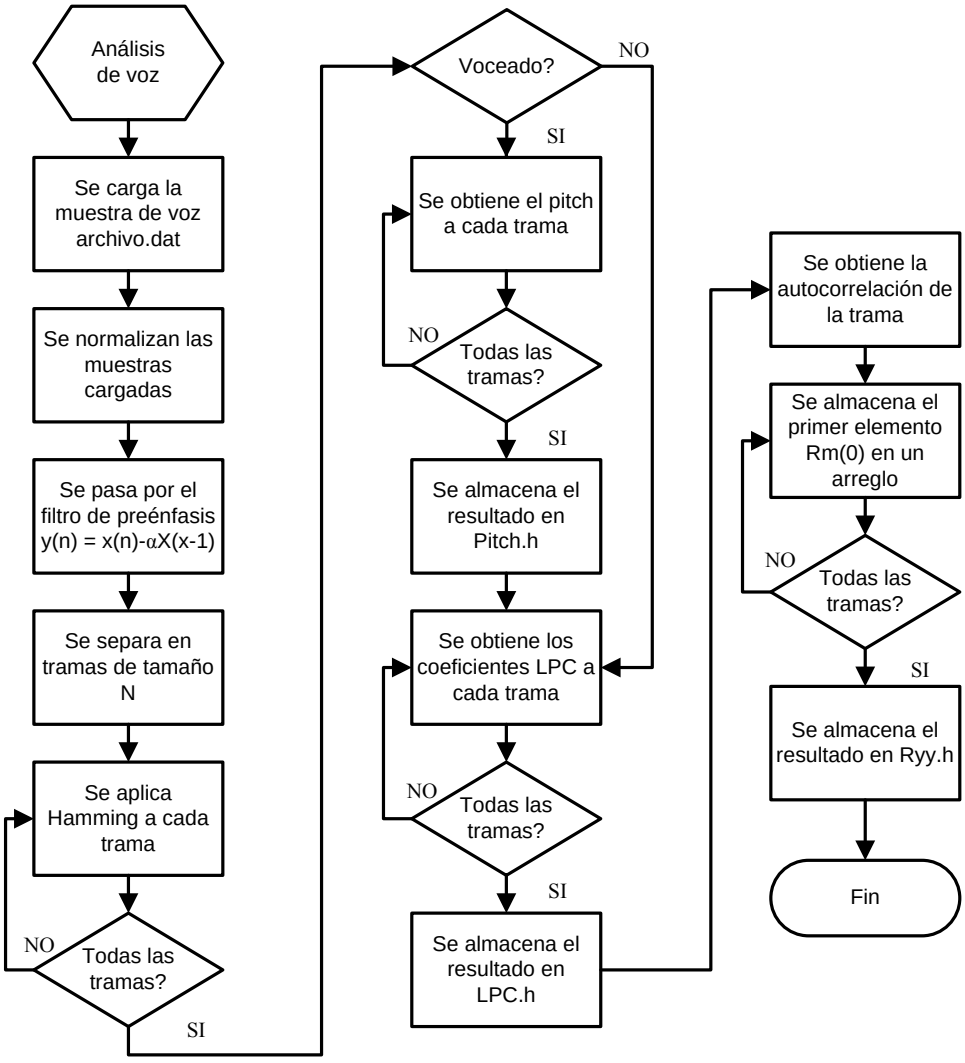



Figura 6.8. Proceso análisis de voz fuera de línea.

**Muestra de voz**

La muestra de voz se captura por medio de un micrófono conectado a la computadora como muestra la figura 6.9. La voz se graba con una resolución de 16 bits monoaural a 8000 muestras por segundo. En la figura 6.10 a) se presenta un ejemplo de la forma de onda de la palabra "Hola" y en 6.10 b) se muestra su espectro.

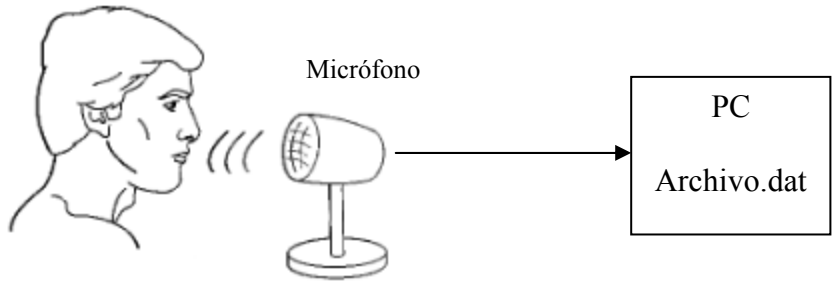
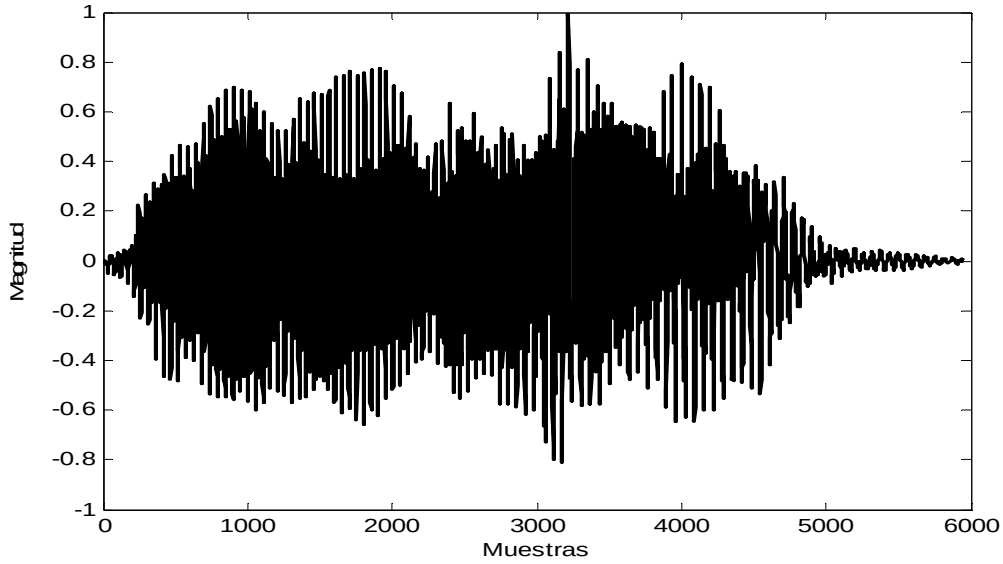
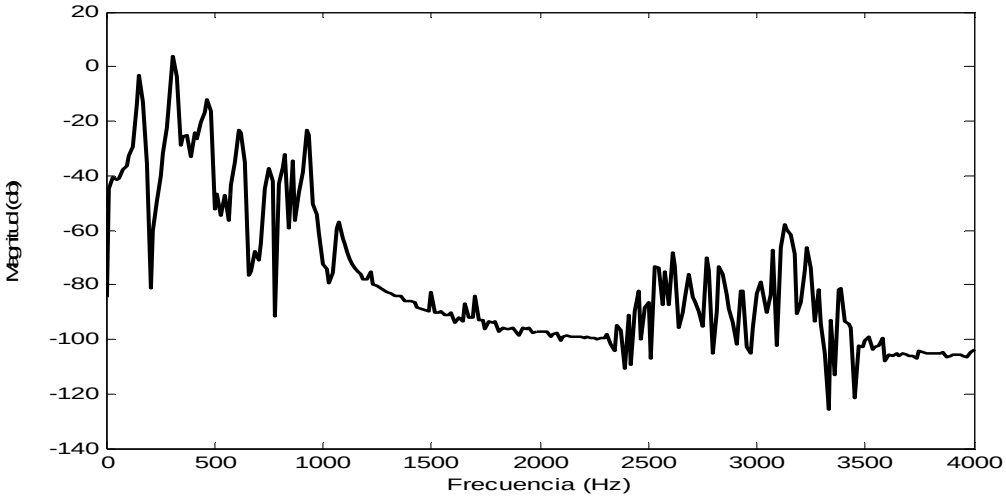



Figura 6.9. Configuración de grabación de voz original.




a)




b)

Figura 6.10. a) Forma de onda de la palabra “Hola”. b) Espectro

**Pre-énfasis**

La aplicación del filtro de pre-énfasis a la muestra de voz, nos permite acentuar las componentes de alta frecuencia ya que se comporta como un filtro paso-altas con una frecuencia de corte de 3800Hz, en la figura 6.11 se muestra el espectro de la palabra "Hola" después del pre-énfasis utilizando la ecuación (6.2). Dado que el método de predicción lineal tiene menor rendimiento a frecuencias altas es conveniente acentuarlas para tener mejores resultado, por lo que el filtro de pre-énfasis es conveniente (ver sección 2.3.3).

$$s'(n) = s(n) - 0.9375\, s(n-1) \tag{6.2}$$

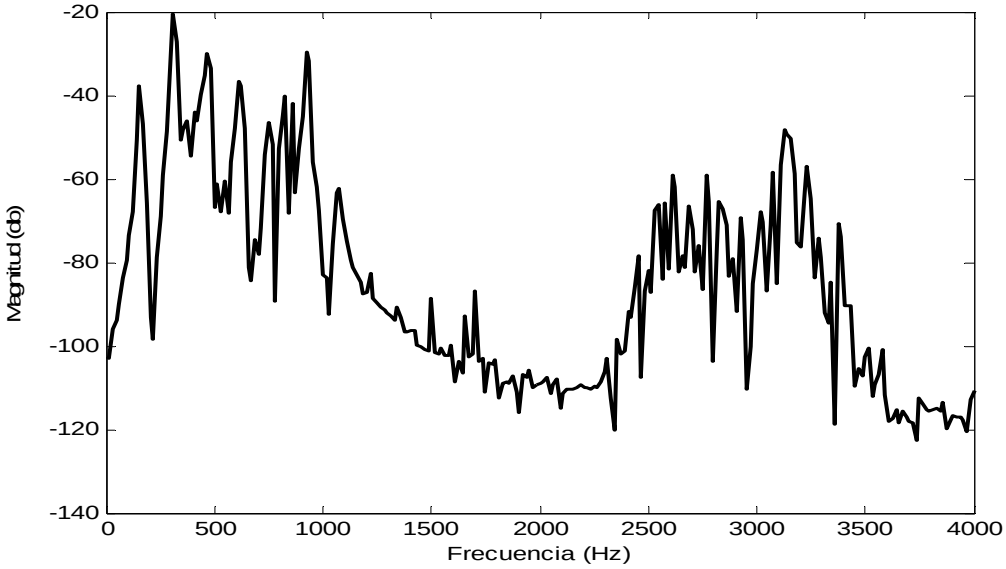



Figura 6.11. Filtro de pre-énfasis aplicado.

**Ventaneo**

El ventaneo se aplica a la señal para seccionarlo en segmentos de 400 muestras que corresponden a 50ms donde se obtuvo una mejor calidad de audio sintetizado. La ventana más comúnmente utilizada es la de *Hamming* que se muestra en la figura 6.12 y es aplicada a cada uno de los segmentos. Esta ventana presenta en su respuesta en frecuencia lóbulos laterales menores, el primer lóbulo se encuentra a 41 dB por debajo del lóbulo principal con pendiente de -6 dB por octava, pero el lóbulo principal es por lo menos dos veces más ancho que el de la ventana cuadrada. Tiene menos oscilaciones en las regiones pasa banda y rechazo de banda de $\mathbf{H_\omega(\omega)}$, su banda de transición se vuelve menos abrupta y más ancha [11] (ver sección 2.3.6).

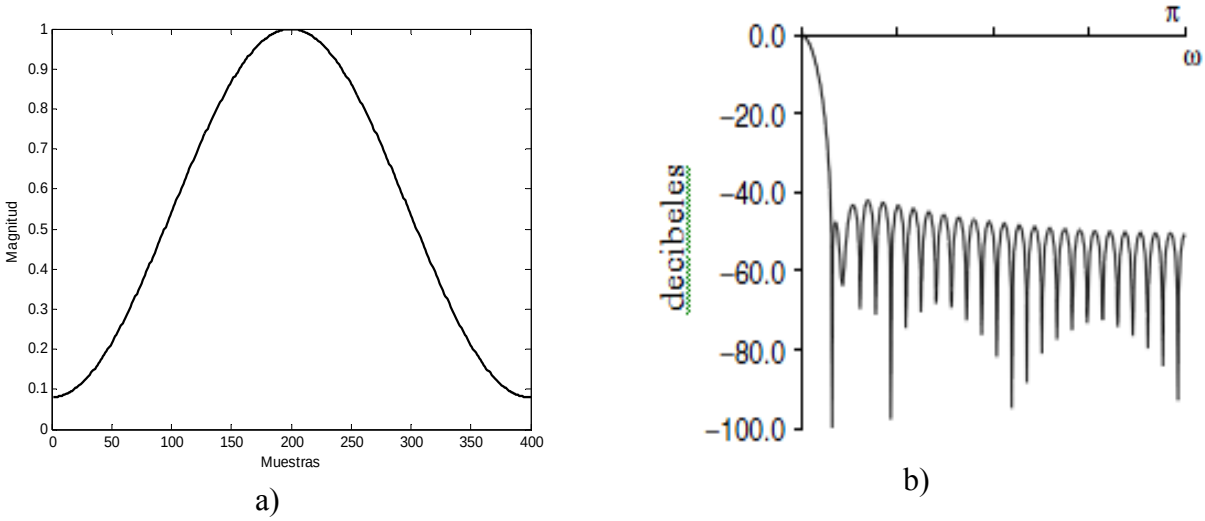



Figura 6.12. Ventana de Hamming a) Tiempo, b) Espectro.

La muestra de voz se organiza en bloques de tamaño *N=400* que representa la longitud del segmento. En la figura 6.13 se muestra un ejemplo de la organización de la muestra de voz.

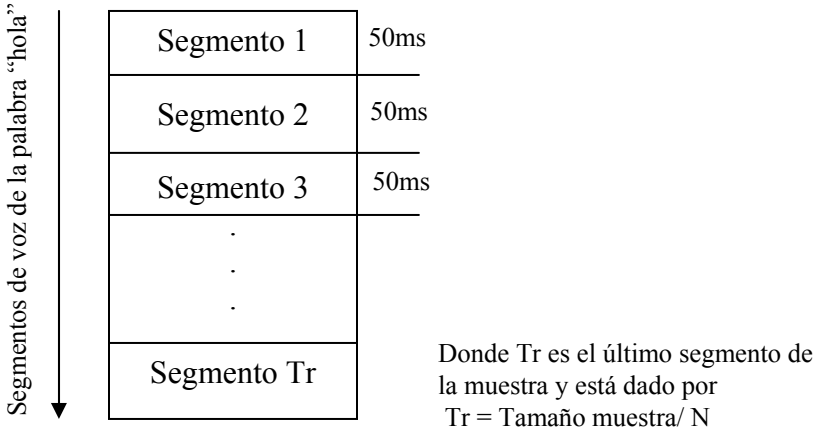



Figura 6.13. Organización de los segmentos de voz.

**Pitch**

Se calcula el Pitch de cada segmento por el método de recorte central expuesto en la sección 2.5.2. Al aplicar el algoritmo para extraer el pitch se obtiene un número entero, el cual si es cero nos indica que se trata de un segmento *no voceado* mientras, que si es mayor a cero se tratará de un segmento *voceado*, con ello se da también el pitch en mismo número. En la figura 6.14 a) se da un ejemplo de segmento *voceado*, mientras que en 6.14 b) corresponde a uno *no voceado*.

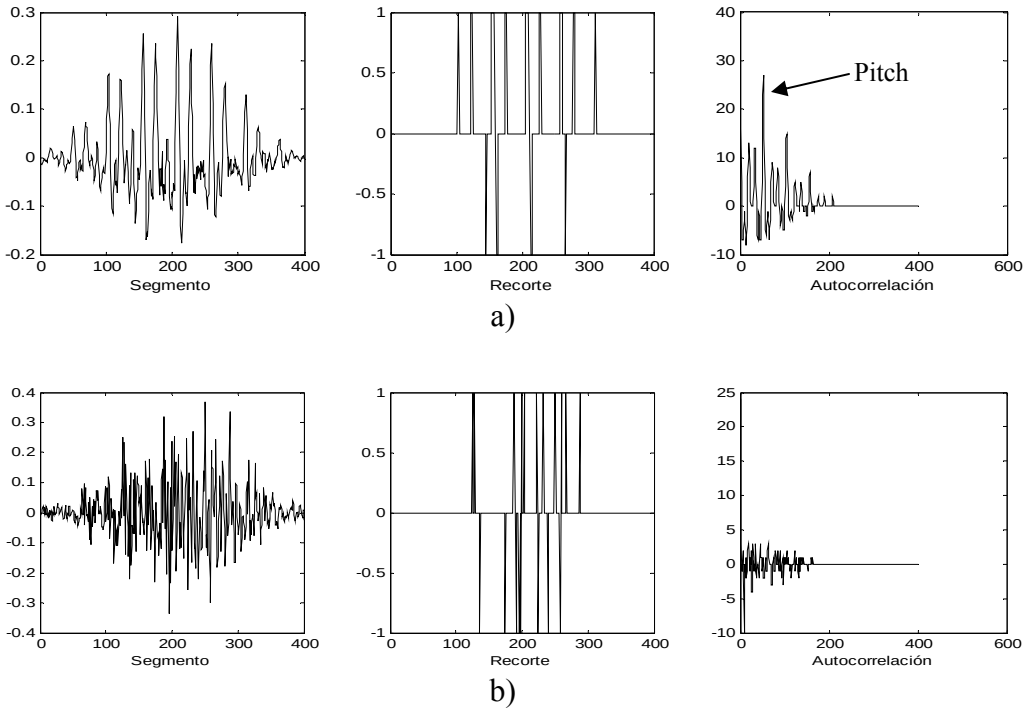



Figura 6.14. a) Segmento voceado, b) Segmento no voceado

En la figura 6.14 a) se puede observar que el segmento es *voceado* dado que presenta un comportamiento periódico, acentuándose más en su autocorrelación, dado que si la señal es periódica su autocorrelación también lo será, el cual es la base para calcular el pitch. En la figura 6.14 b) se puede observar que el comportamiento periódico tanto de la señal como de la autocorrelación desaparece por lo que se trata de un segmento *no voceado*. La información obtenida del Pitch se almacena en un archivo Pitch.h.

**Parámetros LPC**

El orden de predicción lineal nos proporciona el número de formantes a los cuales se puede aproximar a la señal original, entre mayor sea el orden la aproximación de los formantes también será más cercana. En la figura 6.15 se muestra el espectro de la señal original también se presenta la aproximación de los formantes con diferentes valores del orden de predicción.

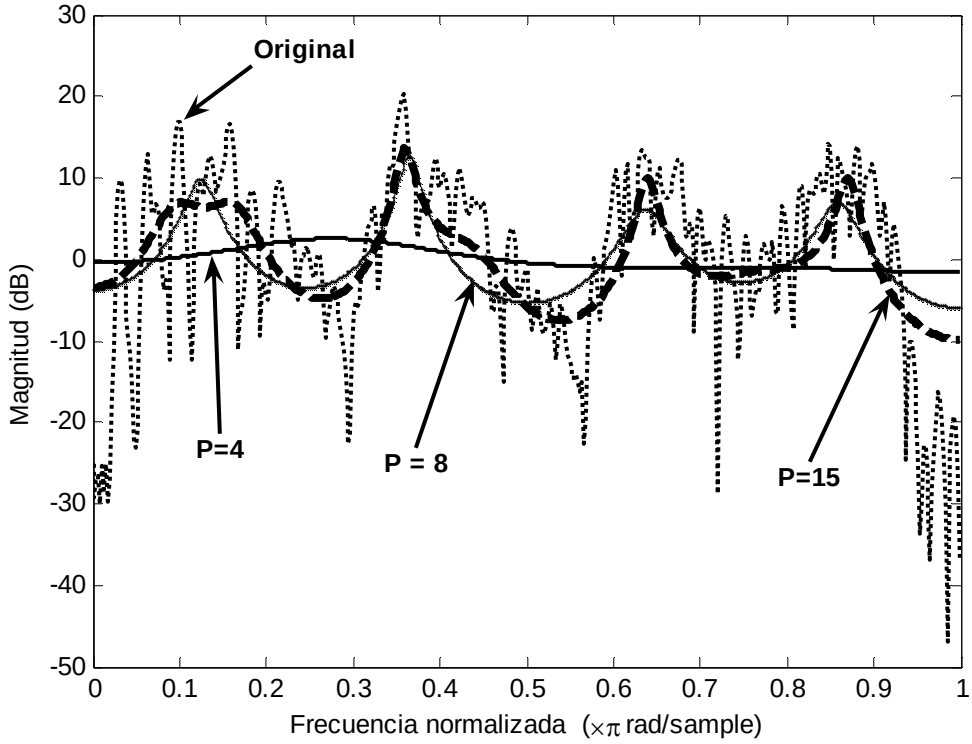



Figura 6.15. Espectro de señal original con las aproximaciones de formantes a diferentes órdenes de predicción

El orden de predicción corresponde al número de polos del filtro de predicción, el orden determina el número de formantes del espectro de la señal a reconstruir, dado que son polos conjugados, para un orden de predicción ocho se deben tener cuatro formantes, para uno de orden diez se tendrían cinco formantes y así sucesivamente. En la práctica el orden de predicción para la síntesis de voz está en un intervalo de 12 a 15.

**Ganancia**

La ganancia del segmento se recupera de la autocorrelación realizada para el cálculo de los parámetros LPC almacenando los valores de **Rm(0)** que corresponden al primer elemento de la matriz de autocorrelación que tiene como información dicha ganancia.

## *6.4.2 Procesamiento en línea o Tiempo Real*

La tarjeta de desarrollo incluye la herramienta de software de Texas Instruments llamada Code Composer Studio (CCS), que es un ambiente integral de desarrollo (IDE) que nos permite de una manera sencilla llevar a cabo el desarrollo de un proyecto en la tarjeta DSK del TMS320C5402 como muestra la figura 6.16.

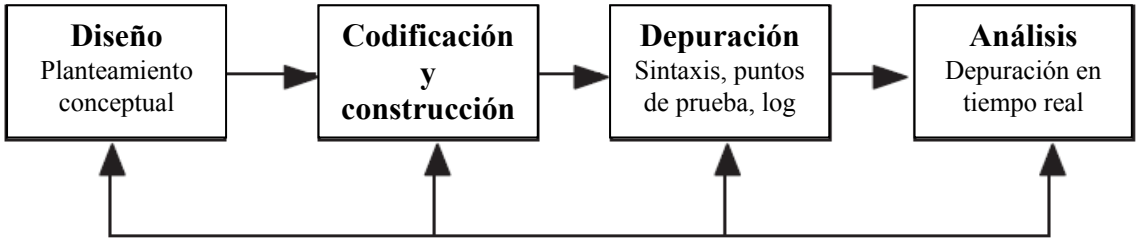



Figura 6.16. Flujo simplificado del desarrollo de un proyecto en el Code Composer Studio.

En el CCS es posible programar en dos lenguajes que va desde bajo nivel como el ensamblador algebraico o de nivel medio que es el lenguaje C, que es utilizado para este proyecto en el caso del procesamiento en tiempo real.

El sistema desarrollado en este proyecto se puede considerar un sistema incrustado (Embedded), donde una característica de programación es que se realiza a partir de un ciclo infinito. Este ciclo es necesario porque la tarea del software embebido nunca termina [22].

Otra característica del software embebido, que difiere de los requerimiento de software que corre en una PC, es que el usuario de la computadora no le interesa si el sistema tarda algunas decenas de microsegundos en responder cuando presiona una tecla, pero en el encendido en un auto controlado por un sistema embebido, debe responder en ese tiempo el disparo de chispa de las bujías, es decir, el sistema embebido trabaja en tiempo real [23]. En el diagrama a bloques del software del sistema en tiempo real en el DSP se muestra en la figura 6.17.

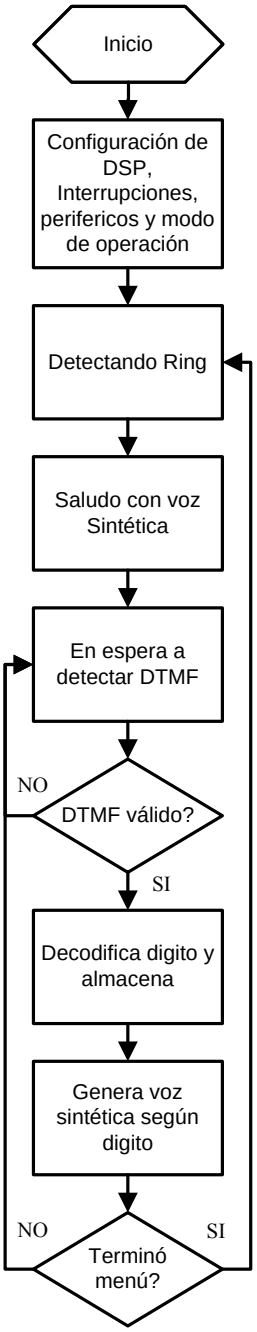



Figura 6.17. Diagrama a bloques del software desarrollado en el DSP.

**Iniciación de periféricos y detección de RING**

Para el control de los diferentes periféricos de la tarjeta de desarrollo TMS320C5402 DSK, el CCS nos ofrece instrucciones de aplicación (API) para llevar a cabo estas tareas. En la figura 6.18, se muestra la forma el comportamiento de la señal de la línea cuando se presenta el RING.

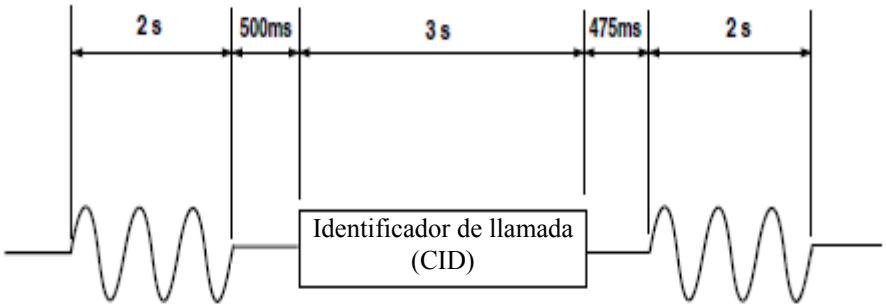



Figura 6.18. Forma de onda de la señal de RING

La señal de RING se presenta a una frecuencia de 17Hz y con un nivel de 75Volts, con dos segundos de presencia por cuatro de silencio. El DAA nos provee de un pin que indica la presencia del RING en la línea por lo que en el DSP se censa ese pin para verificar la presencia de una llamada entrante, en la figura 6.19 se muestra el diagrama a bloques con seudocódigo de esta operación.

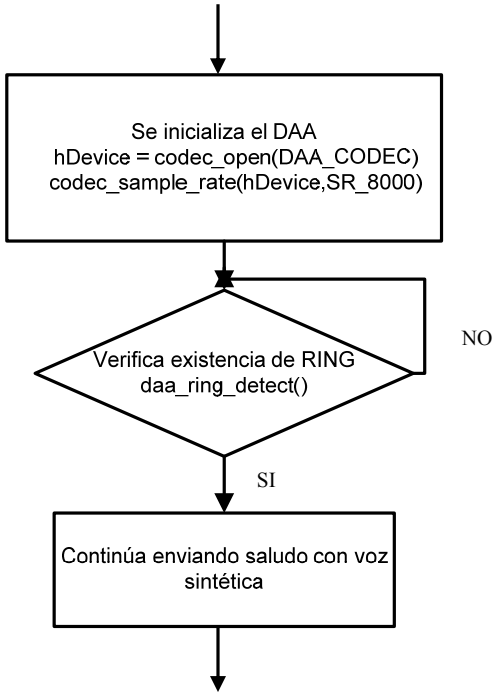



Figura 6.19. Diagrama a bloques para la detección de RING.

**Saludo voz sintética**

El saludo de voz sintética se genera en tiempo real y se envía a la línea telefónica, para ello debe de estar configurado el HANDSET, quien se encarga de la conversión digital analógica hacia el DAA y por último a la línea telefónica. En la figura 6.20 se muestra el diagrama a bloques de esta operación.

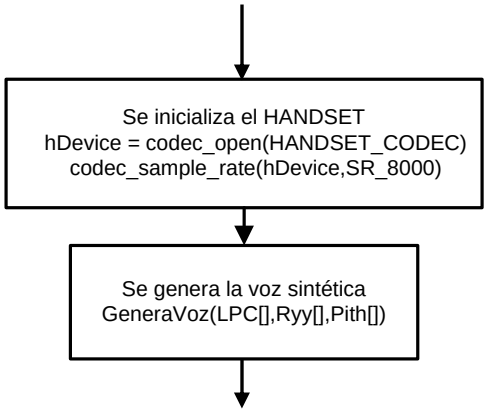



Figura 6.20. Diagrama de generación del saludo.

**Voz sintética**

La generación de la voz sintética necesita de los parámetros previamente calculados que son:

- Coeficientes LPC.
- Ganancia de los segmentos.
- Segmento voceado o no voceado.
- Pitch

En la figura 6.21 se muestra el diagrama a bloques del sintetizador, donde se puede observar a partir de los parámetros previamente adquiridos, la decisión si el segmento es *voceado* o *no voceado*; la generación de la excitación que corresponde según la decisión anterior; posteriormente, la generación de voz determinado por el filtro todo polo, con el comportamiento definido por los parámetros LPC; por último, la entrega de esta señal a la línea telefónica por medio del dispositivo DAA de la tarjeta de desarrollo. En la figura 6.22 se muestra el espectro de la voz sintética generada de la palabra "Hola" con un orden de predicción $p=15$ y un tamaño de segmento $N = 400$ muestras. En la figura 6.21 se muestra el modelo todo polo que utiliza los parámetros LPC.

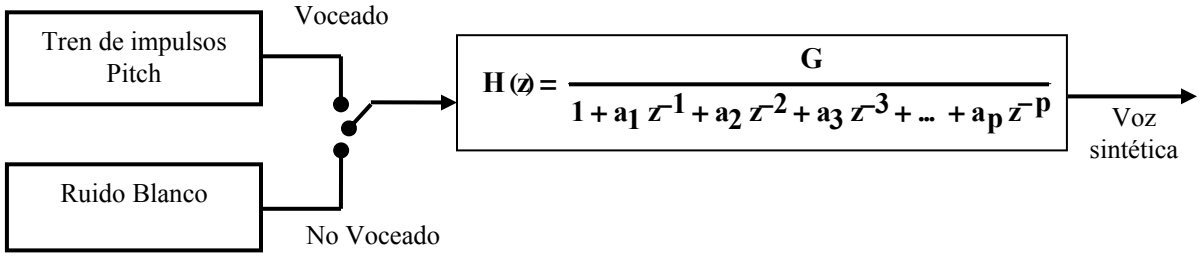



Figura 6.21. Modelo de síntesis con filtro todo polo

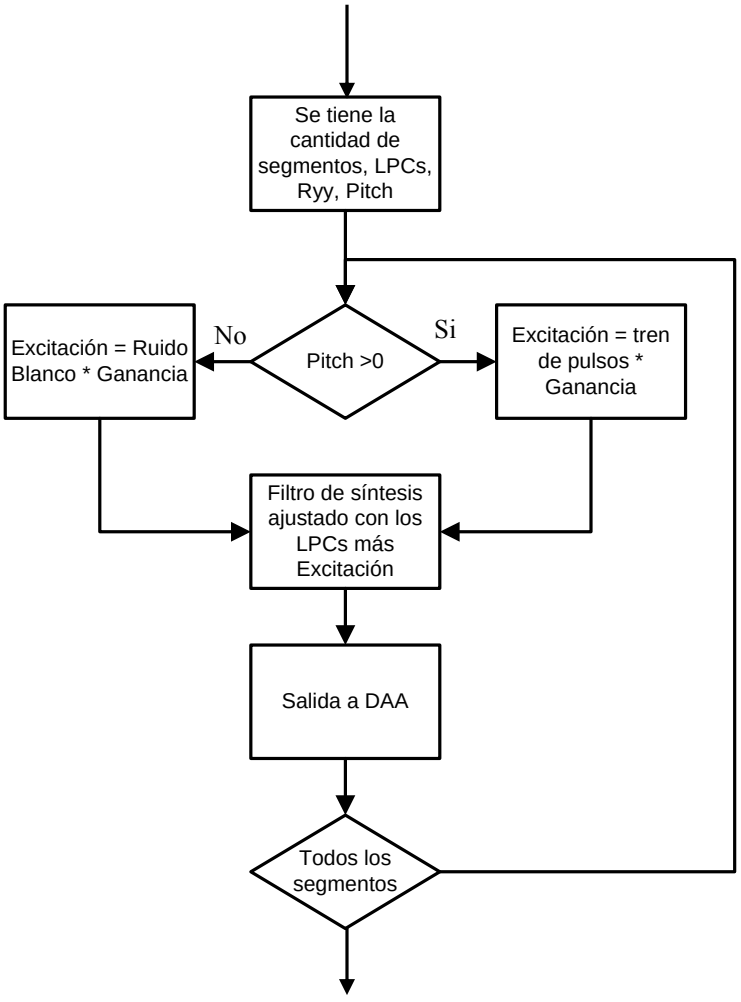



Figura 6.21. Generación de voz sintética.

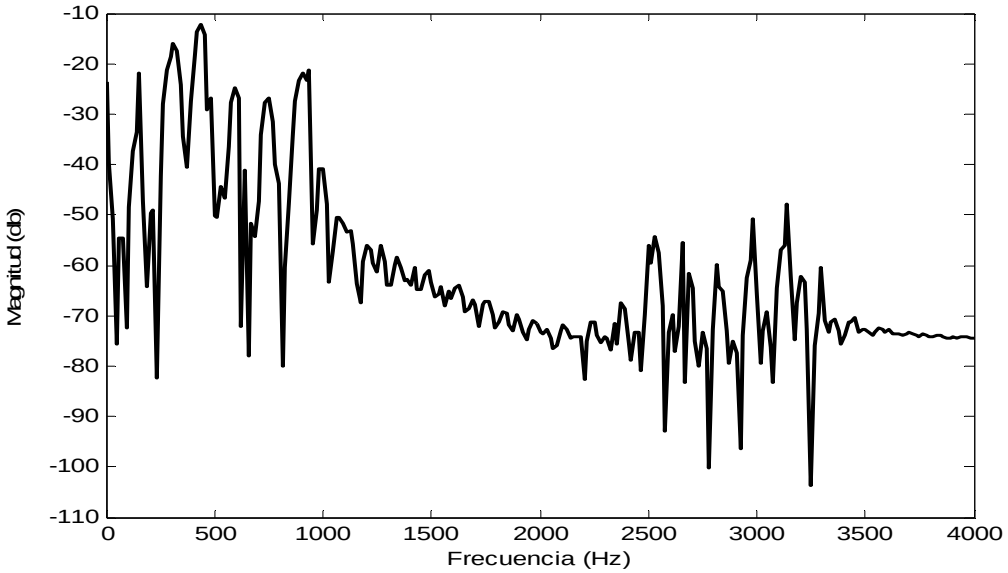



Figura 6.22. Espectro de la señal sintética de la palabra "Hola"

**Detección DTMF**

Cuando se genera la voz sintética y se envía a la línea telefónica, el sistema se pone en espera de un tono DTMF, la espera consiste en estar capturando ventanas de 205 muestras que corresponde a 25ms y probando cada una de ellas para detectar o no la presencia de señales que correspondan a un tono DTMF. En la figura 6.23 se ilustra este comportamiento (ver capitulo 5).

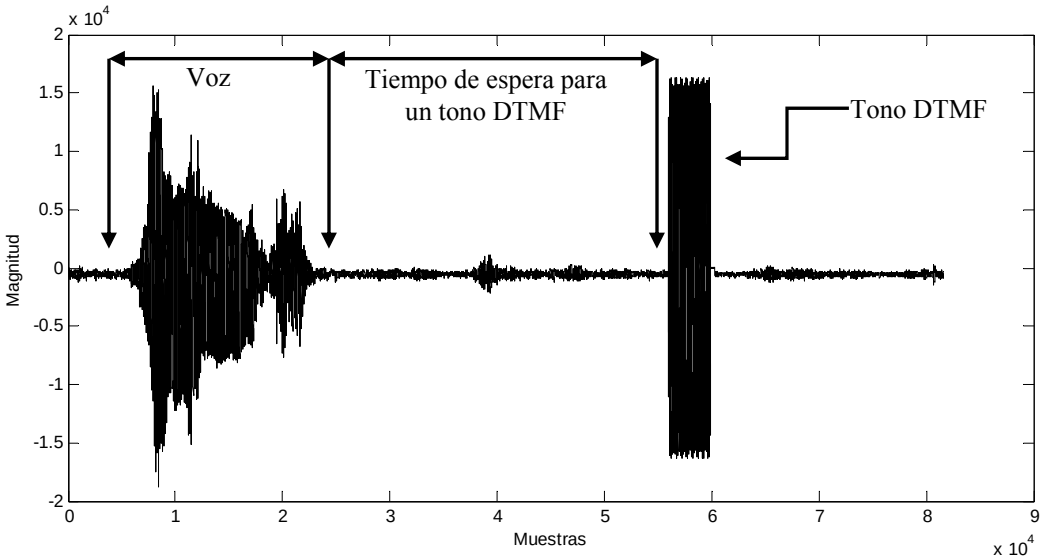



Figura 6.23 Comportamiento de la señal en presencia de un tono DTMF

## 6.5 Resumen

En los sistemas de tipo embebido es necesario un equilibrio tanto en software como en hardware, en este caso el hardware simplifica en gran medida la complejidad por lo que el software hace uso de interfaz de programación de aplicaciones (API) enfocándose en mayor medida en la implementación de algoritmos específicos de la aplicación.

La utilización de Visual Studio 2008 gracias a su simplicidad y su interfaz intuitiva, así como también para el Code Composer Studio, resulta que el ciclo de desarrollo y pruebas sea de una manera ágil y hasta cierto punto simple, ya que muchas de las tareas se realizan en menor tiempo.

# CAPITULO SIETE

# 7. Resultados

En este capítulo se analizan los diferentes factores que pueden influir en la calidad de la voz y en el rendimiento que se logra del sintetizador, variando parámetros como son: la cantidad de coeficientes LPC y el tamaño de la segmentación; se revisa la carga de procesamiento que se presenta en el DPS, la utilización de sus recursos presentes en la tarjeta de desarrollo; Así como también, se revisa el comportamiento del sistema en la decodificación DTMF con diferentes niveles de ruido y con ello determinar su inmunidad, de igual manera se verifica su comportamiento con diferentes tamaños de ventana para procesar.

Para evaluar el desempeño del sistema y los resultados obtenidos, el análisis se ha dividido en tres partes:

- **Síntesis de voz**. Generación de voz a partir de parámetros previamente calculados.
- **Decodificación DTMF**. Identificación del dígito presionado por un usuario.
- **Hardware y Software**.

Cada parte se evalúa modificando sus parámetros correspondientes con fines comparativos y determinando los que den mejores resultados.

## 7.1 Evaluación de la voz sintética generada

Una parte fundamental del sistema es la información que proporciona al usuario por medio de voz sintética que emite a través de la línea telefónica. Para tener una voz sintética inteligible depende básicamente de la calidad de la voz original, así como en el tamaño de segmentación y número de coeficientes LPCs. Evaluando para diferentes cantidades de coeficientes de predicción lineal podemos observar el comportamiento que tiene la señal sintética como resultado.

En las figuras 7.1 a 7.4 se presentan diferentes pruebas de generación de voz sintética para la palabra “Hola”. En las gráficas de la figura 7.1 se muestra el comportamiento para el caso en que se tienen **ocho coeficientes** de predicción y una segmentación de **200 muestras**, donde ya comienza a ser inteligible la voz sintética.

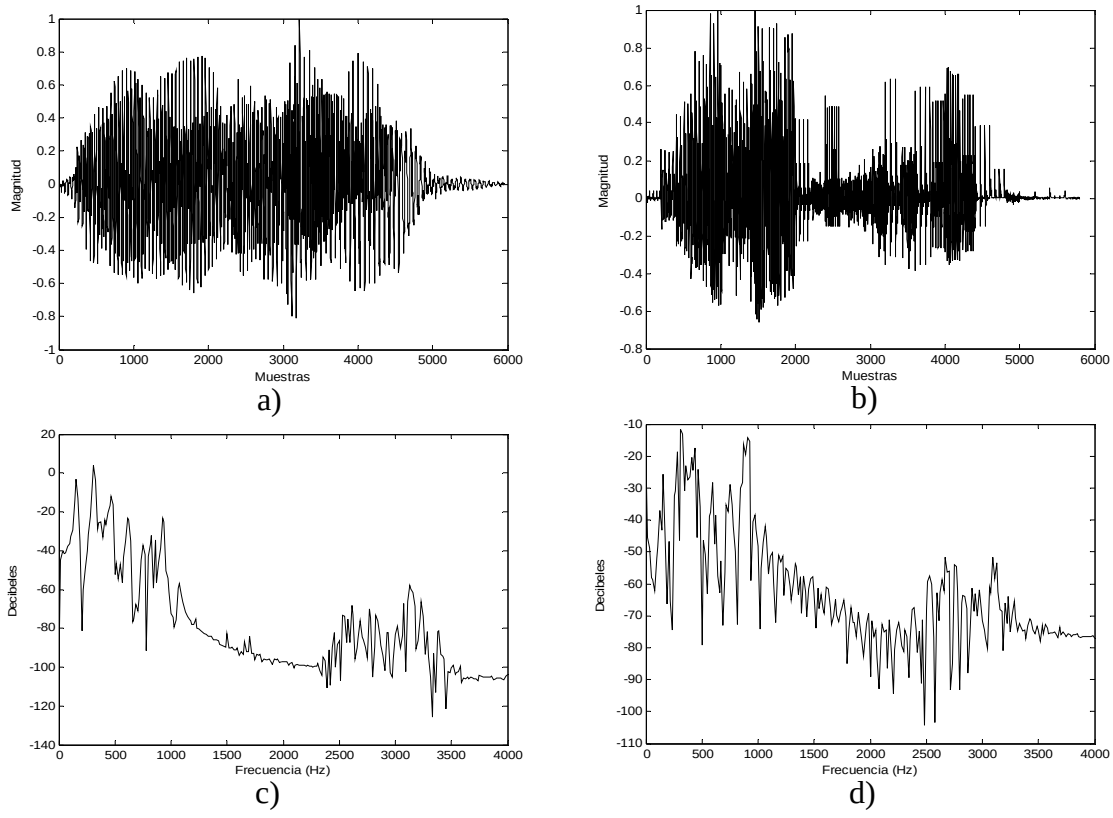



Figura 7.1. Señal: a) Original, b) Sintética, c) Espectro original y d) Espectro sintética

En las gráficas mostradas en la figura 7.2 se observa el comportamiento para el caso en que se tienen **doce coeficientes** de predicción y una segmentación de **400 muestras**.

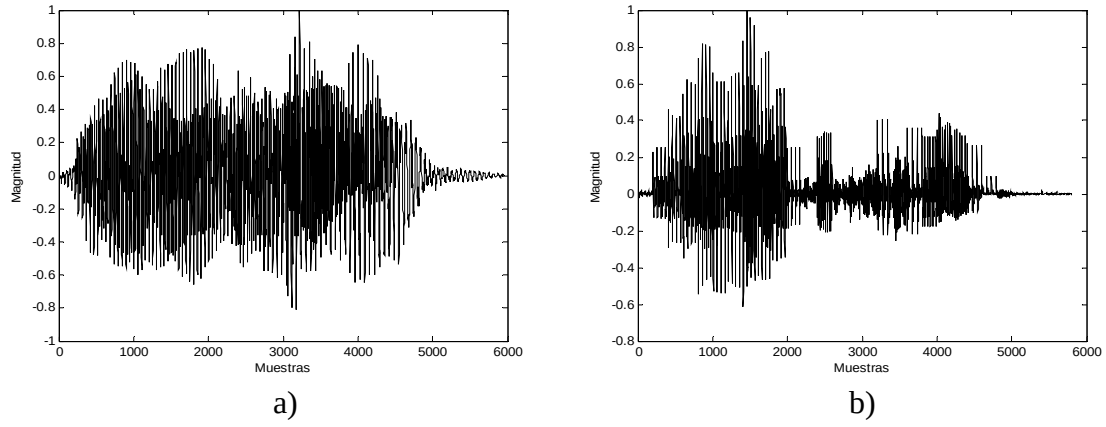
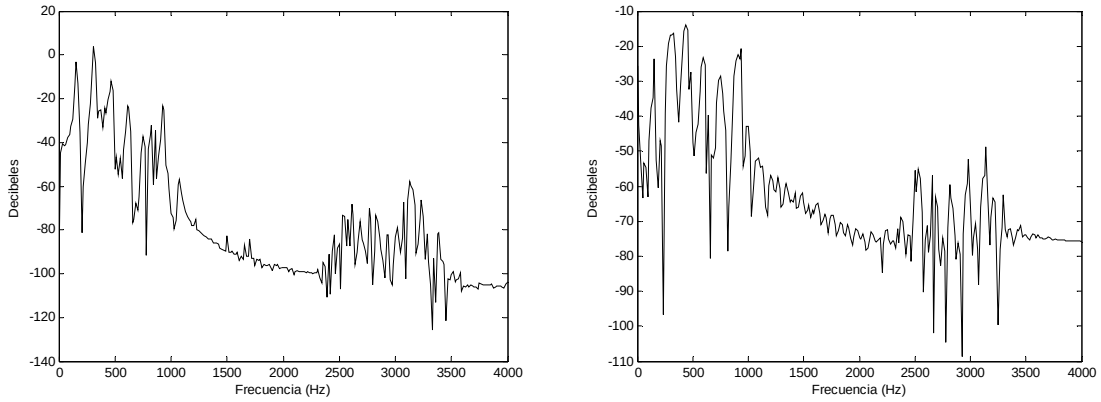


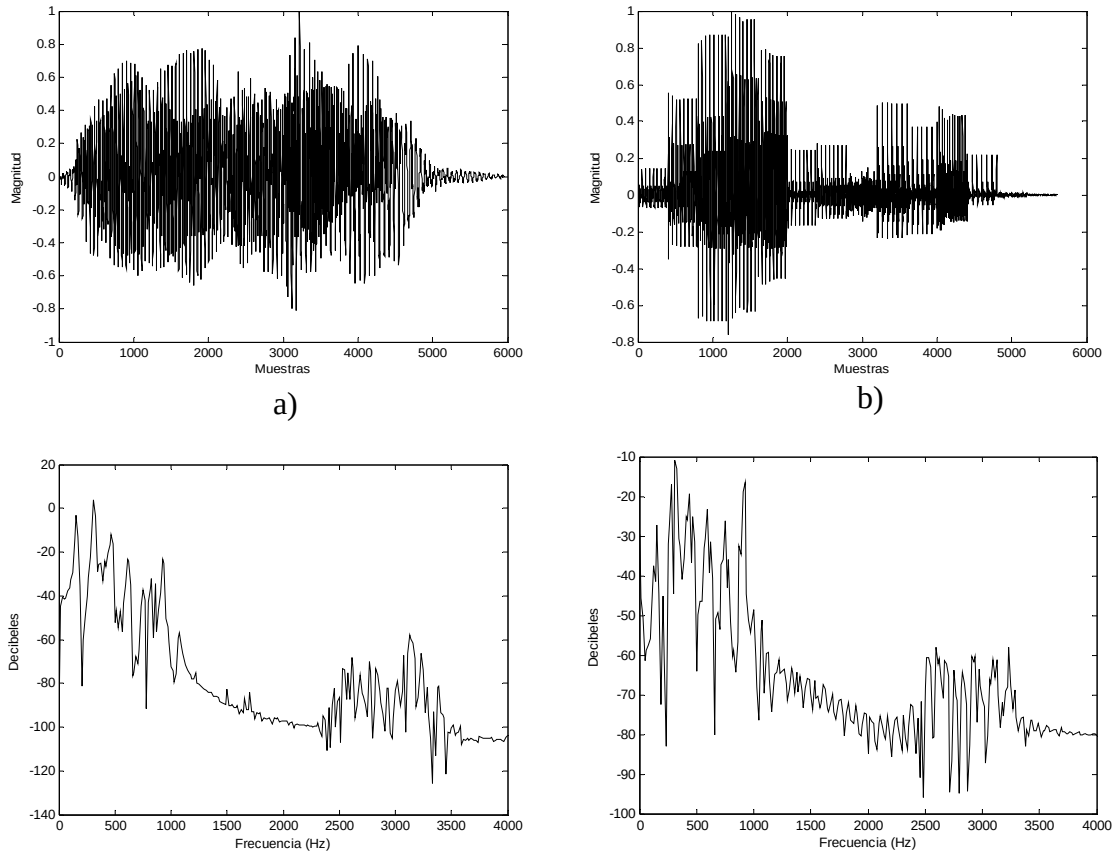

Figura 7.2. Señal: a) Original, b) Sintética.




c) d)

Figura 7.2. Señal: c) Espectro original y d) Espectro sintética

En las gráficas de la figura 7.3 se muestra el comportamiento para el caso en que se tienen **doce coeficientes** de predicción y una segmentación de **200 muestras**.




a) b)

c) d)

Figura 7.3. Señal: a) Original, b) Sintética, c) Espectro original y d) Espectro sintética

Y por último en las graficas mostradas en la figura 7.4 se observa el comportamiento para el caso con **quince coeficientes** de predicción y una segmentación de **400 muestras**, donde se tiene una mejor inteligibilidad de la voz sintética en estas pruebas que en las anteriores.

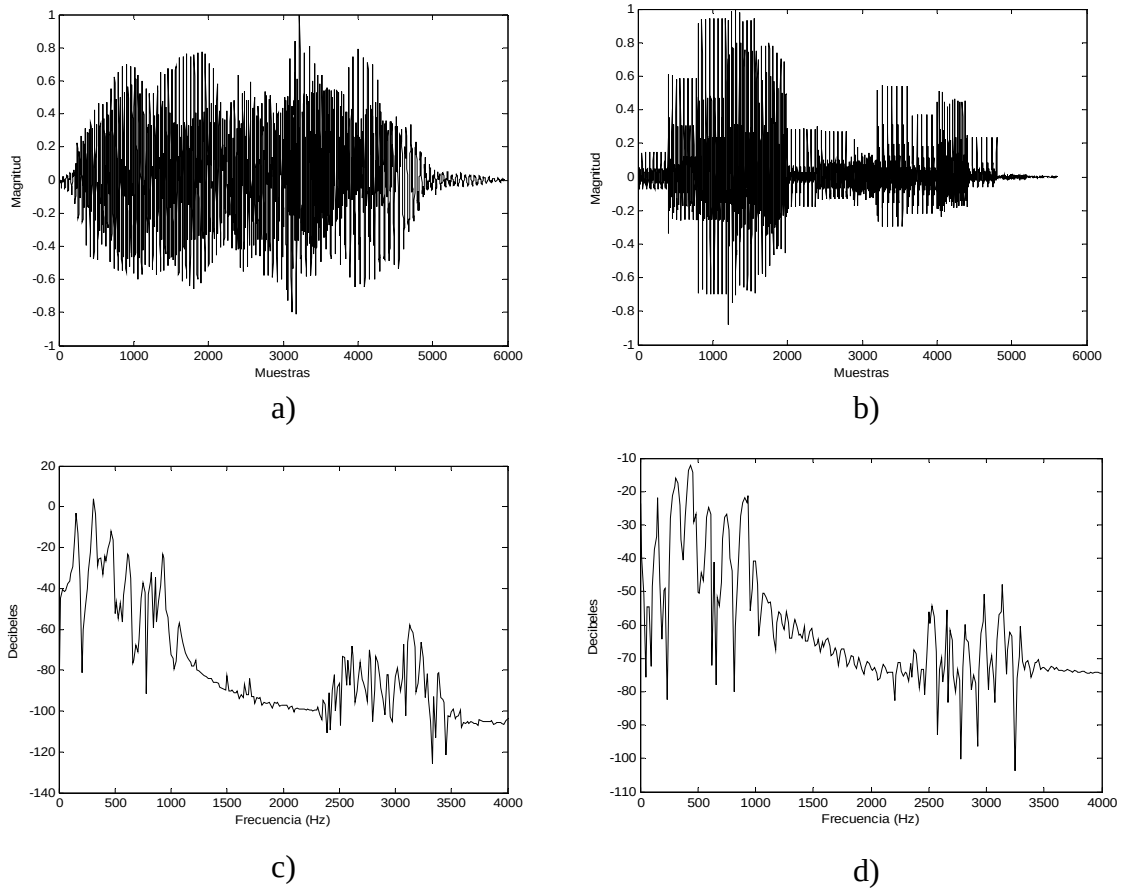



Figura 7.4. Señal: a) Original, b) Sintética, c) Espectro original y d) Espectro sintética

De las gráficas 7.1 a 7.4 se puede observar que a medida que se incrementa el orden de predicción *p,* el espectro de la señal sintética se aproxima al espectro de la señal original por lo que el error de predicción disminuye, dando como resultado una señal sintética más cercana a la original.

## 7.2 Decodificación de tono DTMF

El decodificador de tonos se implementó utilizando el algoritmo de Goertzel, que es el más usado para este tipo de aplicaciones por su facilidad de programación, y especialmente por su eficiencia en el cálculo de la magnitud del espectro, tomando ventaja de que sólo se requiere calcular para ciertas frecuencias que están relacionadas con los tonos DTMF.

Una característica importante para la detección, es la inmunidad al ruido que se caracteriza por la relación señal a ruido (SNR), especialmente cuando el sistema está trabajando en tiempo real se

necesita evaluar su comportamiento a diferentes SNR, que nos indicara si la operación es aceptable. En la tabla 7.1 se muestra una evaluación a diferentes SNR, donde el usuario está con un teléfono celular mientras que el sistema se encuentra conectado a una línea residencial y presionando el digito "1" cada diez segundos hasta completar un minuto (6 veces). Para esto definimos los siguientes términos utilizados para la evaluación realizada:

- **Tono Falso**: Se refiere a la cantidad de tonos que se presentan en el receptor y que no corresponde a una tecla oprimida o bien que el tono interpretado en el receptor es erróneo.

- **Tono Válido**: Indica que el tono decodificado en el receptor corresponde de forma correcta a la tecla presionada en el teclado del teléfono emisor.

- **%Tono válido**: (Tono Válido/Veces Presionada) * 100.

- **%Tono falso**: (Tono Falso/Veces Presionada) * 100.

| SNR(dB) | Tonos Falsos | Tono Válido | %Tono válido | %Tono falso |
|---|---|---|---|---|
| -3 | 28 | 4 | 66 | 466 |
| -2 | 17 | 5 | 86 | 283 |
| -1 | 9 | 5 | 83 | 83 |
| 0 | 3 | 6 | 100 | 50 |
| 1 | 0 | 6 | 100 | 0 |
| 2 | 0 | 6 | 100 | 0 |

Tabla 7.1. Detección del tono uno a diferentes SNR

Como resultado de las pruebas realizadas se puede concluir que el sistema opera adecuadamente a partir de dos decibeles de relación señal a ruido y considerando que en señales reales de DTMF está por arriba de 12dB, por tanto nuestro sistema es confiable.

A continuación se muestran otros resultados con diferentes dígitos.

Para el dígito 2.

| SNR(dB) | Tonos Falsos | Tono Valido | %Tono válido | % Tono falso |
|---|---|---|---|---|
| -3 | 23 | 3 | 50 | 386 |
| -2 | 15 | 4 | 66 | 250 |
| -1 | 11 | 5 | 83 | 183 |
| 0 | 5 | 6 | 100 | 83 |
| 1 | 1 | 6 | 100 | 16 |
| 2 | 0 | 6 | 100 | 0 |

Tabla 7.2. Detección del tono dos a diferentes SNR

Para el dígito 3.

| SNR(dB) | Tonos Falsos | Tono Valido | % Tono válido | % Tono falso |
|---|---|---|---|---|
| -3 | 29 | 3 | 50 | 483 |
| -2 | 14 | 4 | 66 | 233 |
| -1 | 6 | 5 | 83 | 100 |
| 0 | 2 | 6 | 100 | 33 |
| 1 | 0 | 6 | 100 | 0 |
| 2 | 0 | 6 | 100 | 0 |

Tabla 7.3. Detección del tono tres a diferentes SNR

Para dígito 5.

| SNR(dB) | Tonos Falsos | Tono Valido | % Tono válido | % Tono falso |
|---|---|---|---|---|
| -3 | 22 | 4 | 66 | 366 |
| -2 | 12 | 4 | 66 | 200 |
| -1 | 6 | 6 | 100 | 100 |
| 0 | 3 | 6 | 100 | 50 |
| 1 | 0 | 6 | 100 | 0 |
| 2 | 0 | 6 | 100 | 0 |

Tabla 7.4. Detección del tono cinco a diferentes SNR

Para dígito 8.

| SNR(dB) | Tonos Falsos | Tono Valido | % Tono válido | % Tono falso |
|---|---|---|---|---|
| -3 | 31 | 3 | 50 | 516 |
| -2 | 14 | 3 | 50 | 233 |
| -1 | 9 | 5 | 83 | 150 |
| 0 | 3 | 6 | 100 | 50 |
| 1 | 0 | 6 | 100 | 0 |
| 2 | 0 | 6 | 100 | 0 |

Tabla 7.5. Detección del tono ocho a diferentes SNR

De forma gráfica se puede observar el comportamiento de la señal DTMF cuando se le agrega ruido simulado, se da un efecto similar al producido por el canal de transmisión. En la figura 7.5 se muestra el tono correspondiente al dígito ocho, donde prácticamente no presenta defectos de ruido y su correspondiente espectro. En la figura 7.6 se muestra el tono con ruido de 10 dB de SNR y su efecto en el espectro.

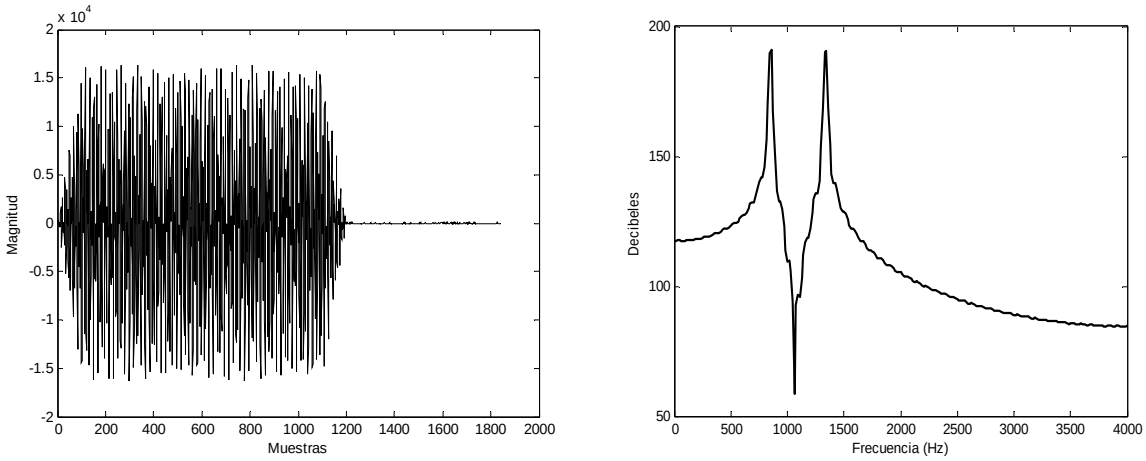



Figura 7.5. Señal y espectro sin ruido.

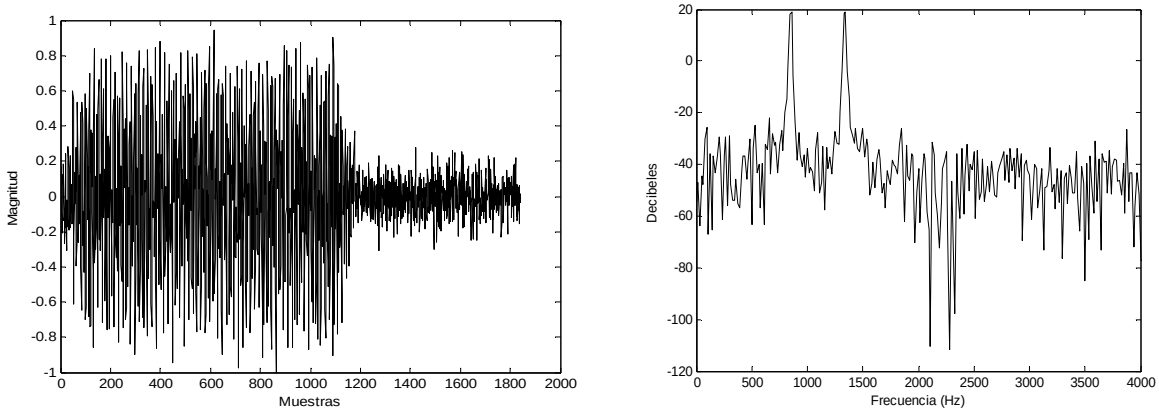



Figura 7.6. Señal y espectro con ruido de 10dB de SNR.

En la figura 7.7 se muestra una comparación del espectro de la señal de tono a diferentes valores de SNR. Como podemos observar el ruido afecta en la amplitud y en menor medida el corrimiento en frecuencia de los tonos. También las componentes fuera de las frecuencias que componen el tono a menor SRN su amplitud aumentan, lo cual ocasiona que los intervalos entre los picos de energía sean menores, dificultando las comparaciones y que se podrían presentar errores al hacer la decodificación del tono.

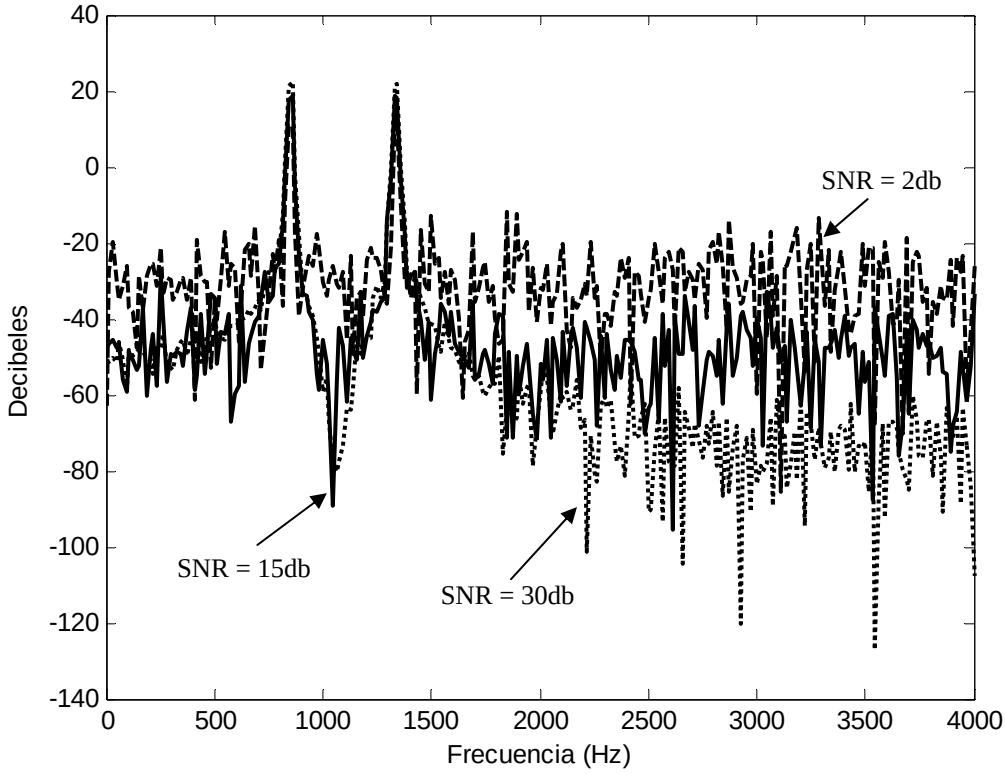



Figura 7.7. Comparación de espectro de un tono a diferentes SNR

En la tablas 7.1, 7.2, 7.3, 7.4 y 7.5 podemos observar que a un SNR de dos decibeles el decodificador funciona correctamente, en la grafica de la figura 7.7 observamos que a dos decibeles aun son totalmente notables las espigas del tono, y mientras aumente más la relación a ruido las espigas del tono serán más notables.

# 7.3 Evaluación de recursos del sistema

En esta sección se realiza la evaluación de recursos utilizados entre los que se encuentran:

- **MIPS**: Millones de instrucciones por segundo, se utiliza para medir qué tanto se tardan los procesos más importantes en realizarse.
- **Memoria**: La cantidad de memoria utilizada en el sistema.
- **Puertos y periféricos**: Puertos utilizados del DSP y periféricos de la tarjeta.

## *7.3.1 Hardware y software*

En lo que respecta al hardware, podemos decir que el contar con las interfaces necesarias para la conexión con la línea telefónica facilitó en gran medida el desarrollo del proyecto, así también

fue necesario conocer su documentación dado que nos proporciona una serie de APIs y banderas para el control y configuración de los diferentes dispositivos proporcionados.

El rendimiento de calcular una muestra estimada a partir de *p* muestras anteriores en el DPS, se observa en la tabla 7.6. Este rendimiento se mide al momento de calcular una muestra a partir del modelo de producción de voz, dependiendo del número de coeficientes de predicción. Para que cumpla con la restricción de funcionamiento para tiempo real, cada muestra debe ser calculada en un tiempo mínimo dado por la frecuencia de muestreo $f_s$ que es de 8000Hz, por lo que el período de tiempo mínimo necesario se calcula por $1/f_s$ correspondiente a 125 microsegundos. El porcentaje de carga del DSP se obtiene al dividir el tiempo que se lleva el proceso entre los 125 microsegundos.

| Orden de Predicción | Tiempo de ejecución (µs) | %Carga del DSP |
|---|---|---|
| 8 | 72 | 57.6 |
| 10 | 91 | 72.8 |
| 12 | 109 | 87.2 |
| 15 | 123 | 98.4 |

Tabla 7.6. Comparación del desempeño en tiempo real.

Otra de las características de la síntesis por predicción lineal, es el nivel de compresión de la señal de voz original. A continuación podemos ver la razón de compresión variando la cantidad de parámetros para su reproducción a través del modelo LPC para el caso de la palabra "Hola" que originalmente tiene 5952 muestras. En la tabla 7.7 se muestran las diferentes tasas de compresión dependiendo del número de parámetros LPC y del tamaño de muestras que componen un segmento o ventana.

| Orden de Predicción "p" | Tamaño de ventana | Número de segmentos | Información de Pitch | Información de ganancia de segmento | Número de parámetros | Taza de compresión |
|---|---|---|---|---|---|---|
| 8 | 400 | 14 | 14 | 14 | 140 | 2.3% |
| 10 | 400 | 14 | 14 | 14 | 168 | 2.8% |
| 12 | 400 | 14 | 14 | 14 | 196 | 3.3% |
| 15 | 400 | 14 | 14 | 14 | 238 | 4.0% |
| 15 | 200 | 29 | 29 | 29 | 493 | 8.3% |

Tabla 7.7. Comparación de razón de compresión.

En lo que respecta a la cantidad de recursos utilizados en el DSK como son la memoria RAM se muestran en la tabla 7.8. Cabe mencionar que la memoria programa y la memoria de datos comparten la RAM y estos son mapeados según la necesidad en el archivo de comandos "*cmd*".

| Palabra | Número de constantes (float) | Memoria bytes |
|---|---|---|
| “bebida” | 323 | 10336 |
| “Chica” | 255 | 8160 |
| “Coca” | 238 | 7616 |
| “Dos” | 323 | 10336 |
| “Grande” | 204 | 6528 |
| “Hawaiana” | 374 | 11968 |
| “Manzana” | 357 | 11424 |
| “Mediana” | 374 | 11968 |
| “Mexicana” | 391 | 12512 |
| “Pastor” | 272 | 8704 |
| “Pepsi” | 238 | 7616 |
| “Pizza” | 272 | 8704 |
| “Selecciona” | 391 | 12512 |
| “Tres” | 289 | 9248 |
| “Uno” | 204 | 6528 |
| **TOTAL** | **4505** | **144160** |

Tabla 7.8. Consumo de memoria de los parámetros.

Además de estos datos se incluye una secuencia de ruido que ocupa 400*32 = 12800 bytes, variables que ocupan 224*32 = 7168 bytes adicionales dándonos un total de 164128 bytes. Si cada kilo byte es de 1024 bytes entonces tendremos un total de 160.28 Kbytes de datos utilizados de los 256Kbytes disponibles de la tarjeta.

Existen dos procesos principales que son los que consumen más tiempo de procesamiento, que son la generación de voz sintética y la decodificación DTMF. Para el primero se tiene que calcular una muestra antes de 125 microsegundos, dado que en ese tiempo se debe entregar una muestra nueva a la salida de la línea telefónica, considerando el escenario en que el orden de predicción es de 15 y con un tamaño de segmento de 400 se consumen 123 microsegundos, lo que corresponden 98.4 MIPS. En la decodificación DTMF consume 33.5 MIPS, es decir, 335 milisegundos en decodificar un dígito por lo que este proceso sería el que consumiría más tiempo de proceso, ambos se pueden mejorar; en primera instancia, al programarlo con operaciones de punto fijo y como segunda mejora, programarlos en ensamblador. En la tabla 7.9 se muestra un resumen de los recursos ocupados por el sistema.

| Recurso | %Uso |
|---|---|
| Memoria | 62.6 |
| Puertos DSP | 14.3 |
| Periféricos DSK | 33.3 |
| MIPS DTMF | 33 |
| MIPS Síntesis | 98.4 |

Tabla. 7.9. Recursos utilizados en el sistema.

## 7.4 Resumen

En este capítulo se analizaron los resultados obtenidos y los recursos utilizados en el proyecto. En lo que respecta a desempeño de procesamiento del DSP para encontrar un dígito DTMF es de 33%, es decir, consume 33 MIPS para llevar el cálculo del dígito por lo que el resto 66 MIPS (66%) se pueden utilizar para realizar otro proceso. En la síntesis de voz se consume el 98.4% por lo que sólo restaría el 1.6% para realizar otra actividad.

El proyecto puede ser extendido en futuros trabajos, ya que la tarjeta de desarrollo utilizada puede expandirse con más periféricos o bien utilizar los que aun están disponibles; como por ejemplo, realizar una interfaz de computadora que explote todos los datos obtenidos a partir de las opciones seleccionadas por los usuarios.

# 8. Conclusiones

En el transcurso del presente trabajo se diseñó e implementó de manera satisfactoria la decodificación DTMF y la síntesis de voz en tiempo real sobre una plataforma DSP, empleando la adquisición de datos a través de la línea telefónica como también el envío de señal a través de ella.

Para la implementación del sintetizador es importante contar con un buen proceso de análisis y contar con una señal de voz original de buena calidad, ya que a partir de ésta, será de igual manera, buena o mala la generación de la señal de voz sintética.

Los parámetros seleccionados para la síntesis, son determinantes para tener una calidad de voz aceptable, en este caso con 15 parámetros de predicción y una ventana de 400 muestras obtuvimos mejor resultados cualitativamente ya que la voz sintética es más clara.

La implementación de la detección DTMF realizada nos da buenos resultados en tiempo real a partir de dos decibeles de SNR.

En las pruebas realizadas en una línea telefónica el SNR es mayor a 15dB por lo que el sistema es funcional y confiable en tiempo real dado que el estándar el SNR es mayor a 12dB.

Según la especificación se deben recibir 10 tonos en un segundo, sin embargo esto no es alcanzable en nuestro sistema dado que tarda 335ms en realizar una conversión, por lo que alcanzamos tres tonos, si se requiere mayor velocidad además de reprogramarlo en punto fijo o en ensamblador se puede reducir la cantidad de muestras para realizar la decodificación pero puede tener repercusiones en la respuesta SNR.

El desempeño de DSP es suficiente para las tareas que se realizan como son: la decodificación DTMF y la síntesis de voz, donde en promedio consume el 66% de MIPS por lo que es posible ocupar el restante 34% para otros procesos.

El hardware utilizado y la herramienta de desarrollo Code Composer, facilitaron en gran medida el desarrollo, pero también es importante contar con otras herramientas que nos permiten realizar tareas de diseño en menor tiempo, como lo es el MatLab. Cabe mencionar que la tarjeta de desarrollo utilizada nos proporcionó las interfaces necesarias para esta tarea, sin embargo, aún tiene otros periféricos que pueden ser explotados en futuros trabajos como una continuación del presente.

# 9. Glosario

| | |
|---|---|
| ADC | Convertidor Analógico Digital. |
| Algoritmo | Conjunto de operaciones y funciones que deben seguirse para resolver algún problema. |
| API | Interface de Programación de Aplicaciones. |
| Artificial | No natural, creado por el hombre. |
| CCS | Code Composer Studio. |
| CID | Identificador de llamada (Caller Indentification) |
| Codificación | Conversión a símbolos para transmisión. |
| Concatenación | Unir o enlazar cosas. |
| Cuasi-periódico | Es cuando la señal tiene un comportamiento en el tiempo tal que sin ser periódico se repite una y otra vez condiciones arbitrariamente cercanas a un estado previo. |
| DAC | Convertidor Digital Analógico. |
| Decodificación | Conversión de símbolos entendibles al receptor. |
| DFT | Transformada discreta de Fourier (Discrete Fourier Transform). |
| DSK | Development Starter Kit. |
| DSP | Procesador digital de señales (Digital Signal Processor). |
| DTMF | Multi-frecuencia de doble tono (Dual Tone Multi-Frecuency). |
| FFT | Trasformada rápida de Fourier (Fast Transform Fourier). |
| FIR | Respuesta Finita al Impulso. |
| Flash | Memoria no volátil eléctricamente borrable. |
| GUI | Interfaz gráfica de usuario. |
| IDE | Ambiente integrado de desarrollo (Integrated Development Environment) |

| | |
|---|---|
| IIR | Respuesta Infinita al Impulso. |
| Inteligible | Que al escuchar se entienda. |
| Internet | Interconexión de redes, es la mayor red del mundo que ofrece diferentes servicios de comunicación. |
| LPC | Codificación por predicción lineal. |
| LUT | Búsqueda en Tabla (Look Up Table). |
| MMSE | Minimización de Error de Mínimos Cuadrados. |
| PC | Computadora personal. |
| PDS | Procesamiento digital de señales. |
| Síntesis | Composición de un conjunto a partir de elementos separados resultado de un análisis. |
| Stand Alone | Es un sistema que trabaja por sí solo. |
| Tono | Sonido a una sola frecuencia audible. |
| UNAM | Universidad Nacional Autónoma de México. |

# 10. Anexos

## Anexo A - Código del DSP.

```
/*****************************************************************************/
/* SISTEMA TELEFONICO CON SINTESIS DE VOZ Y DETECCION DE TONOS
/*
/* Tesis: Carlos Maya León
/*
/* Director de tesis: M. I. Larry Salguero
/*
/*****************************************************************************/
/*****************************************************************************/
/*Archivos include donde también se definen las palabras a reproducir
/*****************************************************************************/
#include <type.h>
#include <board.h>
#include <codec.h>
#include <mcbsp54.h>
#include <daa.h>
#include <math.h>
#include "LPCS_Selecciona.h"
#include "Ryy_Selecciona.h"
#include "PITCH_Selecciona.h"
#include "LPCS_Pizza.h"
#include "Ryy_Pizza.h"
#include "PITCH_Pizza.h"
#include "LPCS_Chica.h"
#include "Ryy_Chica.h"
#include "PITCH_Chica.h"
#include "LPCS_Mediana.h"
#include "Ryy_Mediana.h"
#include "PITCH_Mediana.h"
#include "LPCS_Grande.h"
#include "Ryy_Grande.h"
#include "PITCH_Grande.h"
#include "LPCS_Hawaiana.h"
#include "Ryy_Hawaiana.h"
#include "PITCH_Hawaiana.h"
#include "LPCS_Mexicana.h"
#include "Ryy_Mexicana.h"
#include "PITCH_Mexicana.h"
#include "LPCS_Pastor.h"
#include "Ryy_Pastor.h"
#include "PITCH_Pastor.h"
#include "LPCS_Bebida.h"
#include "Ryy_Bebida.h"
#include "PITCH_Bebida.h"
#include "LPCS_Pepsi.h"
#include "Ryy_Pepsi.h"
```

```
#include "PITCH_Pepsi.h"
#include "LPCS_Coca.h"
#include "Ryy_Coca.h"
#include "PITCH_Coca.h"
#include "LPCS_Manzana.h"
#include "Ryy_Manzana.h"
#include "PITCH_Manzana.h"
#include "LPCS_Uno.h"
#include "Ryy_Uno.h"
#include "PITCH_Uno.h"
#include "LPCS_Dos.h"
#include "Ryy_Dos.h"
#include "PITCH_Dos.h"
#include "LPCS_Tres.h"
#include "Ryy_Tres.h"
#include "PITCH_Tres.h"
#include "Ruido401.h"

/*****************************************************************************/
/* Constantes utilizadas para le sistema
/*****************************************************************************/
#define VENTANA 400
#define ORDEN_P 15
#define N_GOERTZEL 205
#define NUMERO_VENTANAS_DIGITO 2

/*****************************************************************************/
/* Funciones prototipo
/*****************************************************************************/
void delay(s16);

/*****************************************************************************/
/* Global variables
/*****************************************************************************/
float Coeff[8]={1.7033,1.6359,1.5623,1.4829,1.1631,1.0088,0.7901,0.5595};
int Digitos[4][3]={{1,2,3},{4,5,6},{7,8,9},{-1,-1,-1}};
float Muestras[N_GOERTZEL];
HANDLE hDevice;
int Seleccion[100][3];
int nContadorUsuario = 0;

/*************************************************/
/*Estructura: MATRIZ que se encarga de almacenar
/*                              un array de valores enteros
/*************************************************/
typedef struct MATRIZ
{
        int nRenglones;
        int nColumnas;
        float buffer[VENTANA];
}MATRIZ;

/*******************************************************/
/*Funcion: Get
```

```
/*Descripción: Recupera un valor de estructura MATRIZ
/*                              Considerando indice base 1
/*Parametros:       mat (in) Apuntador a MATRIZ
/*                                      r (in) Valor del renglon base 1
/*                                      c (in) Valor del columna base 1
/*Return: regresa el valor dado por r y c
/*************************************************************/
float Get(MATRIZ *mat,int r,int c)
{
        //if(r<1 || c<1)
        //        return -999999;
        r = r-1;
        c = c-1;
        return mat->buffer[r*(mat->nColumnas-1) + r+c];
}

/*************************************************************/
/*Funcion: Set
/*Descripción: Establece un valor de estructura MATRIZ
/*                              Considerando indice base 1
/*Parametros:       mat (in) Apuntador a MATRIZ
/*                                      r (in) Valor del renglon base 1
/*                                      c (in) Valor del columna base 1
/*Return: nada
/*************************************************************/
void Set(MATRIZ *mat,int r,int c,float fvalor)
{
        //if(r<1 || c<1)
        //        return;
        r = r-1;
        c = c-1;
        mat->buffer[r*(mat->nColumnas-1) + r+c] = fvalor;
}

/*************************************************************/
/*Funcion: GenerarVoz
/*Descripción: Se encarga de realizar sistesis de voz
/*                              por el procedimiento de LPC
/*Parametros:       nTotalVentanas (in) Total de frames
/*                                      fpLPCS[][ORDEN_P] (in) Coeficientes LPC
/*                                      fpRYY[] (in) Ganancia de ventanas
/*                                      npPitch[] (in) Indica el periodo de cada
/*                                                              ventana
/*Return: nada
/*************************************************************/
void GeneraVoz(int nTotalVentanas,float fpLPCS[][ORDEN_P],float fpRYY[],int npPitch[])
{
        int n,i;
        int tr;
        int k2;
        float fSalida;
        float fSalidatemp = 0;//LLevara la muestra anterior
        MATRIZ z,ex;
        z.nRenglones = 1;
```

```
		z.nColumnas = VENTANA;
		ex.nRenglones = 1;
		ex.nColumnas = VENTANA;

		for(tr=1;tr<=nTotalVentanas;tr++)
		{

			if(npPitch[tr-1]>0)
			{
				for(i = 1;i<=VENTANA;i++)
					Set(&ex,1,i,0);
				for (i=1;i<=VENTANA;i+=npPitch[tr-1])
					Set(&ex,1,i,1);
			}
			else
			{
				for(i = 1;i<=VENTANA;i++)
					Set(&ex,1,i,Ruido401[i-1]*0.1);
			}

			fSalida = 0;
			for( n=1;n<=VENTANA;n++)
			{
				Set(&z,1,n,0);

				if(fpRYY[tr-1]>0)//Si se trata de una trama de silencio vale cero
				{
					for(k2=1;k2<=ORDEN_P;k2++)
					{

						if(n-k2>0)
							Set(&z,1,n,Get(&z,1,n)+(fpLPCS[tr-1][k2-1]*Get(&z,1,n-k2)));
						else
							break;

						Set(&z,1,n,Get(&z,1,n)+fpRYY[tr-1]*Get(&ex,1,n));
					}

					fSalidatemp = Get(&z,1,n)+0.9375*fSalidatemp;
					fSalida = fSalidatemp*100;
				}

				while (!MCBSP_RRDY(HANDSET_CODEC)) {};
				*(volatile u16*)DXR1_ADDR(DAA_CODEC) = fSalida;
			}
		}
}

/*********************************************************/
/*Funcion: MaxIndex
/*Descripción: Encuentra el maximo en el fBuffer
/*Parametros:	fBuffer[] (in) fBuffer tamaño 4
/*Return: Regresa el indice encontrado del maximo
/*********************************************************/
```

```
int MaxIndex(float fBuffer[])
{
	float Max=0;
	int nIndex=0;
	float MaxAnt = 0;
	int i;
	MaxAnt = fBuffer[0];
	Max = fBuffer[0];
	for(i=1;i<4;i++)
	{
		if(fBuffer[i]>Max)
		{
			MaxAnt = Max;
			Max = fBuffer[i];
			nIndex = i;
		}
	}
	return nIndex;
}

/*********************************************************/
/*Funcion: Goertzel
/*Descripción: Encuentra el espectro con le algoritmo de
/*                    Goertzel de frecuencias objetivos
/*Parametros:    fBuffer[] (in)  Buffer a procesar
/*                              nN (in) Numero de muestras de fBuffer
/*                              pPromedio (out) Promedio de la magnitud
/*Return: regresa el indice del digito calculado
/*********************************************************/
int Goertzel(float fBuffer[],int nN,float *pPromedio)
{
	float Magnitud[8];
	int Frec,i,FrecL,FrecH;
	*pPromedio = 0; //Se reinicia variable global del promedio a cero
	for (Frec=0;Frec<8;Frec++)
	{
	    float Q0 = 0;
	    float Q1 = 0;
	    float Q2 = 0;
	    for (i=0;i<nN;i++) //Ciclo del Algoritmo
	   {
		Q0 =  Coeff[Frec]*Q1-Q2+fBuffer[i];
		Q2 = Q1;
		Q1 = Q0;
	    }
	    Magnitud[Frec] = (Q1*Q1)+(Q2*Q2)-(Q1*Q2*Coeff[Frec]); //Calculo magnitud
	   *pPromedio = Magnitud[Frec]/8 +*pPromedio;
	}
   //El promedio regresa como referencia
	//******Aqui va la comprobacion de los maximos
	FrecL = MaxIndex(&Magnitud[0]);
	FrecH = MaxIndex(&Magnitud[4]);

	return Digitos[FrecL][FrecH]; //Regresa el indice del DTMF calcuado
```

```
}

/*****************************************************************************/
/* MAIN
/*****************************************************************************/
void main()
{
   s16 cnt;
   u16 data;
   u16 silent_count;
            int nIndex = 0;
            int nDigito,nDigitoAnt;
            float fPromedio;
            int nContador;
            int nEstadoPedido = 0;//0.-Seleccion tamaño, 1.-Selección sabor, 2.- seleccion bebida

            if (brd_init(100))
                        return;

   cnt = 2;

   /* Señalización visual de que el programa inicia */
            while ( cnt-- )
            {
                        brd_led_toggle(0);
                        delay(1000);
                        brd_led_toggle(1);
                        delay(1000);
                        brd_led_toggle(2);
                        delay(1000);
            }
   /* Configura DAA codec */
   hDevice = codec_open(DAA_CODEC);
   codec_sample_rate(hDevice,SR_8000);

   /* Configura HANDSET codec */
   hDevice = codec_open(HANDSET_CODEC);
   codec_sample_rate(hDevice,SR_8000);

   /* Inicializa DAA con valores default (off-hook, no caller ID) */
   daa_init();


start:
   brd_led_disable(0);
   brd_led_disable(1);
   brd_led_disable(2);


   /* en espera de ring */
   silent_count = 0;
   while (!daa_ring_detect())
   {
      brd_delay_msec(5);
```

```
   silent_count++;

   /* Señalización visual de ring */
   if (silent_count == 20)
   {
      brd_led_toggle(0);
      silent_count = 0;
   }
}


/* habilta caller ID */
brd_led_disable(0);
brd_led_enable(1);
daa_cid(DAA_CID_ENABLE);

/* Espera a que el primer ring termine */
silent_count = 0;
do
{
   brd_delay_msec(5);
   if (!daa_ring_detect())
      silent_count++;
   else silent_count = 0;
} while (silent_count < 600);


/* Deshabilira caller ID */
daa_cid(DAA_CID_DISABLE);

brd_led_disable(1);
brd_led_enable(2);

/* Espera segundo ring */
silent_count = 0;
do
{
   brd_delay_msec(5);
   if (!daa_ring_detect())
      silent_count++;
   else break;
} while (silent_count < 1200);

/* checa si se colgo regresa al esperar el primer ring */
if (silent_count >= 1200) goto start;

brd_led_disable(2);

/* Descuelga */
daa_offhook();

/* enciende leds para indicar que es atendida la llamada */
brd_led_enable(0);
brd_led_enable(1);
```

```
brd_led_enable(2);

GeneraVoz(nLPCS_Selecciona,LPCS_Selecciona,ryy_Selecciona,PITCH_Selecciona);
delay(200);
GeneraVoz(nLPCS_Pizza,LPCS_Pizza,ryy_Pizza,PITCH_Pizza);

delay(500);
GeneraVoz(nLPCS_Uno,LPCS_Uno,ryy_Uno,PITCH_Uno);
delay(200);
GeneraVoz(nLPCS_Grande,LPCS_Grande,ryy_Grande,PITCH_Grande);
delay(500);
GeneraVoz(nLPCS_Dos,LPCS_Dos,ryy_Dos,PITCH_Dos);
delay(200);
GeneraVoz(nLPCS_Mediana,LPCS_Mediana,ryy_Mediana,PITCH_Mediana);
delay(500);
GeneraVoz(nLPCS_Tres,LPCS_Tres,ryy_Tres,PITCH_Tres);
delay(200);
GeneraVoz(nLPCS_Chica,LPCS_Chica,ryy_Chica,PITCH_Chica);

nEstadoPedido = 0;
nDigito = -1; //Inicia variables para la deteccion DTMF
nDigitoAnt = -1;
        while(1)
        {
                while(1) //Detectando gidito
                {
                        nIndex = 0;
                   while (nIndex < N_GOERTZEL)
                   {
                      /* Verifica que exista una muestra */
                      if (MCBSP_RRDY(DAA_CODEC))
                      {
                         /* Se obtienen muestras para Goertzel */
                         data = *(volatile u16*)DRR1_ADDR(DAA_CODEC);

                         Muestras[nIndex++] = (float)data/65536;
                      }
                   }
                   nDigito = Goertzel(Muestras,N_GOERTZEL,&fPromedio);//Se verifica Goertzel
                   if(nDigito>-1) //Verifica si se trata de un digito valido
                   {
                        if(nDigito == nDigitoAnt)
                        {
                                if(fPromedio >80)
                                {
                                        nContador++;
                                }
                                else
                                {
                                        nContador = 0;
                                        nDigitoAnt = -1;
                                }

                        }
```

```
                else
                {
                        nDigitoAnt = nDigito;
                        nContador = 0;
                }
        }
        else
        {
                nDigitoAnt = -1;
                nContador = 0;
        }
        //El digito debe ser igual por lo menos en 2 ventanas consecutivas para se valido
        if(nContador == NUMERO_VENTANAS_DIGITO)
                break;
}
if(nDigito<4) continue;//Si la tecla presionada es mayor a 3 no hace nada
if(nContadorUsuario >= 100) continue; //Si ya no hay registro para almacenar no hace nada

Seleccion[nContadorUsuario][nEstadoPedido] = nDigito;//Registra selección
switch(nEstadoPedido)
{

        case 0:
                nEstadoPedido = 1; //Cambia al siguiente menu
                GeneraVoz(nLPCS_Selecciona,LPCS_Selecciona,ryy_Selecciona,PITCH_Selecciona);
                delay(500);
                GeneraVoz(nLPCS_Uno,LPCS_Uno,ryy_Uno,PITCH_Uno);
                delay(200);
                GeneraVoz(nLPCS_Hawaiana,LPCS_Hawaiana,ryy_Hawaiana,PITCH_Hawaiana);
                delay(500);
                GeneraVoz(nLPCS_Dos,LPCS_Dos,ryy_Dos,PITCH_Dos);
                delay(200);
                GeneraVoz(nLPCS_Mexicana,LPCS_Mexicana,ryy_Mexicana,PITCH_Mexicana);
                delay(500);
                GeneraVoz(nLPCS_Tres,LPCS_Tres,ryy_Tres,PITCH_Tres);
                delay(200);
                GeneraVoz(nLPCS_Pastor,LPCS_Pastor,ryy_Pastor,PITCH_Pastor);
                break;
        case 1:
                nEstadoPedido = 2; //Cambia al siguiente menu
                GeneraVoz(nLPCS_Selecciona,LPCS_Selecciona,ryy_Selecciona,PITCH_Selecciona);
                delay(200);
                GeneraVoz(nLPCS_Bebida,LPCS_Bebida,ryy_Bebida,PITCH_Bebida);
                delay(500);
                GeneraVoz(nLPCS_Uno,LPCS_Uno,ryy_Uno,PITCH_Uno);
                delay(200);
                GeneraVoz(nLPCS_Coca,LPCS_Coca,ryy_Coca,PITCH_Coca);
                delay(500);
                GeneraVoz(nLPCS_Dos,LPCS_Dos,ryy_Dos,PITCH_Dos);
                delay(200);
                GeneraVoz(nLPCS_Pepsi,LPCS_Pepsi,ryy_Pepsi,PITCH_Pepsi);
                delay(500);
                GeneraVoz(nLPCS_Tres,LPCS_Tres,ryy_Tres,PITCH_Tres);
                delay(200);
```

```
                        GeneraVoz(nLPCS_Manzana,LPCS_Manzana,ryy_Manzana,PITCH_Manzana);
                    break;
                    case 2:
                        nContadorUsuario++;
                        daa_onhook();
                        goto start;
                    break;
                }
    }
    return;
}

void delay(s16 period)
{
    int i, j;

    for(i=0; i<period; i++)
    {
        for(j=0; j<period>>1; j++);
    }
}
```

# 11. Bibliografía


[1] Rabiner L. Digital Processing of Speech Signals. Prentice Hall. USA. 1978.

[2] Huang X. Spoken Language Processing : A Guide to Theory, Algorithm and System Development. Prentice Hall. USA. 2001.

[3] Miyara F. La voz humana. Universidad Nacional de Rosario. Argentina 2001.

[4] Gómez, J.C. Modelo de producción de voz. Universidad Nacional de Rosario. Argentina 2001.

[5] Duxans H. & Bonafonte A. Revisión Técnica de Estimación de Pulso Glotal basada en filtrado inverso. Universidad Politécnica de Catalunya. España.

[6] Barajas G. P. & Molero M. A. Síntesis de voz en tiempo real empleando una arquitectura DSP. UNAM. FI. México D.F. 2004

[7] Rodger J. M. Signal processing for Melody Transcription. Department of Computer Science, University of Waikato, Hamilton. New Zealand.

[8] Chu W. C. Speech Coding Algorithms. Foundation and Evolution of Standardized Coders. Wiley-Interscience. USA. 2003.

[9] Kondoz A. M. Digital Speech. Coding for Low Bit Rate Communication System. Segunda Edición. John Wiley & Sons. Inglaterra. 2004

[10] Quatieri T. F. Speech Signal Processing. Principles and Practice. Prentice Hall PTR. USA. 2002.

[11] Escobar S. L. Diseño de Filtros Digitales. FI. UNAM. México D.F. Noviembre 2006

[12] J. Bigelow S. Understanding Telephone Electronics. Fourth Edition, Newnes, USA 2001.

[13] Proakis J. G. Digital Communication.McGraw-Hill. New York. USA. 2001

[14] Proakis J. G. & Manolakis D. G. Digital Signal Processing Principles, Algorithms, and Applications. Prentice-Hall International. Tercera edición. USA. 1996.

[15] Kuo, S. M. & Lee, B.H. Real-Time Digital Signal Processing. John Wiley & Sons Ltd. USA. 2001

[16] Massey T. & Iyer R. DSP Solution for Telephony and Data/Facsimile Modems. Texas Instruments. USA. 1997

[17] Papamichalis P. E. Practical Approaches to Speech Coding. Prentice Hall. USA. 1987.

[18] Salamanca G. & Marlett S. Curso básico de fonética general. SIL Internacional. 2004

[19] Levinson S. E. Mathematical Models for Speech Technology. John Wiley & Sons Ltd. England. 2005

[20] Smith W. S. The Scientist and Engineer's Guide to Digital Signal Processing. Segunda edición. California Technical Publishing. USA. 1999.

[21] Texas Instruments. TLC320AD50C Data Manual. Mixed Signal Products. Literature number SLAS131E. 2000

[22] Barr M. Programming Embedded Systems in C and C++. O' Reilly. USA. 1999

[23] Ball R. S. Embedded Microprocessor System. Real World Design. Newnes. Segunda Edición. USA. 2002

[24] Swokowski E.W. Cálculo con Geometría Analítica. Segunda Edición. Grupo Editorial Iberoamérica. México D.F. 1989.

[25] Hayes M. H. Digital Signal Processing. McGraw-Hill. USA. 1999

[26] Oppenheim A.V. Signal and Systems. Prentice Hall. USA. 1983

[27] Murray R. S. y Abellanas L. Fórmulas y Tablas de Matemáticas Aplicadas. McGraw-Hill, México 1993.

[28] Escobar S. L. Conceptos Básicos de Procesamiento Digital de Señales. FI, UNAM, México D.F. Marzo 2009.

[29] Escobar S. L., Pseniscka B. y Molero A. M. Arquitectura de DSPs, Familias TMS320C54x y TMS320C54xx, y Aplicaciones. FI, UNAM, México D. F. Octubre 2005.