UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO DE UN CODIFICADOR DE VOZ PARA TELEFONIA CELULAR GSM BASADO EN EL ALGORITMO CELP FS1016

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN TELECOMUNICACIONES

P R E S E N T A

XONIA IVONNE OLAVARRIETA ARRUTI

DIRECTOR DE TESIS: DR. JOSE ABEL HERRERA CAMACHO

CIUDAD UNIVERSITARIA MARZO 2005

Al Ing. Fernando Olavarrieta

A ti te dedico mis versos, mi ser, mis victorias,
a ti mis respetos Señora, Señora, Señora.
A ti mi guerrera invencible,
a ti luchadora incansable,
a ti mi amiga constante de todas las horas.

Agradecimientos

A las personas que participaron en esta tesis como parte de las pruebas de escucha realizadas: Adrián, Camello, Fernando, Ginny, Luis, Mariam, Peco y Toño. Y un doble agradecimiento a mi hermano Fernando por ayudarme en la captura de la información contenida en esta tesis.

Además un agradecimiento a las personas que me apoyaron a lo largo de estos cinco años: profesores y amigos. A los miembros del departamento de Telecomunicaciones, en especial a Víctor García, Miguel Moctezuma y Federico Vargas; así como a mi director de tesis, Abel Herrera. A mis amigos Luis, Rodrigo, Liliana y Agustín, con quienes compartí gran parte de este tiempo, dentro y fuera de la facultad.

Índice general

Índice de figuras

Índice de cuadros

Índice de Códigos

Resumen

El principal medio de comunicación entre los hombres es la voz. Esto se encuentra reflejado en la tecnología moderna que hoy en día transmite, almacena, manipula, reconoce y sintetiza voz.

El desarrollo de VLSI y técnicas DSP han alentado la implementación de algoritmos altamente complejos de procesamiento de voz. Como resultado, la tecnología del procesamiento de voz está siendo usada en las telecomunicaciones y los negocios, en aplicaciones como correo de voz, sistemas de comunicaciones personales, operadores autómatas, seguridad activada por voz, entre otras.

Como en la mayoría de las aplicaciones, en la telefonía celular se ha vuelto indispensable reducir cada vez más la tasa de bits de la información a transmitir, ya que el número potencial de usuarios es muy grande y la capacidad del canal limitada, además de los altos costos de transmisión y almacenamiento.

Dentro de los sistemas de comunicaciones digitales, los codificadores de voz son los responsables de generar representaciones digitales compactas de las señales de voz con el fin de transmitirlas de forma eficiente o almacenarlas. Sin embargo, a bajas tasas de bits, siempre existirá una pérdida en la información, por lo que el objetivo de los sistemas de codificación es maximizar la calidad para una tasa de bits dada, o bien, minimizar la tasa de bits para una calidad dada.

Una clase de codificadores de voz que han sido desarrollados exitosamente son los codificadores de voz basados en la predicción lineal. En estos codificadores, la voz codificada es sintetizada mediante la excitación de un filtro todo polos variante en el tiempo que modela al tracto vocal humano. Los coeficientes del filtro se obtienen mediante un método de estimación regresivo y describen la envolvente espectral de la señal. La predicción lineal constituye la base de varios codificadores a diferentes tasas de bits, como los codificadores CELP usados en telefonía celular, que forman parte de numerosos estándares de codificación.

Un codificador de voz comercial debe mantener un nivel de calidad aceptable bajo todas las circunstancias de operación del mismo; por ejemplo, los teléfonos celulares son usados por muchos usuarios en una amplia variedad de ambientes, consecuentemente deben codificarse diferentes acentos y lenguajes, en muy diversas condiciones de canal y ruidos de fondo. Por lo tanto es necesario realizar pruebas rigurosas antes de que un algoritmo pueda ser estandarizado o aplicado en un producto comercial.

Un mejor entendimiento de las etapas de un codificador de voz es el primer paso para comprender las interacciones entre las diferentes etapas. Además esto puede conducir a un mejoramiento de las técnicas existentes o incluso al desarrollo de nuevas alternativas que no hayan sido consideradas.

La codificación de voz es una importante área de investigación en el procesamiento digital de señales. Constituye un elemento fundamental en las comunicaciones digitales y ha evolucionado rápidamente en paralelo con el aumento de la demanda de los servicios de telecomunicaciones.

CAPÍTULO 1

Introducción. Conceptos básicos sobre la codificación de voz

La voz siempre ha sido el principal medio de comunicación entre humanos, razón por la cual ha sido muy estudiada en su forma y contenido.

Existen varias formas de representar a la voz como medio de comunicación, de entre las cuales la más útil en aplicaciones prácticas ha sido la caracterización de la voz en términos de la señal que transporta la información sobre el mensaje, esto es, la onda acústica.

En los sistemas de comunicaciones por voz, la señal de voz es transmitida, almacenada y procesada de diferentes maneras. En general, existen dos problemas que deben ser considerados en cualquiera de estos sistemas:

- Preservación del contenido del mensaje en la señal de voz.
- Representación de la señal de voz en una forma que sea conveniente para su transmisión o almacentamiento, o bien en una forma flexible de manera que puedan realizarse modificaciones a la señal de voz sin degradar seriamente el contenido del mensaje.

El procesamiento digital de señales (DSP) involucra la obtención de representaciones discretas de una señal basadas en un modelo dado y la posterior aplicación de alguna transformación con el fin de representar la señal en una forma más conveniente. Por tanto, el procesamiento digital de la voz generalmente involucra dos tareas. Primero, la obtención de una representación discreta general de la señal de voz ya sea de forma de onda o paramétrica. En segundo lugar, el procesamiento digital de señales cumple la función de ayudar en el proceso de transformación de la representación discreta de la señal en formas alternativas más apropiadas para aplicaciones específicas.

El campo completo del procesamiento de la voz se encuentra experimentando cambios revolucionarios ocasionados por la maduración de las técnicas y sistemas DSP. Es justo decir que varios de los algoritmos usados en el procesamiento digital de señales fueron desarrollados o puestos primero en práctica por personas que trabajaban en sistemas de procesamiento de voz. Estos algoritmos incluyen técnicas de filtrado digital, PCM, LPC, STFT, representaciones generales en tiempo-frecuencia, técnicas de filtrado adaptable, técnicas de bancos de filtros, HMM y

varios más. Por tanto, estudiando el comportamiento de los sistemas de procesamiento de voz, es posible estudiar el comportamiento de varios de los algoritmos más importandes en el procesamiento digital de señales.

En general, la codificación de voz puede ser considerada como una especialidad particular en el amplio campo del procesamiento de voz, que también incluye el análisis y reconocimiento de voz. La codificación o compresión de la voz es el campo concerniente a la obtención de representaciones digitales compactas de las señales de voz con el fin de transmitirlas de forma eficiente o almacenarlas. Como cualquier otra señal continua en el tiempo, la voz puede ser representada digitalmente a través del muestreo y cuantización. Sin embargo, como muchas otras señales, la voz muestreada contiene una gran cantidad de información que puede ser redundante o perceptualmente irrelevante. La mayoría de los codificadores para telecomunicaciones involucran pérdidas, lo que significa que la voz sintetizada es perceptualmente similar a la original pero puede ser físicamente diferente. El objetivo primordial de la codificación de la voz es representar las señales de voz digitalizadas usando el mínimo número de bits posible pero manteniendo su calidad perceptible.

De todas las áreas del procesamiento de voz, la codificación de voz es la más madura. Existen tres razones:

La primera es simplemente el hecho de que la voz es una de las señales de amplio interés con menor ancho de banda. Las señales con calidad telefónica poseen un ancho de banda de solo 3.2 kHz aproximadamente, mientras que se puede obtener voz de alta calidad con un ancho de banda de 5-6 kHz. Por lo tanto, las bajas frecuencias de muestreo de 6400-12000 muestras por segundo provocan que el procesamiento en tiempo real de la voz sea más rentable que para casi cualquier otra área importante de aplicación.

Un segundo factor importante en el desarrollo de los sistemas de codificación de voz es la continua revolución ocasionada por el rápido desarrollo de la tecnología de circuitos VLSI. Por supuesto, VLSI ha tenido un impacto masivo en casi todas las áreas de la tecnología moderna. Sin embargo, existe una relación especial entre los algoritmos de codificación de voz, la tecnología VLSI y la industria de las telecomunicaciones. Debido a su baja frecuencia de muestreo, la codificación de voz es usualmente la primer área de aplicación DSP usada en el desarrollo de tecnología digital VLSI. Sobre esta relación se encuentra el tremendo impacto económico de la multimillonaria industria internacional de las telecomunicaciones, que requiere soluciones digitales efectivas en áreas como la conmutación de voz, sistemas combinados de voz/datos, correo de voz, telefonía móvil, comunicaciones móviles vía satélite, entre otras. Consecuentemente, los codificadores de voz forman la base de un conjunto único de algoritmos, tecnologías y aplicaciones, que se han vuelto componentes esenciales en las telecomunicaciones.

El tercer factor de importancia es la efectividad de varios algoritmos DSP en la solución de diferentes problemas fundamentales asociados con los sistemas de codificación de voz. Las técnicas DSP han probado ser muy efectivas en el modelado tanto de la producción como de la percepción de la voz. Esto no quiere decir que DSP haya resuelto todos los problemas de los codificadores de voz. Sin embargo, los logros obtenidos en los sistemas de codificación de voz en los pasados 30 años han sido fenomenales, y gran parte de esto puede ser atribuido a la aplicación de algoritmos DSP a los problemas de codificación de voz.

1.1. Producción de la voz

Con el fin de aplicar las técnicas DSP en las comunicaciones por voz es esencial entender los fundamentos del proceso de la producción de la voz.

Los órganos productores de sonidos se agrupan en tres regiones (ver figura 1.1):

- Tracto pulmonar o respiratorio: Formado por los pulmones y la tráquea. Genera flujos de aire. Éstos órganos controlan la amplitud de los sonidos, y la única contribución audible del tracto pulmonar son los silencios inter y entre palabras.
- Laringe: Área situada superiormente a la tráquea e inferiormente a la faringe. Conduce los flujos de aire generados en los pulmones hacia el tracto vocal. La laringe contiene a las cuerdas vocales, las cuales estrechan la trayectoria entre los pulmones y el tracto vocal, y al pasar el aire generado en la exhalación a través de ellas, se abren y cierran parcial o totalmente (vibran), de manera rápida y secuencial, produciendo sonidos. Esta función es llamada excitación.

- Tracto vocal: Engloba los órganos situados entre los labios y la entrada de las cuerdas vocales o glotis, esto es, la faringe y la boca o cavidad oral; además consta de la cavidad nasal, que es acoplada mediante el paladar suave (ver figura 1.1). El tracto vocal y el nasal son tubos no uniformes cuya forma varía con el tiempo. En esta región se modulan los sonidos provenientes de la laringe para producir la voz, esto es, el espectro del sonido es modificado por la selectividad en frecuencia del tracto (resonancias).

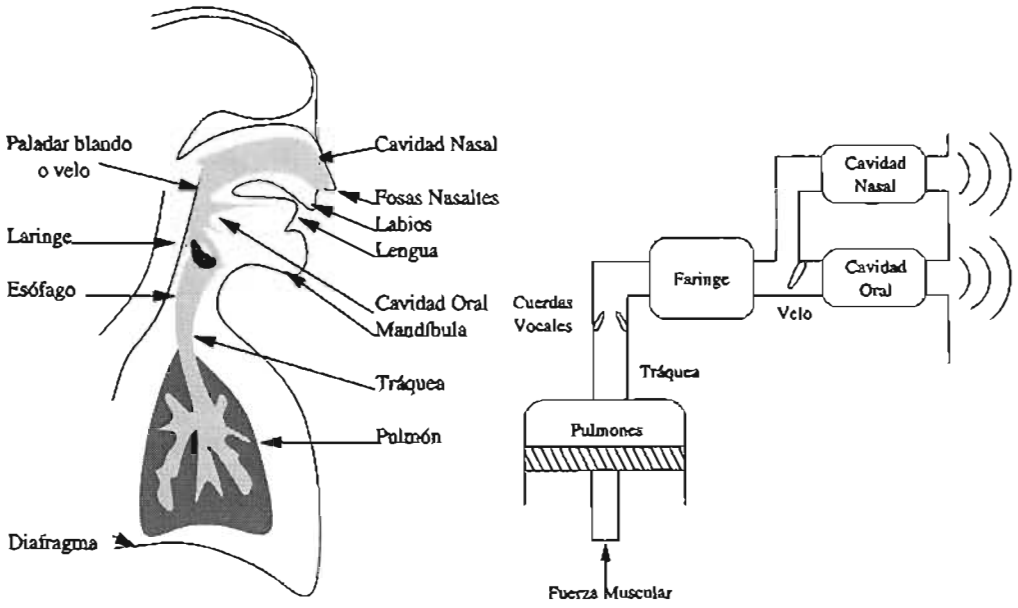


Figura 1.1: *Sistema de producción de la voz*

Los sonidos de voz pueden clasificarse en diferentes clases de acuerdo con el modo de excitación del tracto vocal:

- Sonidos sonoros: El flujo de aire proveniente de los pulmones es modulado por la vibración de las cuerdas vocales en la glotis, generando una excitación tipo pulsos cuasi-periódicos. La frecuencia de los pulsos cuasi-periódicos es llamada frecuencia fundamental y es percibida como el tono del sonido sonoro. Por ejemplo: las vocales. Típicamente la voz posee una frecuencia fundamental promedio de 132 Hz para los hombre y 223 Hz para las mujeres
- Sonidos sordos: El flujo de aire proveniente de los pulmones se torna turbulento al pasar por una constricción en algún punto del tracto vocal, generando una excitación tipo ruido. Por ejemplo: /s/, /ch/.
- Sonidos plosivos: El flujo de aire crea una presión tras un punto completamente cerrado en el tracto vocal, al liberar esta tensión de forma rápida, removiendo la constricción, se genera una excitación transitoria. Por ejemplo: /p/, /t/.
- Sonidos nasales: Se generan debido al acoplamiento acústico del tracto nasal (región entre el paladar suave o velo y las cavidades nasales) con el tracto vocal. Por ejemplo: /n/.

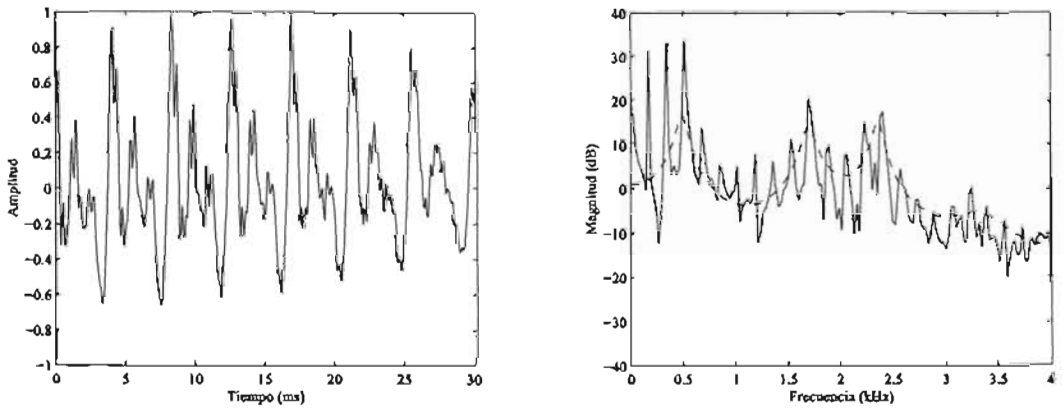
Existen otras clases de sonidos como las fricativas o las semivocales; sin embargo, las clases anteriores son las principales.

El espectro de potencia en tiempo corto de un segmento de voz sonora se caracteriza por su estructura fina y su estructura de formantes. La estructura fina es consecuencia de la cuasi-periodicidad de la voz y puede ser atribuida

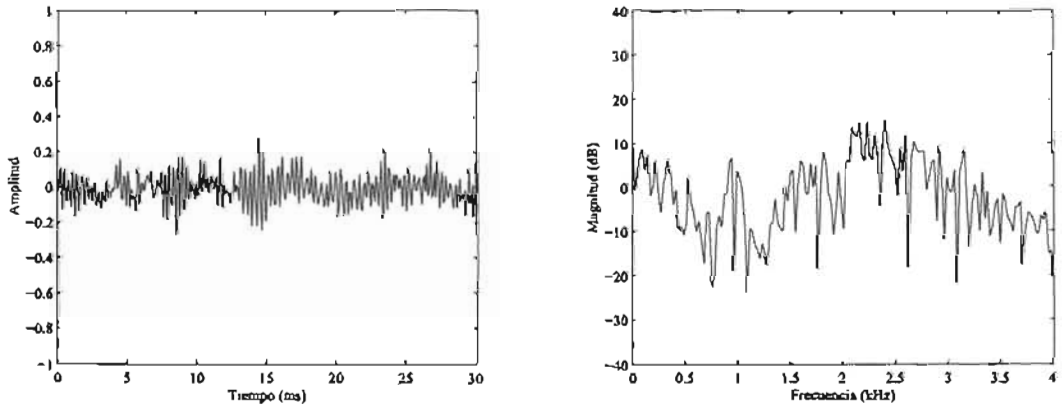
a la vibración de las cuerdas vocales. La estructura de formantes (envolvente del espectro) se debe a la interacción entre la fuente de la excitación y el tracto vocal. La forma de la envolvente que "sigue" al espectro en tiempo corto del segmento sonoro de voz, figura 1.2(a), se encuentra asociada con las características de transferencia del tracto vocal y una pendiente espectral de -6 dB/octava debida a una pendiente de -12 dB/octava debida a la forma de pulso glotal mas una pendiente de 6 dB/octava debida a la radiación de la voz por los labios.

La envolvente del espectro se caracteriza por una serie de picos llamados formantes, que corresponden a las resonancias del tracto vocal. Éstas dependen de la forma y dimensiones del tracto vocal, esto es, las propiedades espectrales de la señal de voz varían con el tiempo al variar la forma del tracto vocal.

Existen normalmente entre 3 y 5 formantes por debajo de los 5 kHz. Las amplitudes y posiciones de las primeras tres formantes, usualmente por debajo de los 3 kHz, son muy importantes tanto en la síntesis como en la percepción de la voz. Las formantes mayores son también importantes en las representaciones de banda ancha así como en los sonidos sordos.



(a) Segmento de voz sonora



(b) Segmento de voz sorda

Figura 1.2: *Segmentos de voz en el tiempo y en la frecuencia*

De lo anterior se llega a la propiedad de cuasi-estacionaridad de la voz. Se tiene que las propiedades de una señal de voz cambian con el tiempo, esto es, la voz es una señal no estacionaria; sin embargo, como el tracto vocal es un sistema mecánico, sus movimientos son relativamente lentos pues se encuentran restringidos por la masa de los

articuladores (la lengua, mandibula, labios, dientes, etc.). Por esta razón, es posible modelar al tracto vocal como un filtro acústico que varía lentamente en el tiempo, excitado por una o más señales. Por lo tanto, para la mayoría de los sonidos de voz es razonable asumir que las propiedades generales de la excitación y el tracto vocal permanecen fijas (cuasi-estacionarias) por periodos cortos, típicamente de 10-30 ms; y sus propiedades estadísticas y espectrales se definirán sobre estos segmentos cortos.

1.2. Comunicación por voz

En su forma más general, la comunicación por voz es el proceso de transmisión verbal de una idea de una persona a otra. La figura 1.3 muestra un diagrama de bloques simple del proceso de comunicación de la voz usando un sistema de codificación de voz. El objetivo de todo el sistema es transmitir correctamente un concepto de un locutor a otro por medio del canal de comunicaciones.
En este proceso, la idea o concepto es primero transformado en una sentencia, que posteriormente es tranformada en gestos musculares del tracto vocal, garganta y pulmones. El tracto vocal transforma la sentencia en una onda acústica de presión de aire que es recibida por un micrófono a la entrada del sistema de codificación de voz. La tarea del codificador de voz es digitalizar la señal de voz y representarla mediante un flujo digital de bits. La tasa de bits de este flujo, sin embargo, debe ser consistente con la capacidad de transmisión del canal. En el receptor, el decodificador de voz recibe el flujo digital de bits e intenta crear una nueva señal de voz que sea perceptualmente tan cercana a la original como sea posible. Esta señal es transmitida acústicamente al oído del escucha. Usando tanto recursos cognitivos como un entendimiento del lenguaje, el oyente entonces "entiende" la sentencia e interpreta su significado.

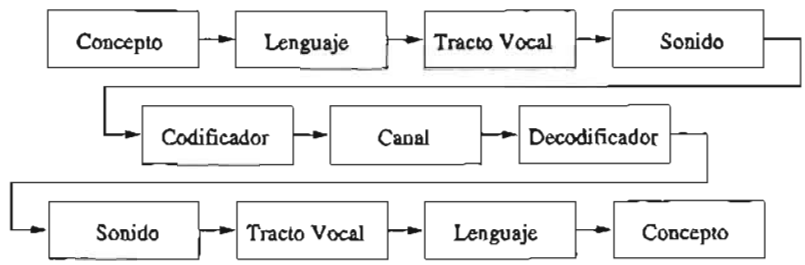


Figura 1.3: *Ambiente de operación de un codificador de voz*

Para poder representar la voz como un flujo de bits es necesario modelar alguno de los bloques del esquema de la figura 1.3. Como los modelos conceptuales y del lenguaje (bloques 1 y 2) se encuentran en estado primitivo, pues no ha sido posible entender completamente la forma en que el cerebro genera los conceptos y produce la entonación de las sentencias, se utilizan modelos del tracto vocal así como del oído.

1.2.1. Modelos digitales para la voz

La representación de las señales de voz en una forma digital es de fundamental importancia. En este aspecto, el bien conocido teorema del muestreo [1] es la base de la teoría y aplicación del procesamiento digital de voz. Existen varias posibilidades para la representación discreta de las señales de voz, en general pueden ser clasificadas en dos grupos: representaciones de la forma de onda y representaciones paramétricas. Las representaciones de la forma de onda, como su nombre lo indica, tratan simplemente de preservar la "forma de la onda" de la señal de voz analógica mediante un proceso de muestreo y cuantización. Las representaciones paramétricas, por otro lado, tratan de representar la señal de voz como la salida de un modelo para la producción de la voz.

[1] Para poder reconstruir completamente una señal limitada en banda, debe ser muestreada a una tasa de al menos la frecuencia de Nyquist ($2f_{max}$)

Modelos del tracto vocal

La figura 1.4 muestra un diagrama de bloques general, representativo de numerosos modelos que han sido usados como base del procesamiento de voz. Estos modelos tienen en común que las características de la excitación (relacionadas con la fuente de los sonidos de voz) se encuentran separadas del las del tracto vocal (relacionadas con los sonidos de voz individuales) y la radiación.

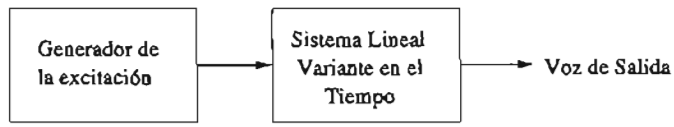


Figura 1.4: *Modelo fuente-filtro para la producción de voz*

El modelo de la figura 1.4 es llamado modelo de "terminales análogas", y es un sistema lineal cuya salida posee las propiedades de la voz deseadas al ser controlado por un conjunto de parámetros que se encuentran de alguna forma relacionados con el proceso de la producción de la voz. El modelo es por tanto equivalente al modelo físico en sus terminales (salida) pero su estructura interna no asemeja la física de la producción de la voz. Este modelo se conoce también como modelo fuente-filtro, donde la fuente es la señal acústica que es filtrada por las resonancias en las cavidades del tracto vocal.

Para producir una señal similar a la voz, el modo de excitación y las propiedades de la resonancia del sistema lineal deben cambiar con el tiempo. Por lo tanto, el modelo de terminal análoga involucra un sistema lineal lentamente variante en el tiempo (que modela los efectos de las resonancias del tracto vocal, así como la radiación por la boca) excitado por una señal cuya naturaleza básica cambie de pulsos cuasi-periódicos (voz sonora) a ruido aleatorio (voz sorda). Los cuatro tipos de excitación fueron reducidos, de manera general, a dos: señales periódicas (sonidos sonoros) y ruido turbulento (sonidos sordos)

Por lo anterior, un modelo completo de la voz debe incluir: los cambios en la señal de excitación, la respuesta del tracto vocal (filtro acústico) y los efectos de los labios en la radiación (ver figura 1.5).

Como el tracto vocal es un tubo no uniforme que varía lentamente con el tiempo, es posible modelarlo como una serie concatenada de tubos uniformes sin pérdidas (ver figura 1.6(b)), cuya función de transferencia corresponde a la de un filtro recursivo todo polos:

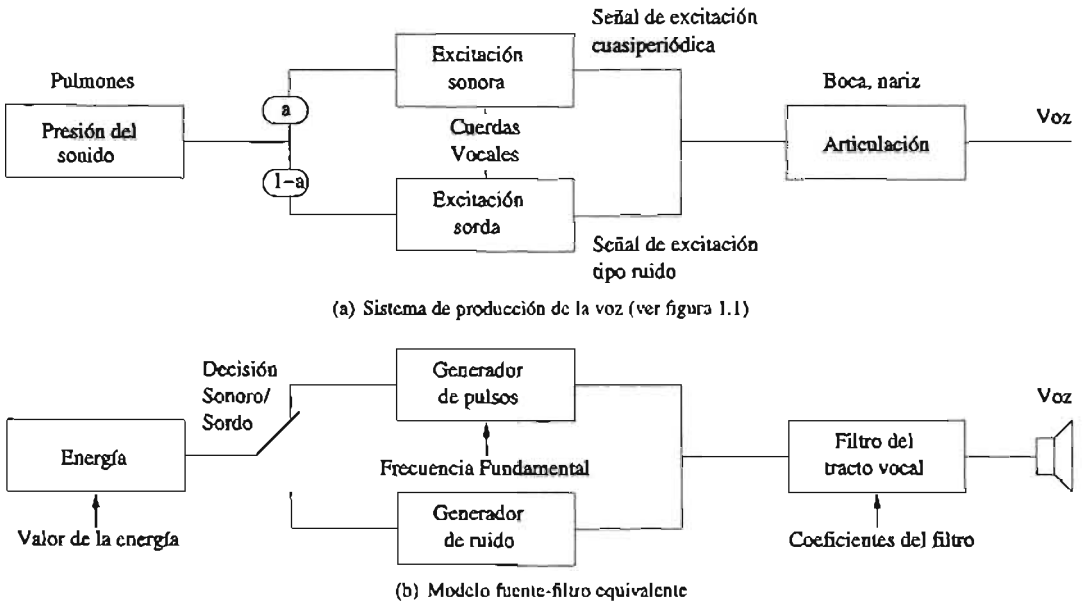
$$V(z) = \frac{G}{1 - \sum_{k=1}^{N} \alpha_k z^{-k}} \tag{1.1}$$

donde G representa el factor de ganancia total, α_k las ubicaciones de los polos y k corresponde al número de secciones o tubos uniformes, por lo que se tendrá un retraso por cada sección. Los polos de $V(z)$, que corresponden a las resonancias o formantes de la voz, pueden obtenerse mediante el análisis por predicción lineal de la voz (ver capítulo 2.2), por lo que los codificadores de voz que usan este modelo son llamados vocoders (LPC).

En este filtro paramétrico, los parámetros G y α_k variarán lentamente con el tiempo. Por lo tanto, las tasas de bits asociadas con los parámetros del filtro del tracto vocal serán menores que las de la señal de voz misma. Esto es llamado propiedad de estacionaridad en tiempo corto del filtro del tracto vocal.

En todos los codificadores de voz que usan un filtro para modelar al tracto vocal, el procedimiento es analizar la voz original en el transmisor con el fin de extraer los parámetros que controlan al filtro. Estos parámetros son codificados y transmitidos al receptor para controlar al filtro del tracto vocal y generar la señal de voz de salida.

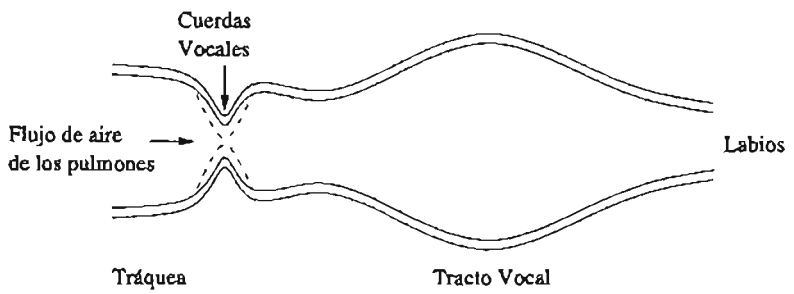
La señal de excitación también posee propiedades que pueden ser representadas paramétricamente. Recordando que la mayoría de los sonidos de voz pueden clasificarse en sonoros o sordos, en términos generales, la fuente que genere



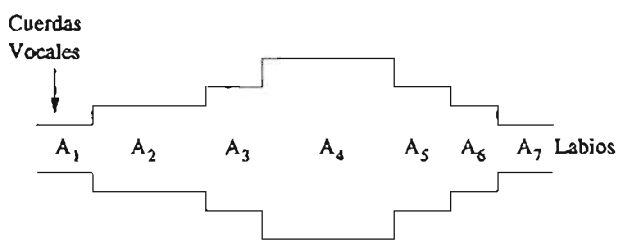
(a) Sistema de producción de la voz (ver figura 1.1)

(b) Modelo fuente-filtro equivalente

Figura 1.5: *Modelo fuente-filtro para la producción de voz II*



(a) Representación esquemática del sistema de producción de la voz



(b) Representación mediante concatenación de tubos sin pérdidas

Figura 1.6: *Representación del sistema de producción de voz*

la entrada del sistema de radiación del tracto vocal debe ser capaz de producir ya sea pulsos cuasi-periódicos o ruido aleatorio. En el caso de la voz sonora, debido a la fisiología de la glotis la frecuencia fundamental de su vibración no varía mucho con el tiempo; por tanto la excitación sonora puede ser modelada adecuadamente usando un modelo de la frecuencia fundamental que varíe lentamente. Esta señal del tono puede ser representada con una tasa de bits menor que la señal de excitación sonora. Típicamente se usa un generador de tren de impulsos para producir una secuencia de impulsos unitarios con la frecuencia fundamental deseada; esta señal excita un sistema lineal cuya respuesta al impulso, $g(n)$, tenga la forma de la onda glotal deseada; mientras que un control de ganancia, A_v controla la intensidad de la excitación sonora. Para los sonidos sordos, el modelo de la excitación es mucho más simple. Todo lo que se requiere es una fuente de ruido aleatorio y un parámetro de ganancia para controlar la intensidad de la excitación sorda.

Finalmente seleccionando entre los generadores de excitaciones sonora y sorda podemos modelar el modo cambiante de excitación (interruptor sonoro/sordo).

En resumen, los parámetros para este modelo de excitación son el periodo del tono para sonidos sonoros, la posición del interruptor sonoro/sordo, y la ganancia de la excitación.

Estos modelos son muy compactos y generalmente operan a tasas de bits muy bajas; sin embargo, poseen algunas limitaciones.

- El tracto vocal en realidad no se encuentra compuesto por cilindros, y sí experimenta pérdidas en los tejidos blandos.
- El modelo no es capaz de modelar adecuadamente los sonidos plosivos o nasales; sin embargo, sigue siendo adecuado. Esto se debe a que el modelo no considera los ceros (antiresonancias) generados en las cavidades nasales cuando la cavidad oral se encuentra cerrada.
- En general la voz producida no es muy natural, y no es posible obtener voz de alta calidad (*toll quality*), lo que significa que las señales de voz codificadas sean subjetivamente indistinguibles de las originales.
- Los procedimientos de análisis (detectores del tono) que deben ser usados para determinar los parámetros de la excitación son muy difíciles de realizar de forma efectiva para una amplia clase de locutores y condiciones.

Sin embargo estas deficiencias no limitan de forma severa la aplicabilidad del modelo.

Como alternativa al modelo anterior, se puede modelar la señal de excitación como la salida de un predictor de tono variante en el tiempo excitado por una señal codificada. Al igual que el modelo anterior, se representa al del tracto vocal como un filtro lineal paramétrico que varía lentamente. En este modelo la redundancia del tono es modelada por el predictor del tono mientras que aspectos de la señal de excitación que no pueden ser modelados por el predictor del tono se encuentran incluidos en la señal codificada. Varios codificadores de voz utilizan este modelo general, entre ellos los codificadores (RELP), (CELP), (MPLPC), (SEV) y muchos más. En general esta clase de codificadores requieren más bits para representar la señal de excitación, pero se encuentran menos restringidos por el modelo paramétrico. Además como esta clase de codificadores de voz no requieren una decisión explícita sonoro/sordo, los procedimientos de estimación de la frecuencia fundamental son más simples y efectivos que para el modelo anterior. Los codificadores que usan este modelo son buenos candidatos para la codificación de voz a baja tasa de bits y alta calidad.

1.3. Clases de codificadores de voz

El término codificador de voz (o codificador de voz de banda angosta) se refiere a los codificadores que operan sobre el ancho de banda telefónico (300-3400 Hz) con una frecuencia de muestreo de 8 kHz. Existen diferentes tipos de codificadores. En la tabla 1.1 se encuentran las cuatro principales clases de codificadores junto con las tasas de bits, complejidad y aplicaciones asociadas.

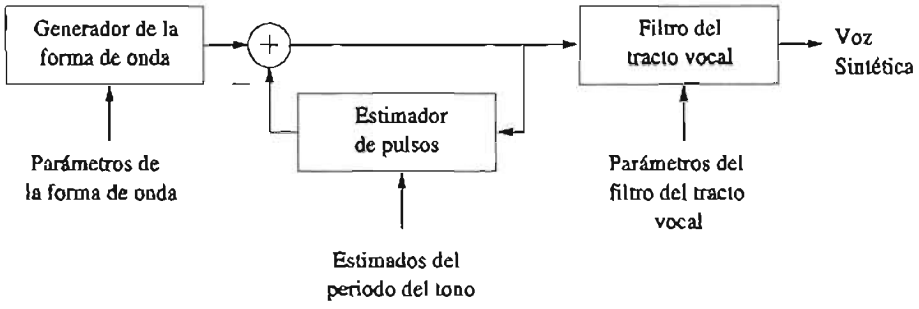


Figura 1.7: *Modelo general para la producción de voz II*

Tipo de codificador	Tasa de bits (kbps)	Complejidad	Aplicaciones
De forma de onda	16-64	Baja	Telefonía terrestre
Subbanda	12-256	Media	Teleconferencia, audio
LPC-AS	4.8-16	Alta	Telefonía celular digital
Vocoder LPC	2.0-4.8	Alta	Telefonía satelital, militar

Cuadro 1.1: *Características de las cuatro clases principales de algoritmos para la codificación de voz*

Los codificadores de forma de onda tratan de codificar la forma exacta de la señal de voz, sin considerar la naturaleza de la producción y percepción humana de la voz. Estos codificadores usan altas de bits (típicamente por encima de los 16 kbps). Son muy útiles en aplicaciones que requieren la codificación de señales tanto de voz como de otro tipo de audio. En la red telefónica pública (PSTN), por ejemplo, es muy importante la transmisión de tonos de señalización de fax y módem, tanto como la transmisión de la voz. Los algoritmos de codificación de la forma de onda más comúnmente usados son PCM uniforme de 16 bits, PCM compandido de 8 bits, y ADPCM.

Los codificadores por predicción lineal (LPC), por otro lado, asumen que la señal de voz es la salida de un sistema lineal variante en el tiempo, como se mencionó en la sección 1.2.1. Si la excitación únicamente es generada en el receptor, con base en la frecuencia fundamental transmitida e información sobre la sonoridad (interruptor sonoro/sordo) de los segmentos de voz, entonces el sistema es llamado vocoder LPC; éstos han sido adoptados como estándares de codificación para tasas de 2.0 a 4.8 kbps.

Los codificadores LPC-AS eligen la mejor función de excitación de un conjunto de varias posibles candidatas; este tipo de codificadores son usados en la mayoría de los estándares que usan tasas entre 4.8 y 16 kbps.

Los codificadores de subbandas son codificadores en el dominio de la frecuencia que tratan de parametrizar la señal de voz en términos de sus propiedades espectrales en diferentes bandas de frecuencias; éstos codificadores son menos usados que los basados en LPC.

1.4. Componentes de un codificador de voz

Los elementos primarios de un sistema digital de codificación de voz se encuentran ilustrados en la figura 1.8. La entrada del sistema es una señal de voz continua, $s(t)$. A esta señal se le aplica un filtrado paso bajas y es muestreada por un convertidor A/D, generando la señal digital de voz, $s[n]$. Ésta es la entrada del codificador de voz.

El codificador de voz generalmente consiste de tres etapas: el análisis de la voz, la cuantización de los parámetros y la codificación de los mismos. La entrada de la etapa de análisis es la señal digital de voz, mientras que la salida es la nueva representación de la señal de voz que será cuantizada y codificada. La salida de la etapa de análisis puede

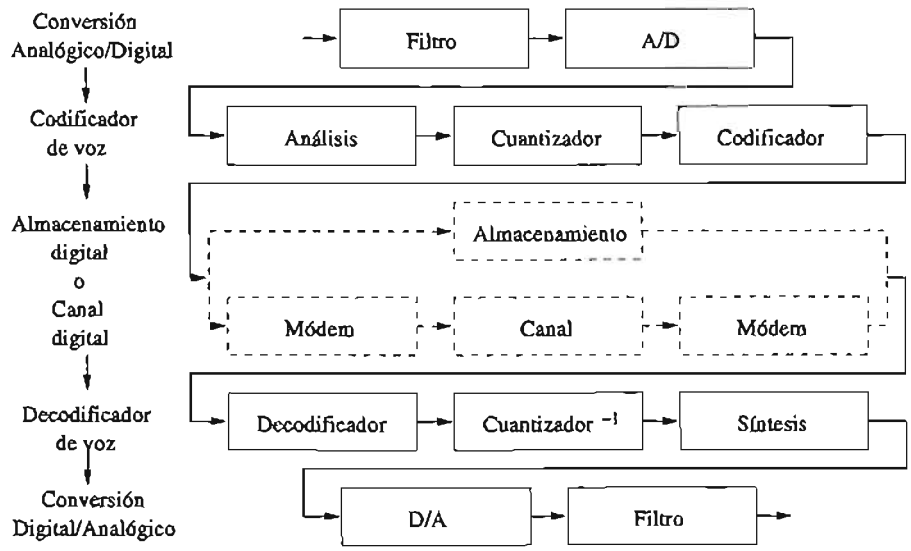


Figura 1.8: *Elementos de un sistema de codificación de voz*

variar dependiendo de como se modele la señal de voz. Para un sistema PCM, por ejemplo, no habría análisis ya que la salida simplemente sería la señal digital de voz. Para los de codificadores de forma de onda, la salida sería una versión procesada de la entrada. Para los codificadores paramétricos, la salida serían los parámetros del modelo de voz. Por tanto, para un vocoder LPC, la salida de la etapa de análisis serían los parámetros de predicción lineal del filtro del tracto vocal, el estado del interruptor sonoro/sordo, el periodo del tono, y la ganancia de excitación.

Después del análisis los parámetros deben ser cuantizados para reducir el número de bits requeridos. A la salida del cuantizador el codificador asigna un código binario único a cada posible representación cuantizada. Estos codigos binarios son empaquetados para su eficiente transmisión o alacenamiento.

La voz codificada digitalmente es a menudo usada tanto en aplicaciones de comunicación como en aplicaciones de almacenamiento y reproducción. En el caso de la aplicación en comunicaciones, el sistema debe minimizar el retraso de codificación, particularmente cuando el canal puede tener otros retrasos significativos (como en las comunicaciones por satélite). Además existe una restricción en el costo y/o potencia del sistema, particularmente en aplicaciones con un alto volumen de consumidores. Esto restringe la intensidad computacional del algoritmo de codificación de voz. Finalmente los sistemas de comunicación real frecuentemente introducen errores que deben ser detectados por el codificador de voz. Para proporcionar cierta protección contra errores se usa parte de la tasa de bits disponible, lo cual deja menos bits para ser usados por el codificador de voz mismo.

El decodificador de voz realiza las operaciones inversas al codificador. Después de que se decodifica el flujo digital de bits, es transformado en versiones cuantizadas de los parámetros de la voz mediante un cuantizador inverso. En ausencia de errores, estos son idénticos a los parámetros de salida de la etapa del cuantizador en el codificador de voz. Estos parámetros son usados para sintetizar la señal de voz codificada, $\hat{s}[n]$. El sintetizador puede ser muy simple, como en el caso de PCM. Más a menudo es bastante complejo ya que realiza todo un modelado paramétrico de la voz para el sistema. La señal digital sintética de voz, $\hat{s}[n]$, es convertida D/A e introducida en un filtro de reconstrucción para generar la señal analógica sintética de voz, $\hat{s}(t)$.

Aplicación	Tasa de bits (kbps)	Organización	Documento de referencia	Codificación	Año
Telefonía	64	ITU	G.711	PCM	1988
Terrestre	16-40	ITU	G.726	ADPCM	1990
	16-40	ITU	G.727	ADPCM	1990
Teleconferencia	48-64	ITU	G.722	ADPCM	1988
	16	ITU	G.728	LD-CELP	1992
Multimedia	5.3-6.3	ITU	G.723.1	CELP	1996
	2.0-18.2	ISO	MPEG-4	CELP	1998
Comunicaciones	2.4	DDVPC	FS1015	LPC-10e	1984
seguras	4.8	DDVPC	FS1016	CELP	1989

Cuadro 1.2: *Algunos estándares para la codificación de voz*

1.5. Estándares de la codificación de voz

Los estándares de la codificación de voz tienen un papel importante en el desarrollo y uso de los codificadores de voz. Para la gran mayoría de las aplicaciones, la interoperabilidad es un asunto importante. Para lograr la interoperabilidad deben definirse e implementarse estándares. En la tabla 1.2 se muestran algunos estándares para la codificación de voz.

Los estándares pueden y son desarrollados por varias organizaciones. Quizás el estándar más simple aunque no es específicamente un estándar de codificación de voz, es el formato usado por los CD's digitales. Tales sistemas usan una codificación lineal PCM de 16 bits. Este formato no usa características específicas de la voz, y por tanto es apropiado para codificar tanto voz como otras señales de audio.

El Departamento de Defensa de los Estados Unidos (DDVPC) publica estándares de codificación de voz para aplicaciones del gobierno estadounidense. Dos estándares de codificación de voz creados por el DDVPC muy importantes, son los estándares federales FS 1015 y 1016. El primero de éstos llamado comúnmente LPC10e, es un vocoder LPC que opera a 2.4 kbps. El segundo es un codificador LPC-AS excitado por código (CELP) que opera a 4.8 kbps.

Otro organismo importante es la ITU, sucesor de la CCITT. Este comité define estándares para la red telefónica internacional. Existen varios estándares ITU importantes actualmente en uso. El más usado es el PCM de 64 kbps encontrado en las aplicaciones de conmutación digital. Éste incluye PCM ley μ para Norteamérica y PCM ley A para Europa (ITU G.711). Otros estándares importantes para telefonía son el estándar ADPCM (G.726) que opera a 16, 24, 32 y 40 kbps, LD-CELP (G.728) que opera a 16 kbps, ACELP (G.723.1) que opera a 5.3 kpbs y CS-ACELP (G.729) a 8 kbps; algunos de éstos usados en aplicaciones de voz sobre IP. La ITU es una rama de la ISO. Además de las actividades realizadas por la ITU, la ISO desarrolla los estándares MPEG.

El área de estándarización más activa a la fecha involucra los estándares para telefonía celular digital (ver tabla 1.3).

En Europa, la organización de estándares es el subcomité GSM del ETSI. Todos sus estándares para telefonía celular digital se basan en algoritmos LPC-AS. El primer estándar GSM se basaba en un precursor de CELP llamado RPE-LTP, mientras que los estándares GSM más recientes usan ACELP.

En norteamérica, la organización de estándares es la TIA, y ha adoptado un estándar basado en VSELP, que es una forma de CELP. La organización de estándares japonesa, JDC, ha adoptado un estándar similar.

1.6. Tasa de bits y Calidad

Un algoritmo de codificación de voz es evaluado en base a su tasa de bits, la calidad de la voz reconstruida (codificada), la complejidad del algoritmo, el retardo introducido y la robustez del algoritmo contra los errores del canal

	Sistema	Origen	Codificación	Velocidad (kbps)	MOS	Documento de referencia	Año
TDMA	GSM/DCS/PCS FR	ETSI	RPE-LTP	13	3.6-3.8	ANSI J-STD-007	1987
						ETSI/ETS 300580	1994
	GSM/DCS/PCS HR	ETSI	VSELP	5.6	3.5-3.7	ETSI/ETS 300581	1995
	GSM EFR	ETSI	ACELP	12.2	4.1	ANSI J-STD-007A	1998
	US PCS 1900 EFR	EUA				ETSI/ETS 300726	1997
	North-American DMR	EUA	VSELP	7.95	3.5	EIA/TIA IS-54	1990
	D-AMPS FR	EUA	VSELP	8.0	3.7	TIA/EIA IS-85	1992
	D-AMPS Ampliado	EUA	ACELP	7.4	4.1	TIA/EIA IS-641	1996
	PDC FR	Japón	VSELP	6.7	3.4	RCR STD-27	1990
	PDC HR	Japón	PSI-CELP	3.45	3.34	RCR STD-27	1993
CDMA	CDMA (IS-96)	EUA	CELP	8/4/2/0.8	3.3	TIA/EIA IS-96	1994
	CDMA (IS-127)	EUA	RCELP	8/4/0.8	4.1	TIA/EIA IS-127	1997
	CDMA DMR	EUA	QCELP	0.8-8.55	3.4	TIA/EIA IS-95	1993
	UMTS	ETSI	ACELP	4.75-12.2		ETSI/ETS 301703	1998

Cuadro 1.3: *Sistemas de codificación de la voz utilizados en el servicio de telefonía celular digital*

y la interferencia acústica. Un codificador ideal tendría una baja tasa de bits, alta calidad perceptible, bajo retardo, baja complejidad y alta robustez contra errores de transmisión. Sin embargo, en la práctica deben compensarse estos factores dependiendo de los requerimientos de la aplicación. Por ejemplo, la codificación de voz de alta calidad a baja tasas de bits se logra usando algoritmos de alta complejidad.

El objetivo del diseño de un codificador de voz, generalmente, es producir la mayor calidad de voz posible a la menor cantidad de bits posible.

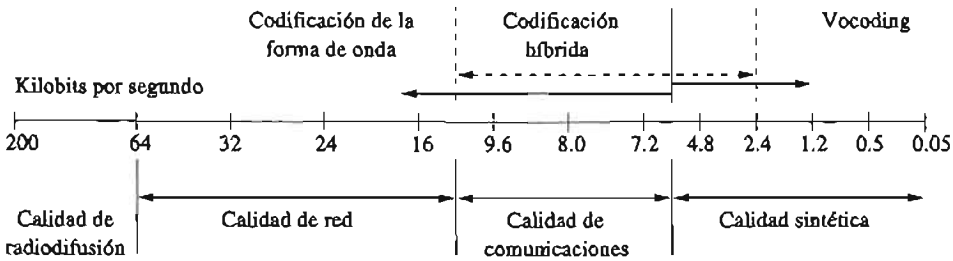
En las comunicaciones digitales, la calidad de la voz es clasificada en cuatro categorías: de radiodifusión, de red (o *toll quality*), de comunicaciones y sintética. La calidad de radiodifusión de banda ancha se refiere a la alta calidad de voz que puede obtenerse a tasas de bits por encima de los 64 kpbs. La calidad de red o *toll quality* se refiere a una calidad de voz comparable a la escuchada a través de la red telefónica conmutada (300-3400 Hz) y puede obtenerse a tasas de bits por encima de los 10 kbps. La calidad de comunicaciones implica una cierta degradación en la calidad de la voz que sin embargo se escucha natural, altamente inteligible, y adecuada para las telecomunicaciones. La voz sintética es usualmente inteligible pero puede no ser natural (tipo máquina) y se encuentra asociada con una pérdida en el reconocimiento del locutor. La voz para comunicaciones se puede obtener a tasas de bits por encima de los 4.8 kbps, mientras que los codificadores que operan a tasas por debajo de los 4 kpbs tienden a producir calidad sintética.

La figura 1.9(a) muestra las diferentes tasas de bits en una escala unidimensional, y una designación aproximada de la calidad de la voz que puede obtenerse a diferentes tasas de bits. La figura 1.9(b) presenta información similar en un diagrama bidimensional, donde se puede observar cómo se compara la calidad de las diferentes técnicas de codificación de voz con las tasa de bits. Por supuesto el diagrama es cualitativo pero proporciona una idea de la calidad esperada para cada sistema.

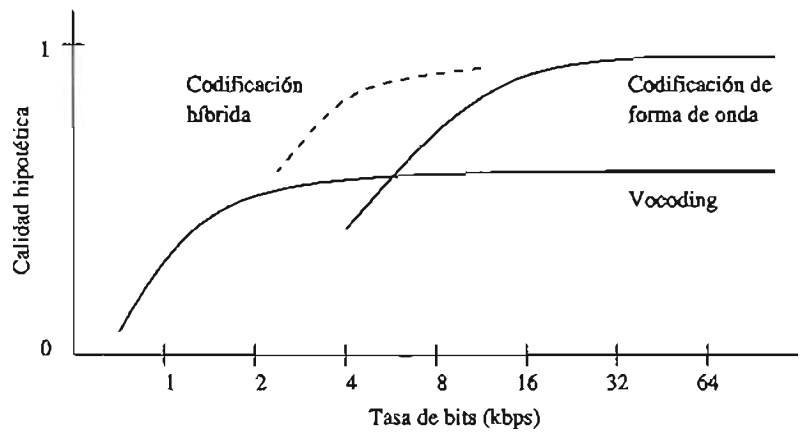
Como se esperaría, para lograr la mejor calidad posible reduciendo la tasa de bits, es necesario usar algoritmos más sofisticados. La "sofisticación" en este contexto, implica programas computacionales más largos, mayor carga computacional, y consecuentemente, mayor tiempo de ejecución.

La compensación entre tasa de bits, calidad y complejidad de los algoritmos dependen del diseñador del sistema. Las preguntas a realizarse son:

- ¿Qué nivel de pérdidas en la calidad de la voz es aceptable?
- ¿Cuáles son las restricciones en la capacidad de almacenamiento o transmisión en bits por segundo (bits/s)?



(a) Espectro de las tasas de transmisión de las codificaciones de voz en una escala no lineal y la calidad asociada



(b) Calidad vs tasa de bits para las técnicas de codificación de voz

Figura 1.9: *Tasas de bits vs Calidad*

- ¿Cuál es la capacidad en tiempo real, área requerida y consumo de potencia del hardware disponible?
- ¿Cuál es el costo aceptable del sistema?

Para las aplicaciones comerciales, la última pregunta es usualmente el factor determinante en la elección final. En la mayoría de los casos no existe una única respuesta a estas preguntas. La experiencia del diseñador así como las restricciones dependientes de la aplicación sugerirán la solución.

1.7. Medidas del funcionamiento de un codificador. Herramientas de desarrollo

En el desarrollo e implementación de los algoritmos de codificación de voz resulta necesario evaluar los pasos sucesivos del desarrollo. Las deficiencias de los algoritmos o errores de programación pueden ser identificados y corregidos usando algunas herramientas visuales y auditivas para el procesamiento de la voz.

Es importante poder observar el comportamiento tanto en el tiempo como en la frecuencia de la señal de voz. Examinando la forma de onda podemos obtener información acerca de la periodicidad de la señal, la que depende de su naturaleza sonora o sorda. El correspondiente espectro muestra el contenido en frecuencia de la señal de voz. Esta información incluye las resonancias (formantes) del tracto vocal y el posible contenido armónico.

Una herramienta muy útil en el procesamiento de voz es el espectrograma. Éste proporciona información muy importante, ya que muestra la dinámica en el tiempo del espectro en frecuencia. Las secciones de mayor amplitud muestran el movimiento de las frecuencias naturales (resonancias) del tracto vocal humano. Estos movimientos, junto con el contenido en frecuencia, son característicos de los diferentes elementos del lenguaje, como las vocales y consonantes.

Además de observar las representaciones en tiempo y frecuencia de las señales de voz, requerimos herramientas que evalúen la calidad de la voz codificada, así como el funcionamiento relativo de los diferentes sistemas. Sin embargo, no ha sido posible cuantificar en una expresión matemática qué significa "calidad de voz". Esto se debe a que no se ha comprendido completamente cómo el oído y el cerebro humano procesan la señal de voz. Además, no existe una definición que no sea ambigua de lo que es una "buena calidad de voz". Por ejemplo, teniendo dos sistemas que posean un funcionamiento similar pero introduzcan diferentes tipos de distorsión, una persona puede preferir el sistema A, mientras que otra podría preferir el B. Además el oído/cerebro humano experimentan un efecto de "entrenamiento". Este efecto puede ocasionar que se acepte una voz, después de oirla repetidamente, que anteriormente se consideraba inaceptable. A pesar de estas limitaciones, aún requerimos de un medio para evaluar la calidad de la voz.

Las herramientas auditivas son de gran importancia en la codificación de la voz ya que lo que se "ve" bien no siempre se "escucha" bien. El escuchar la voz codificada puede producir una mejor evaluación de la calidad que cualquier medida objetiva que use expresiones matemáticas. Sin embargo, para realizar las pruebas subjetivas de forma correcta, requerimos las facilidades apropiadas y, usualmente, grupos de personas. Esto hace que las pruebas subjetivas sean bastante costosas y sugiere que las medidas objetivas pueden ser usadas para auxiliar la comparación.

La evaluación de un codificador consiste en tres pasos: mientras el sistema está siendo desarrollado, una medida objetiva adecuada determinará qué tan bien funciona, especialmente si se compara con otro sistema conocido; el escuchar los resultados de forma informal puede también proporcionar información valiosa. Estas dos primeras etapas no se realizan de forma separada, sino entrelazada. Una vez que se ha decidido que se tiene un buen sistema, es necesario formalizar la evaluación de la calidad mediante una prueba subjetiva, preferentemente de forma que se permita la comparación con otros sistemas conocidos.

1.7.1. Medidas objetivas de la calidad de la voz

El término "medidas objetivas" se refiere a las expresiones matemáticas que son usadas para determinar la calidad de la voz. La relación señal a ruido (SNR) es una de las medidas objetivas más comunes en la evaluación del funcionamiento de un algoritmo de codificación. El SNR se define como la relación entre la energía de la señal, σ_s^2, y la energía del error de cuantización, σ_q^2, por lo que, para señales con media cero, está definido como:

$$SNR = 10log_{10}\frac{\sigma_s^2}{\sigma_q^2} = 10log_{10}\left[\frac{\sum_{n=0}^{M-1} s^2(n)}{\sum_{n=0}^{M-1}\left(s(n)-\hat{s}(n)\right)^2}\right] \tag{1.2}$$

donde $s(n)$ es la voz original, $\hat{s}(n)$ es la voz codificada y M el número de puntos de la señal. El SNR es una medida del error promedio sobre toda la señal de voz y usualmente se encuentra dominado por las porciones de alta energía de la señal. Una medida alternativa del error de codificación es el SEGSNR, que se define como el promedio de los valores SNR (dB) obtenidos para segmentos de N-puntos de la señal.

$$SEGSNR = \frac{1}{L}\sum_{i=0}^{L-1} 10log_{10}\left[\frac{\sum_{n=0}^{N-1} s^2(iN+n)}{\sum_{n=0}^{N-1}\left(s(iN+n)-\hat{s}(iN+n)\right)^2}\right] \tag{1.3}$$

Como los errores en los segmentos de baja y alta energía se calculan de manera separada, la calidad percibida es reflejada de mejor forma por el SEGSNR que por el SNR. El tamaño del segmento se elige usualmente en el orden de la duración de una sílaba (16-20 ms).

Las medidas objetivas son a menudo sensibles tanto a variaciones en la ganancia como al retardo. Además típicamente no toman en cuenta las propiedades perceptuales del oído. Se requieren por tanto evaluaciones subjetivas pues el diseño de la mayoría de los algoritmos de bajas tasas de bits se basan en criterios perceptuales. Sin embargo, como no pueden desarrollarse pruebas subjetivas para evaluar cada ajuste en un codificador, las pruebas objetivas son ampliamente usadas.

1.7.2. Medidas subjetivas de la calidad de la voz

La calidad de la voz es una medida subjetiva de cómo los usuarios individuales perciben la calidad y facilidad para establecer una conversación.

El estándar más usado para evaluar subjetivamente la calidad de síntesis de la voz se encuentra especificado en la recomendación ITU P.800 y es conocido como MOS. Fue normalizado a principio de los años 80 y se le ha utilizado principalmente para medir la calidad en sistemas de comunicación celular digital.

El test consiste en realizar una encuesta de opinión a un conjunto de individuos de prueba los cuales deben evaluar una grabación de voz. A partir de esta encuesta se obtiene una nota de opinión media de la calidad de la voz tras haber recolectado notas entre 1 (malo) y 5 (excelente), donde una calidad excelente implica que la voz codificada es indistinguible de la original y no se percibe ruido, mientras que una calidad mala implica la presencia de mucho ruido y distorsión en la voz codificada. En cada experimento subjetivo, las calificaciones MOS pueden diferir, dependiendo del diseño del experimento, el rango de condiciones incluídas en el estudio, etc.

Calidad de la señal vocal	Nota
Excelente	5
Buena	4
Regular	3
Mediocre	2
Mala	1

Cuadro 1.4: *Escala de calidad de escucha. MOS*

Pruebas de opinión sobre la escucha

El método de prueba recomendado para pruebas de escucha es el de determinación de "índices de categorías absolutas" (ACR). Los índices de categoría se aplican a grupos breves de frases sin relación. Este método es bien conocido y se ha aplicado a conexiones telefónicas analógicas y digitales y a dispositivos de telecomunicación como los códecs digitales. Por ejemplo, en los trabajos realizados en las Recomendaciones G.726, G.728, G.729 y G.722, los laboratorios de varios países llevaron a cabo pruebas subjetivas utilizando el mismo método en condiciones físicas semejantes y con sistemas de transmisión idénticos, y los resultados mostraron un alto grado de coherencia.

Las pruebas de escucha tienen aplicación directa en la evaluación de los sistemas de transmisión física que son esencialmente unidireccionales, entre los que se encuentran los circuitos de radiodifusión, los sistemas de avisos públicos y los de anuncios grabados, donde puede haber degradaciones de la escucha tales como atenuaciones, ruido y distorsión.

Con ciertas reservas pueden aplicarse los resultados de las pruebas de escucha a la evaluación de las conversaciones efectuadas a través de un sistema bidireccional, por ejemplo, la conexión a una red telefónica pública conmutada.

Material de conversación El material de conversación debe consistir en frases sencillas, breves y con significado, elegidas al azar y fáciles de entender (por ejemplo extraídas de publicaciones no técnicas o de periódicos). Con estas frases deben confeccionarse listas de forma aleatoria, de tal manera que no haya ninguna conexión evidente entre el significado de una frase y la siguiente. Deben evitarse las frases muy cortas y muy largas.

Oyentes Los participantes en las pruebas de escucha se escogen al azar entre la población que utiliza normalmente el servicio telefónico, fijando las siguientes condiciones:

1. que no hayan participado directamente en trabajos relacionados con la evaluación de la calidad de transmisión de los circuitos telefónicos o tareas afines, como codificación de la palabra;
2. que no hayan participado en pruebas subjetivas, de cualquier naturaleza, al menos durante los seis meses precedentes, ni en pruebas de opinión sobre la escucha al menos durante un año; y
3. que no hayan oído antes la misma lista de frases.

Por supuesto los oyentes deben estar familiarizados con las condiciones así como el rango con que deben evaluar el material escuchado.

Escala de opinión recomendada por el UIT-T En la tabla 1.4 se muestra la escala de opinión utilizada con más frecuencia por la ITU-T.

La magnitud evaluada a partir de las notas (nota media de opinión sobre la calidad de escucha) se representa por el símbolo MOS.

El rango MOS se relaciona con la calidad de la voz de la forma siguiente: una nota MOS 4–4.5 implica calidad de red o *toll quality*, 3.5-4 implica calidad de comunicaciones, 2.5-3.5 implica calidad sintética.

Las evaluaciones MOS pueden variar considerablemente dependiendo de diversos factores como: el género del oyente, el lenguaje y la locación del oyente. Por estas razones, una prueba MOS debe evaluar varios codificadores en paralelo bajo las mismas condiciones de prueba. Si la prueba MOS se realiza cuidadosamente, una diferencia de 0.15 entre codificadores puede ser considerada significativa.

CAPÍTULO 2

Algunas codificaciones básicas

2.1. Codificadores Diferenciales de Forma de Onda

Los codificadores diferenciales (DPCM) forman parte de la clase de codificadores de forma de onda, y como tales tratan de codificar la forma exacta de la señal de voz.

Examinando de cerca las señales de voz, especialmente durante periodos sonoros, observamos que existe un cambio relativamente suave de una muestra de voz a la siguiente. En otras palabras, existe una correlación considerable entre muestras adyacentes, de hecho la correlación es significativa incluso entre muestras que se encuentran separadas por varios intervalos de muestreo. Como resultado, se esperaría que la diferencia entre muestras adyacentes tenga una variancia y rango dinámico menores que las muestras de la voz mismas. Tomando esta idea, en lugar de la señal de voz $s[n]$, la entrada del codificador puede ser la diferencia cuantizada $\hat{d}[n]$.

La codificación diferencial se refiere entonces a la codificación de la diferencia entre dos señales en lugar de las señales mismas. Con esto se remueve gran parte de la redundancia en tiempo corto de la forma de onda de la voz. Esto se logra formando la señal de error o señal diferencia sustrayendo un estimado a la señal original. El estimado se obtiene generalmente mediante un predictor lineal que estima las muestras de voz actuales a partir de una combinación lineal de una o más muestras pasadas.

Como resultado, $d[n]$ posee un rango dinámico menor que la señal de entrada del cuantizador $s[n]$, esto es, la señal diferencia $d[n]$ es generalmente menor en amplitud que la señal original. Como el ruido de cuantización es proporcional al tamaño del paso del cuantizador, una señal con menor rango dinámico puede ser codificada de manera más precisa usando un número dado de niveles de cuantización.

La forma más simple de un codificador diferencial es la que usa un predictor fijo y un cuantizador uniforme también fijo. Los DPCMs más sofisticados incluyen sistemas con cuantizadores adaptables (ADPCM[1]), predictores adaptables (APC), o ambos.

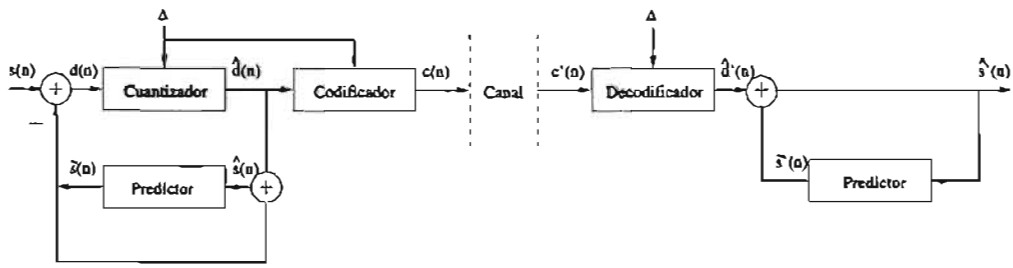


Figura 2.1: *Diagrama de bloques de DPCM*

2.1.1. DPCM

La figura 2.1 muestra el diagrama de bloques del DPCM básico, que consta de un predictor fijo y un cuantizador uniforme fijo.

Los codificadores DPCM usan una técnica que ha sido usada por otros tipos de codificadores: dentro del codificador se incluye un decodificador, de forma que la señal reconstruida $\hat{s}(n)$ es conocida en el mismo codificador. Usando $\hat{s}(n)$ para crear el estimado $\tilde{s}(n)$, se evita la amplificación del error de cuantización, pues es intuitivamente claro que mientras más precisa sea la señal estimada $\tilde{s}[n]$, menor será la variancia de $d[n]$, y mejor será el funcionamiento del cuantizador.

En la figura 2.1, $\tilde{s}[n]$ denota el estimado de $s[n]$ y es obtenido a partir de la combinación lineal de las muestras previamente codificadas como

$$\tilde{s}[n] = \sum_{i=1}^{P} a_i \hat{s}[n-i] \tag{2.1}$$

donde a_i, $i = 1, ..., P$ son los coeficientes del predictor lineal $A(z)$, y $\hat{s}[n]$ son las muestras de voz previamente codificadas. El predictor lineal, $A(z)$, se define mediante

$$A(z) = \sum_{i=1}^{P} a_i z^{-i} \tag{2.2}$$

donde P es el orden del predictor. Los coeficientes del predictor, a_i, se calculan usualmente mediante un análisis por predicción lineal de la voz (ver sección 2.2).

Siguiendo con el esquema de la figura 2.1 tenemos que la señal de error de predicción, $d[n]$, está dada por (ver figura 2.2)

$$d[n] = s[n] - \tilde{s}[n] \tag{2.3}$$

que es la cantidad que será cuantizada y codificada.

La señal de error cuantizada puede ser representada como

$$\hat{d}[n] = Q(d[n]) = d[n] + q_d[n] \tag{2.4}$$

[1]Mientras que el término ADPCM se usa a menudo en la literatura para referirse a codificadores con predictores adaptables como el estándar de codificación de voz de la ITU G.726, en el presente texto se refiere a los codificadores con cuantizadores adaptables y predictores fijos

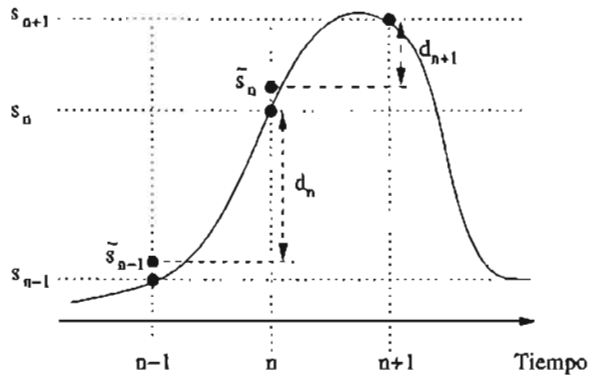


Figura 2.2: *Obtención de la señal de error de predicción en DPCM*

donde $q_d[n]$ es el error de cuantización. Como se muestra en la figura 2.1, la entrada del predictor es

$$\hat{s}[n] = \hat{d}[n] + \tilde{s}[n] \tag{2.5}$$

Combinando las ecuaciones 2.3, 2.4 y 2.5, $\hat{s}[n]$ puede escribirse como

$$\hat{s}[n] = s[n] + q_d[n] \tag{2.6}$$

que es la señal cuantizada. Esto muestra que el error de cuantización de la señal de voz $\hat{s}[n] - s[n]$ en este sistema es igual al error de cuantización de la señal de error de predicción $\hat{d}[n] - d[n]$. Esto es significativo ya que $d[n]$ posee un rango dinámico (variancia) menor que la señal de voz, y consecuentemente si se cuantiza la señal diferencia se incurrirá en un error de cuantización menor que si se cuantizara la señal original.

Debe notarse que es la señal de diferencia cuantizada la que será codificada para su transmisión o almacenamiento. El sistema para la reconstrucción de la entrada cuantizada a partir de las palabras de código se encuentra en la figura 2.1(b), consiste en un decodificador para reconstruir la señal de diferencia cuantizada a partir de la cual la entrada cuantizada es reconstruida usando el mismo predictor que el usado en la figura 2.1(a). Claramente, si $c'[n]$ es idéntica a $c[n]$ entonces $\hat{s}'[n] = \hat{s}[n]$, que difiere de $s[n]$ únicamente en el error de cuantización incurrido al cuantizar $d[n]$.

La cuantización de la señal de error de predicción puede ser realizada de varias formas. El DPCM básico utiliza un cuantizador uniforme fijo, cuyos parámetros son elegidos de acuerdo al rango dinámico y distribución de $d[n]$. La tabla 2.1 muestra los parámetros del DPCM básico, incluyendo los parámetros del cuantizador uniforme y el predictor fijo.

Parámetros	Nombre	Rango	Valor Típico
Orden del predictor	P	1-10	1
Número de bits	B	1-16	6
Tipo de cuantización	-	-	*mid-tread* o *mid-riser*

Cuadro 2.1: *Parámetros de DPCM*

Los sistemas DPCM con predictores fijos pueden proporcionar mejoras de 4 a 11 dB sobre la cuantización directa (PCM). La mayor mejora ocurre al pasar de no predicción a una predicción de primer orden con algunas ganancias adicionales resultado del incremento del orden del predictor hasta 4 ó 5, después del cual se obtienen ganancias

adicionales muy pequeñas. Esta ganancia en el SNR implica que un sistema DPCM puede lograr un SNR dado usando un bit menos del que se requeriría usando el mismo cuantizador directamente sobre la señal de voz. Por ejemplo, para un sistema DPCM con cuantizador uniforme fijo, el SNR sería aproximadamente 6 dB mayor que el SNR para un cuantizador con el mismo número de niveles actuando directamente sobre la entrada. El esquema diferencial se comportaría aproximadamente de la misma manera que el esquema PCM directo; es decir, el SNR se incrementaría 6 dB por cada bit añadido a las palabras de código. Similarmente, el SNR de un cuantizador de ley μ mejoraría 6dB una configuración diferencial y al mismo tiempo su insensibilidad característica con respecto al nivel de la señal se mantendría.

Existe una amplia variación de la ganancia de predicción dependiendo del locutor y el ancho de banda, así como entre diferentes locuciones. Todos estos efectos son resultado, por supuesto, de la no estacionaridad de la señal de voz.

Esta variación del funcionamiento con el locutor y material de voz, junto con las variaciones en el nivel de la señal inherentes al proceso de comunicación de la voz, hacen necesarias la predicción y cuantización adaptables con el fin de lograr un mejor desempeño sobre un amplio rango de locutores y situaciones de locución. Dichos sistemas son llamados sistemas PCM diferenciales adaptables.

2.1.2. ADPCM

El sistema DPCM usa un predictor y cuantizador fijos. Sin embargo, se puede ganar mucho adaptando el sistema para que siga el comportamiento en el tiempo de la señal de voz de entrada. La adaptación puede ser realizada en el cuantizador, en el predictor, o en ambos. El sistema resultante es llamado ADPCM.

DPCM con cuantización fija provee una mejora promedio de 6 dB en el SNR comparado con PCM con el mismo número de bits/muestra. Los cuantizadores adaptables pueden ser usados para obtener mayores mejoras en la operación del codificador.

En la adaptación del cuantizador distinguimos dos casos: con alimentación hacia delante (*feedforward*) y con realimentación (*feebdack*). Las figuras 2.3 y 2.4 muestran los diagramas de bloques de ADPCM-FF y ADPCM-FB.

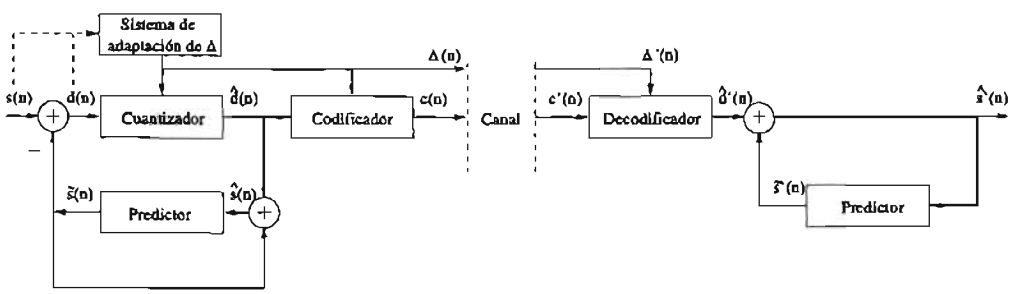


Figura 2.3: *Diagrama de bloques de ADPCM-FF*

La figura 2.3 muestra un sistema ADPCM-FF. En los esquemas de este tipo se usa la varianza de la señal de entrada para estimar los parámetros de la adaptación.

Una forma de usar la varianza es adaptando el tamaño del paso del cuantizador haciéndolo proporcional a la varianza de la entrada del cuantizador. Sin embargo, como la señal diferencia $d[n]$ será proporcional a la entrada, es razonable controlar el tamaño del paso ya sea a partir de $d[n]$ o, como se muestra en la figura 2.3, a partir de la entrada, $s[n]$.

La variancia de la señal puede ser estimada calculando la energía en tiempo corto de la señal usando una ventana rectangular de longitud M

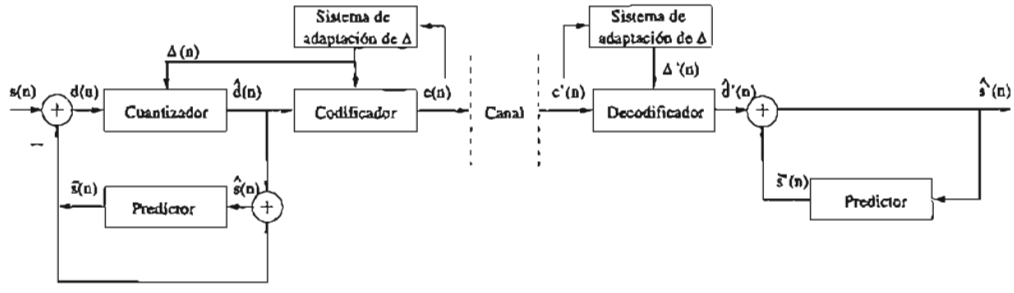


Figura 2.4: *Diagrama de bloques de ADPCM-FB*

$$\sigma^2(n) = \frac{1}{M} \sum_{m=n}^{n+M-1} s^2(m) \tag{2.7}$$

El tamaño del paso en el tiempo n entonces está dado por

$$\Delta(n) = \Delta \times \sigma(n) \tag{2.8}$$

donde $\sigma^2(n)$ es la variancia de la señal, y Δ es el valor elegido para una variancia unitaria.

$\Delta(n)$ se restringe al rango $\Delta_{min} \leq \Delta(n) \leq \Delta_{max}$. El valor de Δ_{min} se elige lo suficientemente pequeño de forma que se minimice el ruido del canal, mientras que $\Delta(max)$ se elige lo suficientemente grande para que se minimice el recorte en el cuantizador. La relación $\Delta_{max}/\Delta_{min}$ determina el rango dinámico del sistema. Usualmente se elige $\Delta_{max}/\Delta_{min} = 100$.

Para reconstruir correctamente la señal, la lógica de adaptación del decodificador debe ser similar a la del codificador. Como el decodificador no posee información sobre la adaptación de la voz cuantizada, es necesario transmitir información adicional sobre del tamaño del paso, $\Delta(n)$. En estos sistemas, la adaptación usualmente se realiza por sílaba (16-20 ms) ya que se deben transmitir los parámetros una vez que son actualizados. La información adicional que debe transmitirse representa usualmente un pequeño porcentaje (alrededor del 1 %) de la tasa total de bits.

Otra forma de usar la varianza es en una adaptación de ganancia, en la que la señal es escalada por un factor de ganancia variante en el tiempo, $G(n)$, antes de ser cuantizada, con el fin de obtener un rango uniforme. El cuantizador es fijo y sus características se eligen de forma que se aproximen a las características de la señal escalada. La ganancia se elige inversamente proporcional a la varianza de la señal, esto es:

$$G(n) = \frac{G_0}{\sigma(n)} \tag{2.9}$$

donde G_0 es una constante igual a la ganancia para una varianza unitaria. En esta ecuación, una señal de baja energía tendrá una ganancia alta y una señal de alta energía tendrá una ganancia pequeña. Esto produce una señal con un rango relativamente uniforme que es más apropiada para un cuantizador fijo que la señal original. Para prevenir un sobreescalamiento de la señal, la variación de la ganancia se limita usualmente de forma que $G_{min} \leq G(n) \leq G_{max}$. La relación G_{max}/G_{min} controla el rango dinámico de la señal escalada.

La figura 2.4 muestra un sistema ADPCM-FB. En este esquema, la variancia es estimada a partir de $\hat{d}[n]$, las muestras cuantizadas de $d[n]$. Para este caso la información adicional sobre el tamaño del paso no es necesaria ya que los cálculos están basados en la señal codificada.

El tamaño del paso es adaptado de acuerdo con la regla

Bits del Cuantizador	Multiplicador P
2	0.6, 2.2
3	0.85, 1, 1, 1.5
4	0.8, 0.8, 0.8, 0.8, 1.2, 1.6, 2.0, 2.4
5	0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6

Cuadro 2.2: *Multiplicadores P del tamaño del paso del cuantizador para ADPCM-FB y diferentes tamaños del cuantizador*

$$\Delta(n) = P\Delta(n-1) \tag{2.10}$$

El valor del multiplicador P depende únicamente del valor de $|c(n-1)|$, es decir, la magnitud de la palabra de código en el instante de tiempo anterior. La tabla 2.2 proporciona algunos valores típicos de P para cuantizadores tipo midriser. Como ejemplo, si el cuantizador es de 2 bits (4 niveles) y $c(n-1)$ corresponde al nivel de cuantización (positivo o negativo) más cercano a cero, entonces $P = 0{,}6$. Si $c(n-1)$ fuera la palabra de código correspondiente a los niveles (positivos o negativos) más lejanos a cero, $P = 2{,}2$.

En cualquier caso la adaptación del cuantizador provee un rango dinámico y un SNR mejorados. La principal ventaja del control de realimentación es que la información del tamaño del paso se deriva de la secuencia de palabras de código, por lo que no se necesita transmitir o almacenar información adicional sobre el tamaño del paso (tasa de bits menor). Esto, sin embargo, ocasiona que la calidad de la salida reconstruida sea más sensible a errores en la transmisión. Con el control de alimentación hacia delante, las palabras de código y el tamaño del paso sirven como representación de la señal. Aunque esto incrementa la complejidad de la representación, existe la posibilidad de aplicar protección contra errores sobre $\Delta(n)$, mejorando significativamente la calidad de la salida para una tasa alta de errores de transmisión.

Como se muestra en la tabla 2.3, los parámetros de ADPCM incluyen los parámetros del predictor y del cuantizador. ADPCM-FF usa una ventana rectangular de longitud M para la estimación de la variancia. Esta variancia es actualizada cada M muestras. En los sistemas ADPCM-FB, se usa una ventana exponencial con parámetro α.

Parámetros	Nombre	Rango	Valor Típico
Orden del predictor	P	1-10	5
Tamaño de la ventana rectangular	M (sólo FF)	1-500	50
Número de bits	B	1-16	5
Máximo nivel del cuantizador	S_{max}	0-32767	32767
Tipo de cuantizador	-	-	*midtread* o *midriser*
Escalamiento del tamaño del paso	Δ_0	-	-
Mínimo tamaño del paso	Δ_{min}	50-500	100
Máximo tamaño del paso	Δ_{max}	100-10000	5000
Escalamiento de la ganancia	G_0	-	-
Ganancia mínima	G_{min}	1-100	0.1
Ganancia máxima	G_{max}	10-1000	100

Cuadro 2.3: *Parámetros de ADPCM*

Los procesos de adaptación del tamaño del paso pueden proveer mejoras en el SNR de alrededor de 5dB sobre PCM no adaptable de ley μ estándar. Esta mejora junto con los 6dB que se pueden obtener de la configuración diferencial con predicción fija indican que ADPCM-FF y ADPCM-FB logran cerca de 10-12 dB más que un cuantizador fijo con el mismo número de niveles.

De acuerdo con una evaluación subjetiva mediante tests de preferencia, ADPCM de 4 bits es calificado como PCM log de 6-7 bits.

2.2. Vocoder LPC

Una de las técnicas de análisis de la voz más poderosas es el método del análisis por predicción lineal. Esto se debe a tres razones básicas. Primero, el modelo de predicción lineal se encuentra intimamente relacionado con el modelo de producción de la voz discutido en el capítulo 1, en el que se mostró que la voz puede ser modelada como la salida de un sistema lineal variante en el tiempo excitado ya sea por pulsos cuasi-periódicos (durante segmentos sonoros), o ruido aleatorio (durante segmentos sordos). En segundo lugar aunque el modelo de predicción lineal no iguala al sistema de producción de la voz, puede capturar las propiedades perceptualmente importantes de la misma: tono, formantes, espectro, funciones del área del tracto vocal. Finalmente las técnicas de análisis por predicción lineal son técnicas en el domnio del tiempo, lo que las hace una buena elección para las técnicas DSP e implementaciones VLSI.

Las técnicas de predicción lineal han sido también discutidas en el contexto de los métodos de codificación de forma de onda de la sección 2.1. Se sugirió que un predictor lineal podría aplicarse al esquema de codificación diferencial para reducir la tasa de bits de la representación digital de la forma de onda de la voz. De hecho, la base matemática para un predictor adaptable de orden mayor a uno usado para la codificación DPCM es idéntica al análisis que presentaremos a continuación. En esta sección mostraremos cómo la predicción lineal básica conduce a un conjunto de técnicas de análisis que pueden ser usadas para estimar los parámetros de un modelo de voz. Este conjunto general de técnicas de análisis por predicción lineal se conoce como codificación por predicción lineal o LPC.

LPC es una técnica usada en una gran variedad de tipos de codificadores de voz. Por ejemplo es usada en codificadores específicos para la voz o *vocoders*, codificadores de forma de onda, codificadores de análisis por síntesis e incluso en codificadores en el dominio de la frecuencia.

El término "predicción lineal" se refiere a una gran variedad de formulaciones del problema del modelado de la voz, que son esencialmente equivalentes. Las diferencias entre estas formulaciones se encuentran a menudo en la forma en que se aborda el problema. En otros casos las diferencias se encuentran en algunos cálculos usados para obtener los coeficientes del predictor.

Los vocoders LPC tienen la ventaja de ser capaces de operar a bajas tasas de bits usando recursos computacionales relativamente modestos pero proporcionando una representación codificada de la señal original de voz que es muy usada. Su principal desventaja es que limita la calidad del codificador. Esto es, los vocoders LPC pueden lograr voz de buena calidad a tasas de bits bajas e incluso muy bajas, pero no pueden proporcionar voz de calidad óptima sin importar el número de bits empleados.

El vocoder LPC es completamente paramétrico. Esto significa que la voz codificada se encuentra completamente caracterizada por los parámetros lentamente variantes en el tiempo de un modelo de síntesis de voz. El modelo fuente-filtro de la figura 1.4 ha sido utilizado por la mayoría de los vocoders de bajas tasas de bits.

2.2.1. Modelo LPC

La idea básica detrás del modelo LPC es que dada una muestra de voz en un tiempo n, $s(n)$, ésta puede ser aproximada como una combinación lineal de las p muestras de voz previas, de forma que:

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \ldots + a_p s(n-p) \tag{2.11}$$

o bien

$$\tilde{s}(n) \approx \sum_{k=1}^{p} a_k s(n-k) \tag{2.12}$$

Debido a que el tracto vocal se mueve de forma relativamente lenta, la voz puede considerarse un proceso aleatorio cuyas propiedades varían lentamente. Esto conduce al principio básico de estacionaridad en tiempo corto usado en el análisis LPC. Este principio establece que la señal de voz puede ser considerada estacionaria durante una ventana de N muestras siempre y cuando N sea lo suficientemente pequeña. Este principio conduce a un modelado de la voz mediante una sucesión de filtros fijos, $H(z)$'s, cuyos coeficientes $a_1, a_2, \ldots, a_p$, (siendo p el orden del predictor) permanecen constantes dentro de la ventana o trama de voz analizada.

Si el estimado de la ecuación (2.12) se resta a la señal de voz original, obtenemos una señal de error, $e(n)$, llamada señal de error de predicción o señal residual (ver figura 2.5):

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{2.13}$$

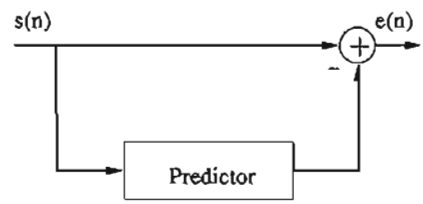


Figura 2.5: *Obtención de la señal residual de un segmento de voz*

Por lo tanto, para que el modelo LPC produzca una señal sintética de buena calidad, $\hat{s}(n)$, debe obtenerse una buena representación de la señal residual, $e(n)$ usando una señal de excitación, $Gu(n)$.

$$\hat{s}(n) = \tilde{s}(n) + e(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{2.14}$$

donde $\hat{s}(n)$ es la voz sintética producida por el modelo, $u(n)$ es una excitación normalizada y G es la ganancia de la excitación usada para igualar la energía de la voz sintética con la de la señal original.

En la figura 2.6 se observan dos segmentos de voz y su correspondiente señal residual para un predictor de orden $p = 10$; la señal residual del segmento de voz sonora es pseudoperiódica y la del segmento de voz sorda es tipo ruido, tal y como se estableció en el capítulo 1. Por lo tanto, el modelo de la señal de excitación, $u(n)$, de los vocoders LPC consiste simplemente en impulsos periódicos o ruido blanco.

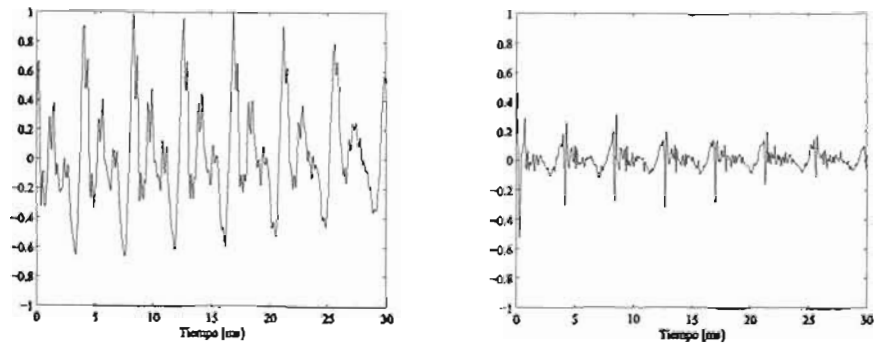
Si expresamos a 2.14 en el dominio de z, se tiene:

$$\hat{S}(z) = \sum_{k=1}^{p} a_k z^{-k} S(z) + GU(z) \tag{2.15}$$

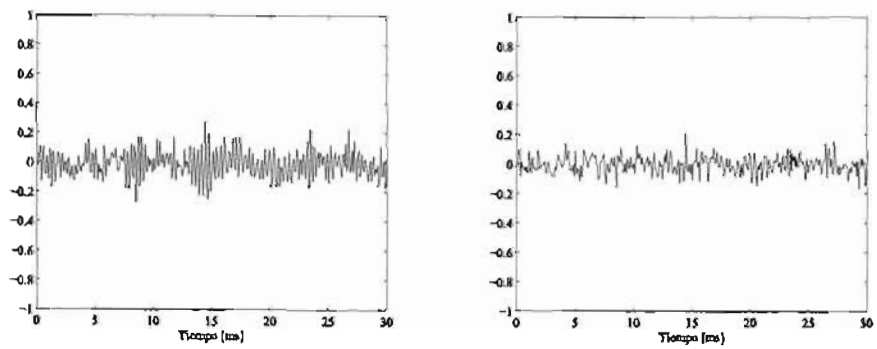
que conduce a la función de transferencia del filtro LPC de síntesis:

$$H(z) = \frac{\hat{S}(z)}{GU(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{1 - A(z)} \tag{2.16}$$

donde el filtro de predicción lineal, $A(z)$, es:



(a) Segmento sonoro de voz y su residuo



(b) Segmento sordo de voz y su residuo

Figura 2.6: *Segmentos de voz (izquierda) con sus correspondientes residuos LPC (derecha)*

$$A(z) = \sum_{k=1}^{p} a_k z^{-k} \tag{2.17}$$

que claramente corresponde a función de transferencia $\tilde{S}(z)/S(z)$ de la relación mostrada en la ecuación (2.12).

La interpretación de la ecuación (2.16) se proporciona en la figura 2.7, que muestra una fuente de excitación normalizada, $u(n)$, escalada por una ganancia G, y que actúa como entrada del sistema todo-polos, $H(z)$, para producir una señal de voz, $\hat{s}(n)$.

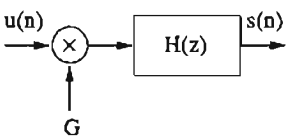


Figura 2.7: *Modelo de predicción lineal para la voz*

Como la función de excitación para la voz es esencialmente: un tren de impulsos cuasi-periódicos (para sonidos sonoros) o una fuente de ruido aleatorio (para sonidos sordos), el modelo de síntesis de voz, correspondiente al análisis LPC, es el mostrado en la figura 2.8.

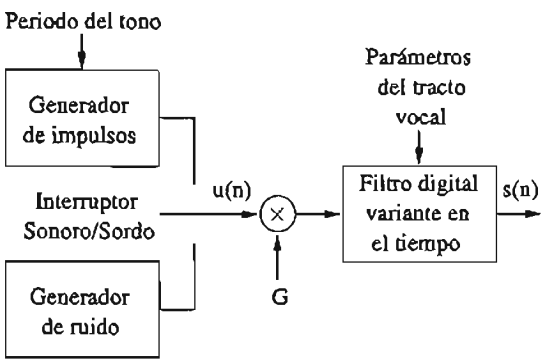


Figura 2.8: *Modelo de síntesis de la voz basado en el modelo LPC*

La fuente de excitación normalizada es elegida mediante un switch cuya posición es controlada por el carácter sonoro/sordo de la voz. Por tanto, los parámetros de este modelo son la clasificación sonoro/sordo, el periodo de los sonidos sonoros, el parámetro de ganancia, y los coeficientes del predictor, a_k; los coeficientes del predictor son los parámetros del tracto vocal, que proporcionan la información sobre la envolvente del espectro del segmento de voz; mientras que los demás son parámetros de la señal de excitación, que proporcionan la información sobre los detalles espectrales de la voz. Todos estos parámetros se obtienen a partir de la señal de voz de entrada y varían lentamente con el tiempo.

El principal problema del análisis por predicción lineal es determinar el conjunto de coeficientes del predictor a_k, directamente de la señal de voz, de tal forma que las propiedades espectrales del filtro digital de la figura 2.8 sean iguales a las de la trama de voz analizada. El objetivo entonces es encontrar un conjunto de coeficientes del predictor que minimicen la energía del error de predicción medio cuadrático, E_n, sobre una trama de voz. (Usualmente este tipo de análisis espectral de tiempo corto es realizado en tramas sucesivas de voz, con un espaciamento entre tramas del orden de 10ms).

$$E_n = \sum_m e_n^2(m) \tag{2.18}$$

la cual, usando la definición de $e_n(m)$ en términos de $s_n(m)$, de la ecuación (2.13) puede ser escrita como:

$$E_n = \sum_m (s_n(m) - \tilde{s}_n(m))^2 = \sum_m \left[s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k) \right]^2 \tag{2.19}$$

Existen varias formas de obtener los coeficientes del predictor que minimicen la energía del error de la ecuación (2.19). Dos de los métodos más populares son el método de la autocorrelación y el método de la covarianza. Ambos métodos son técnicas en el dominio del tiempo por lo que son fácilmente implementadas en procesadores DSP. Los dos métodos producen conjuntos de coeficientes LPC para el filtro del tracto vocal; y además, ambos proporcionan soluciones eficientes. La principal diferencia entre ellos es la forma en que se aplica una ventana a la voz.

En el método de la autocorrelación, se aplica primero una ventana a la señal de voz, produciendo una aproximación en tiempo corto (distorsionada) de la voz original. Posteriormente se calcula la autocorrelación y se obtienen los coeficientes LPC. Matemáticamente, este es un proceso bastante manejable, que garantiza la producción de filtros estables del tracto vocal.

El método de la covarianza, por otro lado, no aplica una ventana directamente a la señal de voz, sino que la aplica al error. Por lo tanto, este método posee un potencial mejor funcionamiento, pues la señal no fue "predistorsionada", pero la aplicación de la ventana a la señal de error produce otro tipo de distorsión que es menos manejable matemáticamente, por lo que es posible que el método de la covarianza produzca filtros inestables del tracto vocal.

En esta sección abordaremos únicamente el método de la autocorrelación ya que ha sido el más usado en la práctica.

2.2.2. Analizador LPC

La figura 2.9 muestra el diagrama de bloques del analizador LPC. Su principal tarea es analizar periódicamente la voz de entrada para estimar, cuantizar, codificar y transmitir los parámetros de modelo del tracto vocal y los parámetros del modelo de la excitación.

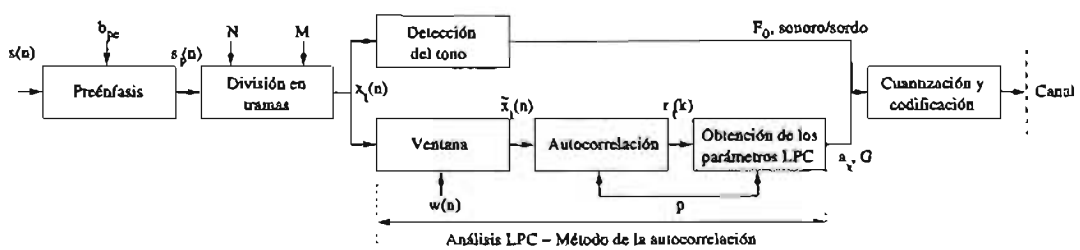


Figura 2.9: *Diagrama de bloques de un analizador LPC*

El transmisor del vocoder LPC realiza dos tipos de análisis: el análisis de la excitación (detección del tono) y el análisis del tracto vocal (análisis LPC).

Las salidas del detector del tono (parámetros de la excitación) consisten en una decisión sonoro/sordo para cada trama, y, para tramas sonoras, el periodo del tono. Los resultados del análisis LPC (parámetros del tracto vocal) para cada trama son un parámetro de ganancia y un conjunto de coeficientes LPC. Todos estos parámetros son cuantizados, codificados y multiplexados en el flujo de información de salida para su transmisión o almacenamiento.

Como se muestra en la figura 2.9, la obtención (usando el método de la autocorrelación) de los parámetros del tracto vocal y de la excitación se logra básicamente de la siguiente forma:

- Filtro de preénfasis
- Análisis LPC. Método de la autocorrelación
  - División en tramas
  - Aplicación de la ventana
  - Cálculo de la autocorrelación
  - Obtención de los parámetros LPC
  - Cálculo de la ganancia
- Detección del Tono

Posterior a la obtención de los parámetros necesarios, se encuentra la

- Cuantización y codificación

de los mismos; los cuales son multiplexados para su transmisión.

A continuación se describirá cada uno de estos procedimientos.

Preénfasis

La teoría acústica indica que el espectro de los sonidos sonoros posee una pendiente de -6 dB/octava, conforme aumenta la frecuencia. Ésta es una combinación de una pendiente de -12 dB/octava debido a la fuente de excitación de la voz y una pendiente +6 dB/octava debida a la radiación producida por la boca. Esto significa que cada vez que se duplica la frecuencia, la amplitud de la señal, y por tanto la respuesta del tracto vocal, son reducidas a la mitad. Si se permite que esto ocurra, el modelo LPC aproximará exitosamente las bajas frecuencias, pero hará un trabajo pobre con las altas frecuencias. Es por tanto deseable compensar la caida de -6 dB/octava preprocesando la señal de voz para darle un levantamiento de +6 dB/octava en el rango apropiado, de forma que el espectro tenga un rango dinámico similar a través de toda la banda de frecuencias. A este proceso se le llama preénfasis, pues enfatiza las altas frecuencias. En un sistema digital de procesamiento de señales, el preénfasis puede ser implementado usando un filtro digital paso-altas de primer orden. En el caso de sonidos sordos no hay necesidad de aplicar preénfasis, ya que no hay ninguna pendiente que deba ser removida; sin embargo, por simplicidad, se aplica preénfasis normalmente a los sonidos sordos también.

El sistema digital usado en el preénfasis puede ser ya sea fijo o adaptable. Quizás la red de preénfasis mayormente usada es el sistema fijo de primer orden (figura 2.10):

$$H(z) = 1 - b_{pe}z^{-1} \quad 0{,}9 \leq b_{pe} \leq 1{,}0 \tag{2.20}$$

La constante, b_{pe}, controla el grado de preénfasis. Aunque el valor óptimo de b_{pe} puede ser estimado estadísticamente, éste valor es diferente para diferentes locutores y el análisis no es muy sensible a su valor, por lo que el valor utilizado no es crítico; sin embargo, el valor más común es $b_{pe} = 15/16 = 0{,}9375$.

En este caso, la salida de la red de preénfasis, $\tilde{s}(n)$, se encuentra relacionada con la entrada de la red, $s(n)$, mediante la ecuación en diferencias:

$$s_p(n) = s(n) - b_{pe}s(n-1) \tag{2.21}$$

La figura 2.11 muestra las características de la magnitud de $H(e^{j\omega})$ para el valor $b_{pe} = 0{,}95$ y una frecuencia de muestreo de 8 kHz. Puede observarse que en $F = 4kHz$ (la mitad de la frecuencia de muestreo) existe un incremento en la magnitud de 32dB con respecto a $F = 0$.

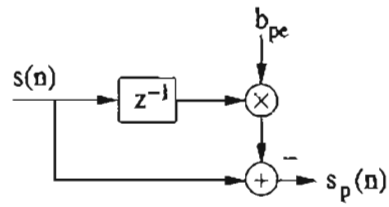


Figura 2.10: *Filtro digital de preénfasis*

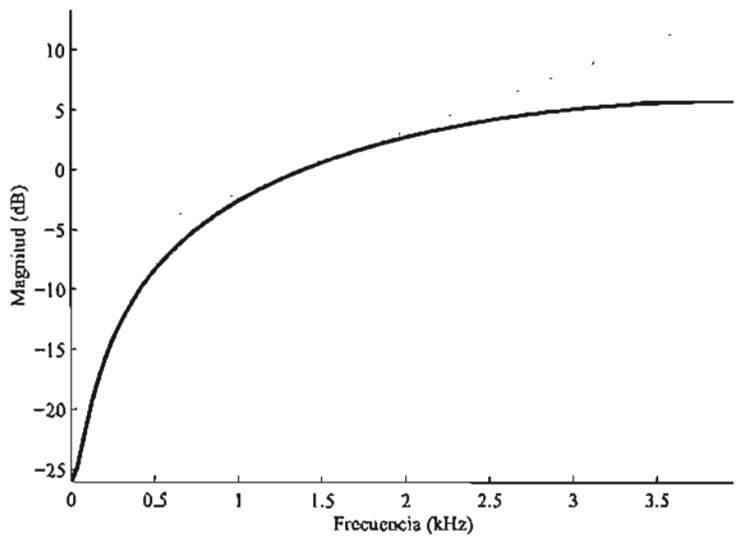


Figura 2.11: *Magnitud del espectro de la red de preénfasis LPC para* $b_{pe} = 0{,}95$. *La línea punteada muestra la pendiente de 6 dB/octava*

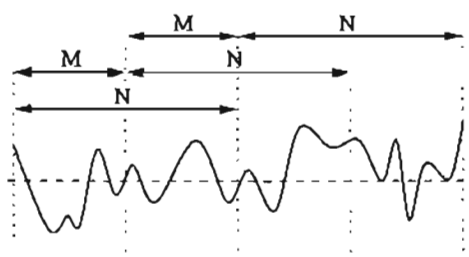


Figura 2.12: *División de la voz en tramas que se traslapan*

División en tramas

Una vez que se aplica el preénfasis, la señal resultante, $s_p(n)$, es dividida en tramas de N muestras, separadas M muestras entre sí. La figura 2.12 ilustra la división en tramas para el caso en el que $M = (1/2)N$.

La primera trama mostrada consiste en las primeras N muestras de voz. La segunda trama comienza M muestras después de la primera trama, y se traslapa con ella por $N - M$ muestras. De manera similar, la tercera trama comienza $2M$ muestras después de la primera trama (o M muestras después de la segunda) y la traslapa $N - 2M$ muestras. Este proceso continúa hasta que toda la voz se ha considerado para una o más tramas. Es fácil observar que si $M \leq N$, entonces las tramas adyacentes se traslapan (como en la figura 2.12), y los espectros LPC estimados resultantes estarán correlacionados de trama a trama; si $M \ll N$, entonces los espectros LPC estimados de trama a trama serán bastante suaves. Por otro lado, si $M > N$, no habrá traslape entre tramas adyacentes; de hecho, algunas partes de la señal de voz se perderán por completo (es decir, no aparecerán en ninguna trama de análisis), y la correlación entre los espectros LPC estimados resultantes de tramas adyacentes contendrán una componente ruidosa cuya magnitud aumentará con el incremento en M (es decir, mientras más voz es omitida del análisis). Esta situación es intolerable en cualquier análisis LPC práctico. Si denotamos a la $\ell^{ésima}$ trama de voz como $x_\ell(n)$, y existen L tramas en toda la señal de voz, entonces:

$$x_\ell(n) = s_p(M\ell + n) \quad n = 0, 1, \ldots, N-1 \quad \ell = 0, 1, \ldots, L-1 \tag{2.22}$$

Esto es, la primera trama de voz, $x_0(n)$, incluye a las muestras $s(0)_p, s(1)_p, \ldots, s_p(N-1)$, la segunda trama de voz, $x_1(n)$ incluye a las muestras $s_p(M), s_p(M+1), \ldots, s_p(M+N-1)$ y la $L^{ésima}$ trama de voz, $x_{L-1}(n)$ incluye a las muestras $s_p(M(L-1)), s_p(M(L-1)+1), \ldots, s_p(M(L-1)+N-1)$.

Ventana

El método LPC es preciso cuando se aplica a señales estacionarias, esto es, señales cuyo comportamiento no cambia con el tiempo. Sin embargo éste no es el caso de la voz. Para poder aplicar el método de la autocorrelación del análisis LPC, segmentamos la señal en tramas, las cuales son cuasiestacionarias.

Aunque es una posibilidad, generalmente evitamos extraer las tramas simplemente haciendo cero todo lo que se encuentre fuera de la región de interés, ya que esto esparciría la señal en el dominio de la frecuencia.

Por lo tanto, para obtener las tramas de la señal se utiliza una ventana que disminuya la amplitud de la señal cerca de $n = 0$ y de $n = N$ (N es el tamaño de la ventana de análisis) para minimizar los errores en los límites de la sección.

Este proceso se ilustra en la figura 2.13. Se multiplica la señal de voz, $x_\ell(n)$, por una ventana, $w(n)$, la cual es igual a cero fuera del intervalo que deseamos extraer. Por cada desplazamiento M de la ventana, usualmente de 10 a 30 ms, se crea una trama de análisis, $\tilde{x}_\ell$. Por tanto $\tilde{x}_\ell(n) = 0$ para $n < 0$ y $n \geq N$.

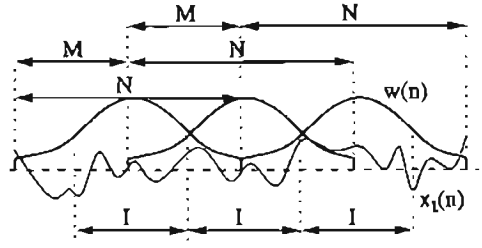


Figura 2.13: *Segmentación de la señal de voz en tramas cuasiestacionarias mediante la aplicación de una ventana. Las tramas se traslapan*

Si definimos una ventana $w(n)$, $0 \leq n \leq N-1$, entonces el resultado de aplicar esta ventana a la señal:

$$\tilde{x}_\ell(n) = x_\ell(n)w(n) \quad 0 \leq n \leq N-1 \tag{2.23}$$

En la figura 2.14 se muestran algunas de las ventanas típicas usadas en la obtención de las tramas de análisis, mientras que en la figura 2.15 se muestran los espectro correspondientes. La más usada entre ellas es la ventana de Hamming.

Hanning

$$w[n] = 0{,}5 - 0{,}5cos\left(\frac{2\pi n}{N-1}\right) \quad 0 \leq n \leq N-1 \tag{2.24}$$

Hamming

$$w[n] = 0{,}54 - 0{,}46cos\left(\frac{2\pi n}{N-1}\right) \quad 0 \leq n \leq N-1 \tag{2.25}$$

Blackman

$$w[n] = 0{,}42 - 0{,}5cos\left(\frac{2\pi n}{N-1}\right) + 0{,}08cos\left(\frac{4\pi n}{N-1}\right) \quad 0 \leq n \leq N-1 \tag{2.26}$$

donde la longitud de la ventana, N, generalmente se elige de forma que se incluyan unos cuantos periodos del tono (20-40 ms) para reducir los efectos de la excitación en la estimación de los parámetros del filtro del tracto vocal, y obtener un estimado preciso del espectro de la voz. El valor típico de N es de 30 ms, considerando una señal muestreada a 8 kHz, la este valor corresponde a $N = 240$ muestras.

La longitud de la trama de análisis, I, determina el número de muestras sobre las que se usarán los coeficientes LPC resultantes. La relación I/N representa el traslape entre dos tramas de análisis adyacentes, $N - M$, donde M es el periodo de la trama. En la práctica se usa típicamente un traslape del 50 % ($I = N/2$). La elección dependerá de la tasa de bits deseada (mientras las tramas sean transmitidas con más frecuencia, la tasa de bits será mayor), así como de la calidad deseada (mientras menor sea el periodo de la trama, mejor será la calidad, pues es posible capturar con mayor precisión las transiciones de la señal de voz).

El periodo de la trama se expresa también en milisegundos (o en muestras si se conoce la frecuencia de muestreo), y típicamente se usan valores entre 10-30 ms. El inverso del periodo de la trama es la tasa de tramas, expresada en tramas/s; por ejemplo, un periodo de trama de 20 ms corresponde a una tasa de tramas de 50 tramas/s (o Hz).

Autocorrelación

Para resolver la ecuación (2.19), y encontrar los coeficientes del predictor, derivamos E_n con respecto a cada a_k e igualamos el resultado a cero,

$$\frac{\partial E_n}{\partial a_k} = 0, \quad para\ k = 1, 2, \ldots, p \tag{2.27}$$

Lo cual nos conduce al conjunto de ecuaciones lineales:

$$a_1 r_\ell(0) + a_2 r_\ell(1) + \ldots + a_p r_\ell(p-1) = r_\ell(1) \tag{2.28}$$

$$a_1 r_\ell(1) + a_2 r_\ell(0) + \ldots + a_p r_\ell(p-2) = r_\ell(2) \tag{2.29}$$

$$\vdots \tag{2.30}$$

$$a_1 r_\ell(p-1) + a_2 r_\ell(p-2) + \ldots + a_p r_\ell(0) = r_\ell(p) \tag{2.31}$$

$$\tag{2.32}$$

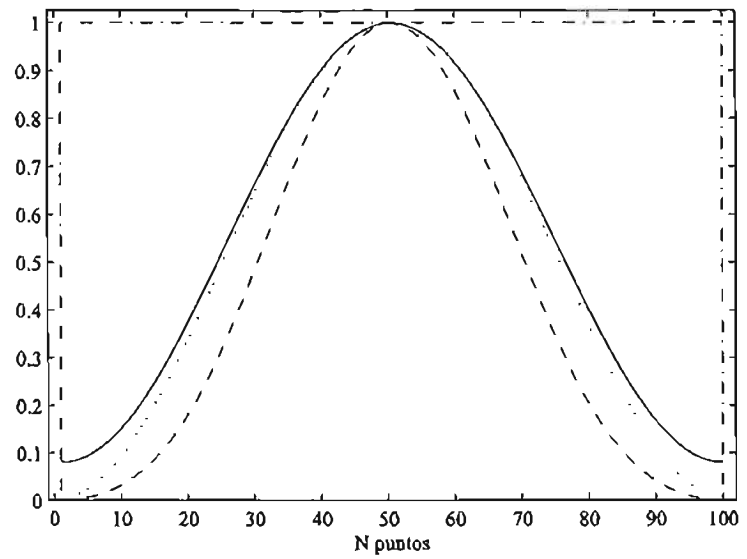


Figura 2.14: *Tipos de ventanas: — Hamming, · · · Hanning, - - - Blackman, ···· rectangular*

En las ecuaciones anteriores hemos definido a la $m^{ésima}$ autocorrelación de la trama de análisis como

$$r_\ell(m) = r_\ell(-m) = \sum_{n=0}^{N-1-m} \tilde{x}_\ell(n)\tilde{x}_\ell(n+m) \quad m = 0, 1, \ldots, p \tag{2.33}$$

Esto es, cada trama de la señal a la que se aplicó la ventana es a continuación autocorrelacionada para obtener el conjunto de coeficientes de la autocorrelación que serán usados posteriormente en el cálculo de los coeficientes LPC.

El orden de la autocorrelación, p, es el orden del análisis LPC. Normalmente dos polos para modelar cada formante de la voz, por lo que en la práctica, típicamente se usan valores de p desde 8 hasta 16. Un beneficio de la autocorrelación es que el término $R_\ell(0)$ de la autocorrelación es la energía de la $\ell^{ésima}$ trama. La energía de la trama es un parámetro importante para los sistemas de detección del tono.

El conjunto de ecuaciones 2.28 se pueden expresar en forma matricial como

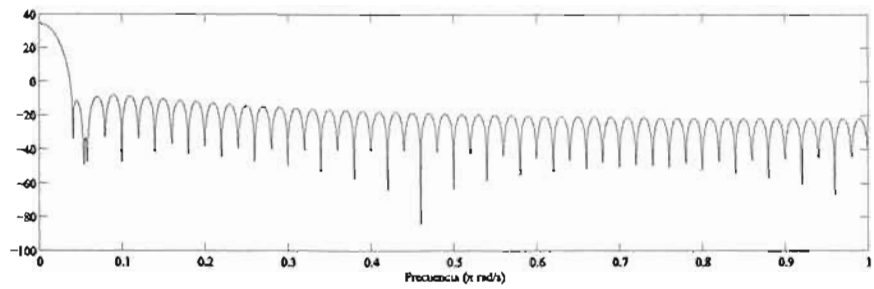
$$\mathbf{R} \cdot \mathbf{a} = \mathbf{r} \tag{2.34}$$

$$\begin{bmatrix} r_\ell(0) & r_\ell(1) & r_\ell(2) & \ldots & r_\ell(p-1) \\ r_\ell(1) & r_\ell(0) & r_\ell(1) & \ldots & r_\ell(p-2) \\ r_\ell(2) & r_\ell(1) & r_\ell(0) & \ldots & r_\ell(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_\ell(p-1) & r_\ell(p-2) & r_\ell(p-3) & \ldots & r_\ell(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r_\ell(1) \\ r_\ell(2) \\ r_\ell(3) \\ \vdots \\ r_\ell(p) \end{bmatrix} \tag{2.35}$$

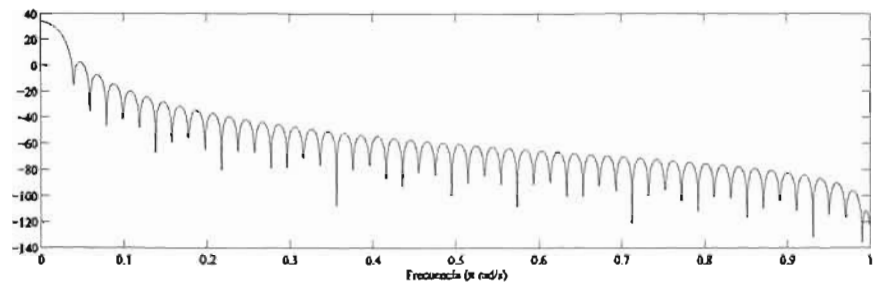
La matriz $p \times p$ con los valores de la autocorrelación es una matriz Toeplitz simétrica [2]; como este tipo de matriz es no singular [3] puede ser invertida, y por lo tanto puede encontrarse una solución

[2] Los elementos en sus diagonales son iguales, esto es, $a_{i,j} = a_{i-1,j-1}$

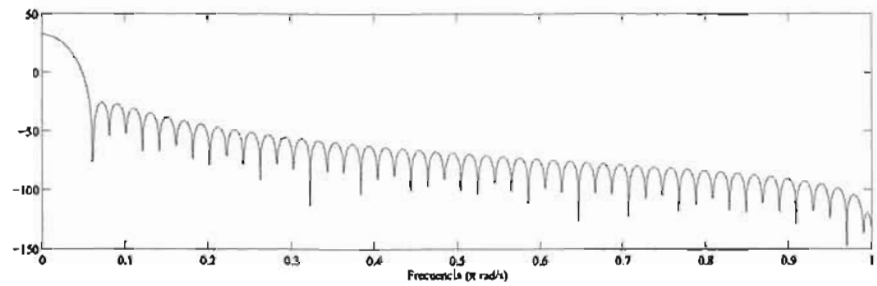
[3] $det(\mathbf{R}) \neq 0$



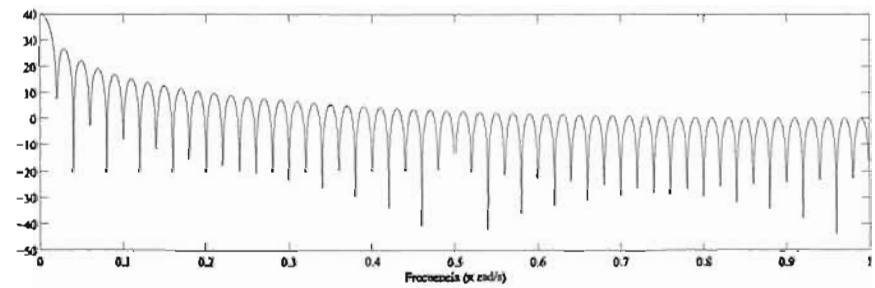
(a) Hamming



(b) Hanning



(c) Blackman



(d) Rectangular

Figura 2.15: *Espectro de las ventanas de Hamming, Hanning, Blacknan y rectangular*

$$\mathbf{a} = \mathbf{R}^{-1}\mathbf{r} \tag{2.36}$$

Existen varios algoritmos para encontrar los coeficientes del predictor. Sin embargo, uno de los más usados por su fácil implementación y eficiencia computacional es el algoritmo de Durbin. Este algoritmo es recursivo y usa la estructura Toeplitz de $\mathbf{R}$ para encontrar los coeficientes LPC.

Obtención de los parámetros LPC

Una vez que se obtiene cada trama de $p+1$ autocorrelaciones, se convierte en un conjunto de parámetros LPC. El método formal de conversión de los coeficientes de la autocorrelación en un conjunto de parámetros LPC (para el método LPC de la autocorrelación) es el método de Durbin

Este es un algoritmo recursivo que se aplica para resolver sistemas de ecuaciones lineales de la forma $\mathbf{R} \cdot \mathbf{a} = \mathbf{r}$ donde $\mathbf{R}$ es una matriz Toeplitz Hermitiana [4] definida positiva [5].

El algoritmo de Durbin está dado por las siguientes ecuaciones (por conveniencia, se omitirá el subíndice ℓ en $r_\ell(m)$):

$$E^{(0)} = r(0) \tag{2.37}$$

$$k_i = \frac{r(i) - \sum_{j=1}^{(i-1)} \alpha_j^{i-1} r(\mid i-j \mid)}{E^{(i-1)}} \quad 1 \le i \le p \tag{2.38}$$

$$\alpha_j^{(i)} = k_i \tag{2.39}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad 1 \le j \le i-1 \tag{2.40}$$

$$E(i) = (1-k_i^2)E^{(i-1)} \tag{2.41}$$

donde la sumatoria de la ecuación (2.39) se omite para $i = 1$. El conjunto de ecuaciones 2.38-2.41 son resueltas recursivamente para $i = 1, 2, \ldots, p$, y a partir de éstas se pueden obtener diferentes conjuntos de parámetros LPC:

LPC

$$a_i = \alpha_i^{(p)} \tag{2.42}$$

PARCOR

$$k_i \tag{2.43}$$

LAR

$$g_i = log\left(\frac{1-k_i}{1+k_i}\right) = log\left(\frac{A_{i+1}}{A_i}\right) \tag{2.44}$$

Cepstral

$$c_0 = lnG^2 \tag{2.45}$$

$$c_i = a_i + \sum_{k=1}^{i-1}\left(\frac{k}{i}\right) c_k a_{i-k} \quad 1 \le i \le p \tag{2.46}$$

$$c_i = \sum_{k=1}^{i-1}\left(\frac{k}{i}\right) c_k a_{i-k} \quad i > p \tag{2.47}$$

[4] $a(i,j) = conj(a(j,i))$ en el caso de matrices reales Hermitiana y simétrica son equivalentes

[5] Matriz simétrica con autovalores y pivotes positivos. Definición: $x^T Ax > 0$ a menos que $x = 0$

LSP

$$P(z) = 1 + \sum_{k=1}^{p} (a_k + a_{p+1-k})\, z^{-k} + z^{-(p+1)} \tag{2.48}$$

$$Q(z) = 1 + \sum_{k=1}^{p} (a_k - a_{p+1-k})\, z^{-k} - z^{-(p+1)} \tag{2.49}$$

O cualquier transformación deseada de los conjuntos anteriores.

El algoritmo de Durbin requiere aproximadamente $O(p^2)$ operaciones (p divisiones, p^2 multiplicaciones y p^2 sumas), por lo que es mucho más eficiente que el algoritmo estándar de eliminación gaussiana, que requiere $O(p^3)$ operaciones ($n^3/3$ sumas y $n^3/3$ multiplicaciones y divisiones).

Los coeficientes LPC son generalmente inapropiados para la cuantización debido a su relativamente grande rango dinámico y posibles problemas de inestabilidad de los filtros. Pequeños errores en la cuantización de los coeficientes LPC individuales producen errores espectrales relativamente grandes, y también pueden ocasionar inestabilidad en el filtro de sintesis cuantizado. Para evitar una distorsión inaceptable, se requiere una gran cantidad de bits (80-100 bits/trama) para la cuantización escalar de los coeficientes LPC. Por lo tanto es necesario transformar los coeficientes LPC en un conjunto equivalente de parámetros que posean menor sensibilidad espectral y aseguren la estabilidad del filtro todo polos tras la cuantización.

Estas representaciones adecuadas son las presentadas con anterioridad: PARCOR, LAR, Cepstral, LSP. Todas ellos transportan la misma información, pero poseen diferentes propiedades de cuantización e interpolación, por lo que son usadas en los casos en que sus propiedades pueden proporcionar una mejor calidad en la voz.

Coeficientes PARCOR Los coeficientes k_i para $i = 1, \ldots, P$ contienen la misma información que los coeficientes LPC, y son llamados coeficientes de reflexión o PARCORs. Estos coeficientes son espectralmente menos sensibles a la cuantización que los LPC, además son muy importantes en la realización física del filtro de síntesis todo polos, ya que son los multiplicadores de un filtro lattice (sección 2.2.3).

La cantidad E^i es la energía del error de predicción, como E^i es una cantidad positiva, la ecuación (2.41) indica que todos los coeficientes PARCOR tienen una magnitud menor a uno. Esto es,

$$-1 < k_i < 1 \tag{2.50}$$

Como el filtro LPC del tracto vocal es recursivo, la estabilidad es un importante. Resulta que la condición de la ecuación (2.50) es necesaria y suficiente para garantizar la estabilidad del filtro del tracto vocal.

Coeficientes LAR Los coeficientes LAR corresponden al logaritmo de la relación entre las áreas de las secciones adyacentes, A_i, de los tubos uniformes sin pérdidas equivalentes al tracto vocal (figura 1.6), que poseen la misma función de transferencia que el modelo de predicción lineal.

Coeficientes Cepstral Los coeficientes cepstral, son los coeficientes de la representación mediante transformada de Fourier del logaritmo de la magnitud espectro de la señal de voz, esto es, el cepstrum se define como la transformada de Fourier inversa de $C_i(\omega) = log_e S(\omega)$.

Coeficientes LSP Los LSP fueron introducidos por Itakura como una alternativa a los coeficientes LPC. Para el filtro inverso LPC de orden p, podemos extender el orden a $p+1$ sin introducir información adicional haciendo que el $(p+1)^{ésimo}$ coeficiente de reflexión sea 1 ó -1. Esto es equivalente a establecer completamente cerrado o completamente abierto el correspondiente modelo de tubo acústico en la $(p+1)^{ésima}$ etapa. Por lo tanto tenemos:

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)}A(z^{-1}) \\ &= 1 + \sum_{k=1}^{p} (a_k + a_{p+1-k})\, z^{-k} + z^{-(p+1)} \qquad (2.51) \\ Q(z) &= A(z) - z^{-(p+1)}A(z^{-1}) \\ &= 1 + \sum_{k=1}^{p} (a_k - a_{p+1-k})\, z^{-k} - z^{-(p+1)} \qquad (2.52) \end{aligned}$$

(2.53)

Es obvio que $P(z)$ es un polinomio simétrico mientras que $Q(z)$ es un polinomio antisimétrico y

$$A(z) = \frac{P(z) + Q(z)}{2} \qquad (2.54)$$

$P(z)$ y $Q(z)$ poseen tres propiedades importantes:

- Todos los ceros de $P(z)$ y $Q(z)$ se encuentran dentro del círculo unitario.
- Los ceros de $P(z)$ y $Q(z)$ se encuentran entrelazados.
- La propiedad de fase mínima de A(z) se preserva fácilmente tras la cuantización de los ceros de $P(z)$ y $Q(z)$. Como los ceros de $P(z)$ y $Q(z)$ se encuentran dentro del círculo unitario, pueden ser expresados como $e^{j\omega}$ y las ω's son llamadas LSF.

  Si el orden p es un número par mayor a 2, se tienen las siguientes propiedades adicionales para LSP:

- -1 es un cero de $P(z)$ mientras que 1 es un cero de $Q(z)$.
- Además de ± 1, $P(z)$ y $Q(z)$ poseen otros $p/2$ pares de ceros conjugados cada uno.

Por lo tanto, $P(z)$ y $Q(z)$ pueden ser reescritos como:

$$\begin{aligned} P(z) &= (1+z^{-1}) \prod_{i=1}^{p/2} (1 - z^{-1}e^{j\omega_i})(1 - z^{-1}e^{-j\omega_i}) \\ &= (1+z^{-1}) \prod_{i=1}^{p/2} 1 - 2cos\omega_i z^{-1} + z^{-2} \qquad (2.55) \\ Q(z) &= (1-z^{-1}) \prod_{i=1}^{p/2} (1 - z^{-1}e^{j\theta_i})(1 - z^{-1}e^{-j\theta_i}) \\ &= (1-z^{-1}) \prod_{i=1}^{p/2} 1 - 2cos\theta_i z^{-1} + z^{-2} \qquad (2.56) \end{aligned}$$

Aquí $\omega_i (1 \leq i \leq p/2)$, las fases de los ceros conjugados de $P(z)$, o los LSF's del polinomio simétrico, y $\theta_i (1 \leq i \leq p/2)$, las fases de los ceros conjugados de $Q(z)$, o los LSF del polinomio antisimétrico, se encuentran entrelazados entre sí en el intervalo $(0, \pi)$. Esto es

$$0 < \omega_1 < \theta_1 < \omega_2 < \theta_2 < \ldots < \omega_{p/2} < \theta_{p/2} < \pi \tag{2.57}$$

Por tanto tenemos que los coeficientes de $A(z)$ son equivalentes a las fases de los ceros de $P(z)$ y $Q(z)$, ω_i y θ_i. El conjunto de frecuencias de $P(z)$ y $Q(z)$ es llamado conjunto de Pares de líneas espectrales (LSP) o Frecuencias de líneas espectrales (LSF).

Estos parámetros, junto con VQ, poseen un papel muy importante en la codificación/decodificación de la voz. La famosa codificación CELP es un buen ejemplo del uso del principio de la predicción lineal así como de LSP para producir voz de alta calidad a bajas tasas de bits. Los parámetros LSP/LSF poseen un buen rango dinámico y garantizan la estabilidad del filtro, y pueden ser usados para codificar la información espectral LPC de manera más eficiente que cualquier otro tipo de parámetros.

Cálculo de la Ganancia

El parámetro de la ganancia en el modelo LPC es usado para producir una señal de voz sintética que tenga la misma energía que la señal de voz original. Esto puede lograrse igualando la energía de la respuesta al impulso del filtro LPC con la energía de la señal de voz original. A partir de esto se obtiene la siguiente relación entre la ganancia, G, y los coeficientes de la autocorrelación y LPC:

$$G = \left[r(0) - \sum_{k=1}^{p} a_k r(k) \right]^{\frac{1}{2}} \tag{2.58}$$

Detección del tono

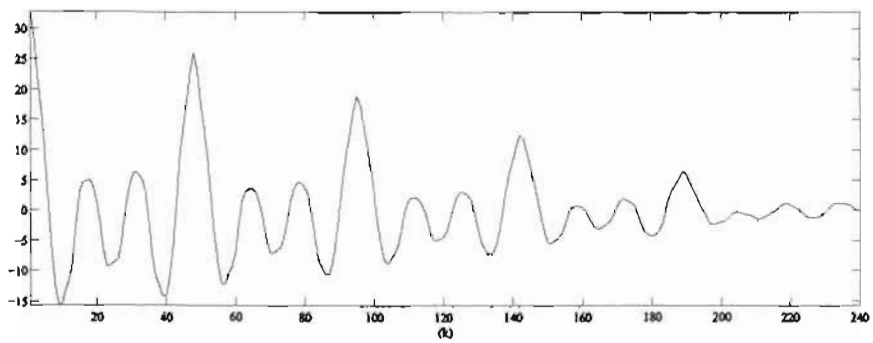
La determinación de la periodicidad de un segmento de voz es muy importante en muchos algoritmos de codificación de voz.

La frecuencia fundamental de la señal de voz es usualmente llamada tono, F_o. El inverso de la frecuencia del tono es el periodo del tono, τ, y usualmente se expresa en milisegundos o, si se conoce la frecuencia de muestreo, en muestras.

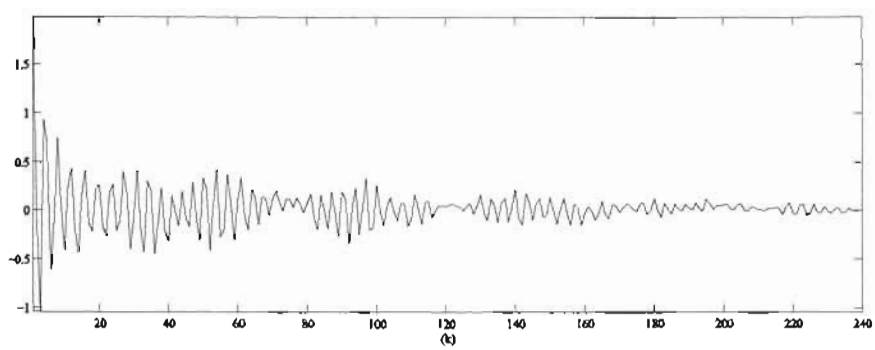
La determinación del tono consiste en dos etapas. En la primera, se debe determinar si el segmento de voz es sonoro o sordo. Si es sonoro, el periodo del tono se estima en la segunda etapa.

La estimación del tono es la parte más vulnerable de los vocoders LPC, ya que los segmentos sonoros que poseen una clara periodicidad y los segmentos sordos que claramente carecen de ella pueden ser fácilmente identificados; sin embargo, la identificación de segmentos intermedios es más difícil.

Este hecho ha estimulado el desarrollo de numerosos algoritmos de detección del tono, en el dominio del tiempo o de la frecuencia. Las técnicas basadas en la autocorrelación en tiempo corto de la voz o voz preprocesada han sido estudiados de manera extensa.



(a) Segmento sonoro



(b) Segmento sordo

Figura 2.16: *Ejemplos de la función de autocorrelación*

Detección del tono mediante autocorrelación Un método popular de detección del tono es el uso de la función de autocorrelación. Sea $s(n)$ un segmento de la señal de voz que comienza en $n = 0$ y que tiene una longitud de N muestras. Asumiendo que a $s(n)$ se le ha aplicado una ventana, la función de autocorrelación es definida por la ecuación

$$R(k) = \sum_{n=0}^{N-1} s(n)s(n+k) \quad k = 0, 1, 2, \ldots \tag{2.59}$$

La figura 2.16 muestra ejemplos de la función de autocorrelación para un segmento sonoro y uno sordo.

La variable k es el índice del tiempo. Para segmentos que demuestran periodicidad (sonoros), la función de autocorrelación posee picos en los tiempos correspondientes al periodo del tono, mientras que en la autocorrelación de segmentos sordos no se encuentran tales picos. Estas observaciones sugieren que la función de autocorrelación puede ser usada en la decisión sonoro/sordo y, en el caso de segmentos sonoros, en el cálculo del periodo del tono.

En la figura 2.16 se observa que la función de autocorrelación contiene la información necesaria para la determinación de la sonoridad de un segmento de voz, pero también contiene información asociada con el tracto vocal. Esta información adicional puede conducir a decisiones erróneas. Para mejorar la representación de la información del tono, procesamos la señal de voz recortándola. Al recortar la señal se eliminan partes de ella entre ciertos niveles, y lo que no es suprimido comienza desde un nivel cero. La figura 2.17 muestra la característica del recortador, la cual puede ser expresada de la forma siguiente:

$$\begin{aligned} y(n) &= s(n) - C_L \quad \text{si } s(n) \geq C_L \\ y(n) &= s(n) + C_L \quad \text{si } s(n) \leq -C_L \\ y(n) &= 0 \quad c.c. \end{aligned} \tag{2.60}$$

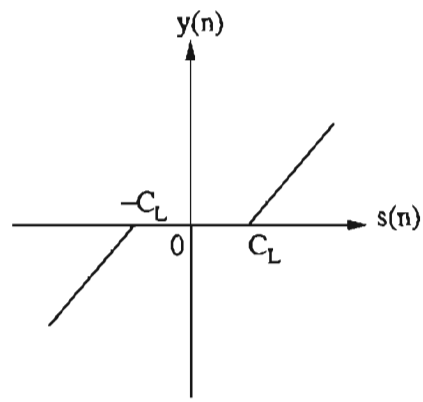
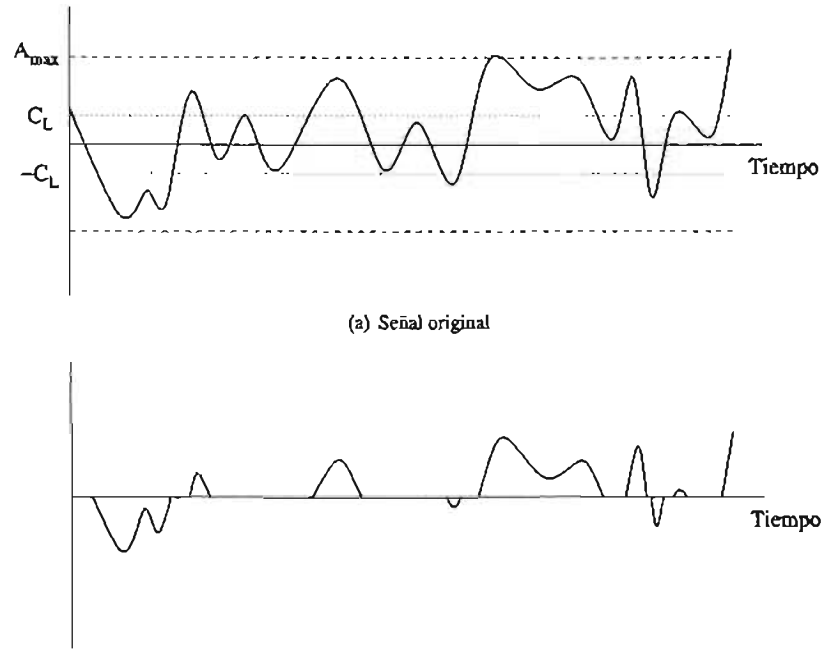


Figura 2.17: *Característica del recortador*

La figura 2.18 es un ejemplo de una señal antes y después de ser recortada. El efecto que provoca el recortar la señal es el de destruir la información de las formantes y retener únicamente la información de la periodicidad. El umbral C_L es calculado para cada segmento de voz como un porcentaje del máximo valor de la señal, A_{max}. Un valor típico es

$$C_L = 0{,}3A_{max} \tag{2.61}$$

Mientras mayor sea el nivel que se recorte de la señal, mayor será la indicación de la periodicidad. Si usamos un nivel mayor al de la ecuación (2.61) cuando la señal varíe considerablemente durante el segmento de voz mucha



(a) Señal original

(b) Señal recortada

Figura 2.18: *Ejemplo de la aplicación del recorte a una señal de voz*

de la información se perderá. Para evitar este problema y usar mayores porcentajes del valor máximo de la señal se puede hacer lo siguiente:

- Encontrar la máxima amplitud A_1 del primer tercio del segmento de voz, y la máxima amplitud A_2 del ultimo tercio.
- Establecer el umbral como

$$C_L = Kmin[A_1, A_2] \quad K = 0{,}6 - 0{,}8 \tag{2.62}$$

Es posible usar una forma especial de recorte de la señal usando tres niveles

$$\begin{aligned} y(n) &= 1 \quad \text{si } s(n) \geq C_L \\ y(n) &= -1 \quad \text{si } s(n) \leq -C_L \\ y(n) &= 0 \quad c.c. \end{aligned} \tag{2.63}$$

De esta forma los productos de la sumatoria para la función de autocorrelación serán 1,-1 ó 0, reduciendo de esta forma la carga computacional.

Además de las condiciones mencionadas anteriormente, algunas consideraciones adicionales para la implementación del presente detector del tono son las siguientes:

- Se aplica un filtro paso bajas a la señal de voz con una frecuencia de corte de 900 Hz para eliminar armónicas y ruido de alta frecuencia.
- La longitud típica de la ventana es de 30 ms, y las ventanas sucesivas se colocan separadas 10-20 ms (20-10 ms de traslape).

- El mayor pico de la función de autocorrelación es localizado y comparado con un umbral igual a $0{,}3R(0)$. Si se encuentra por debajo del umbral, el segmento es declarado sordo. De otra forma, es declarado sonoro y el periodo del tono es igual al índice del tiempo del mayor pico.
- La decisión sonoro/sordo se encuentra influenciada por la decisión de la trama anterior y la decisión preliminar de la trama posterior: una trama sonora entre dos sordas es declarada sorda, y de manera similar, una trama sorda entre dos sonoras es declarada sonora.

Cuantización de los parámetros del modelo LPC

Los parámetros que son transmitidos para cada intervalo de análisis son:

1. Los coeficientes del predictor, $a_i \quad i = 1, \ldots, P$
2. El periodo de tono, τ_0
3. La ganancia, G
4. La clasificación de la sonoridad: sonora/sorda

El periodo del tono, la ganancia y la clasificación sonora/sorda pueden ser cuantizados y codificados usando cuantizadores escalares. Los coeficientes LPC, por otro lado, pueden ser representados de varias formas, como se mencionó en la sección "Obtención de los parámetros LPC", algunas de las cuales son más apropiadas para la cuantización que otras. Normalmente se evita la cuantización directa de los coeficientes del predictor ya que requieren una gran cantidad de bits por cada coeficiente (entre 8 y 10), pues estos coeficientes son muy sensibles a errores de cuantización. Esto significa que pequeñas diferencias pueden tener un impacto significativo en el filtro de síntesis resultante. Las formas equivalentes que son menos sensibles a la cuantización son:

- Los coeficientes de reflexión, k_i (PARCOR's)
- Las primeras P muestras de la respuesta al impulso de $H(z)$, $h(n)$
- Los coeficientes *Log Area Ratios* (LAR's), g_i
- Los coeficientes de la autocorrelación, $r(i)$
- Los coeficientes cepstral, c_i

En la tabla 2.4 se muestran los parámetros de un analizador LPC junto con los valores típicos usados.

Parámetros	Nombre	Rango	Valores típicos
Orden del predictor	P	1-20	10
Longitud de la ventana	N	160-320	240
Longitud de la trama	I	40-160	120
Factor de preénfasis	a	0.7-0.95	0.8

Cuadro 2.4: *Parámetros del análisis LPC*

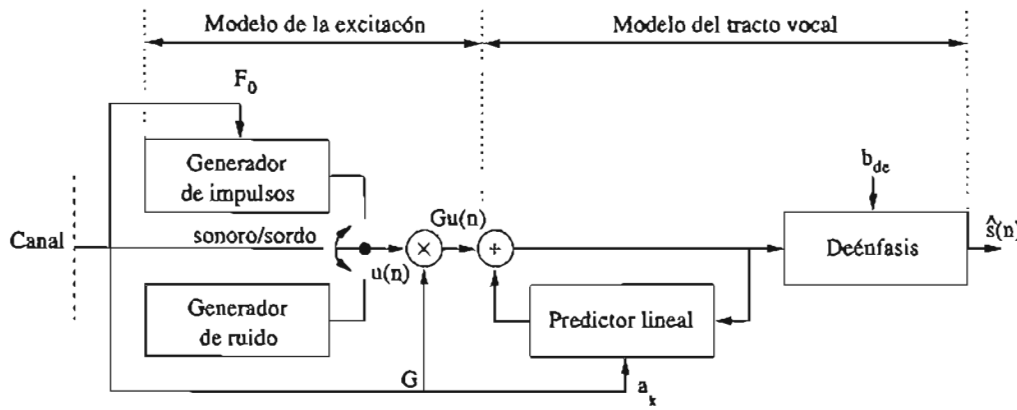


Figura 2.19: *Diagrama de bloques de un sintetizador LPC*

2.2.3. Sintetizador LPC

En el receptor, los parámetros codificados son extraídos y usados para sintetizar la voz codificada. El diagrama de bloques del sintetizador para el vocoder LPC es mostrado en la figura 2.19, y de la misma forma que el analizador LPC, puede ser dividido en dos partes: el modelo de la excitación y el modelo del tracto vocal. El modelo del tracto vocal a su vez posee dos componentes: el filtro del tracto vocal y el filtro de deénfasis.
La operación del sintetizador puede resumirse como sigue:

- La información del canal digital es demultiplexada en sus componentes: tono y decisión sonoro/sordo, ganancia y parámetros LPC.
- Modelo de la excitación
  - El interruptor sonoro/sordo, cuya posición es controlada por la clasificación de la sonoridad, determina el tipo de excitación normalizada $u(n)$, eligiendo ya sea
    - Un generador de pulsos, controlado por τ_0, o
    - Un generador de ruido
  - La ganancia G controla la amplitud de la excitación $u(n)$, produciendo $Gu(n)$
- Modelo del tracto vocal
  - El filtro del tracto vocal es controlado por los parámetros LPC para producir la voz sintetizada, $\hat{s}(n)$, a partir de la excitación $Gu(n)$ (ver ecuación (2.14))
  - El filtro de deénfasis invierte el efecto del filtro de preénfasis del transmisor

Implementación del filtro del tracto vocal

El filtro del tracto vocal puede ser implementado de varias formas. En su forma más simple, el tracto vocal es modelado como un filtro digital recursivo todo polos controlado por los coeficientes LPC (ver figura 2.20).
Sin embargo, es posible implementar el filtro del tracto vocal directamente en términos de los coeficientes PARCOR, como se muestra en la figura 2.21. Este tipo de filtro es análogo al modelo del tubo sin pérdidas de la figura 1.6, donde cada sección del filtro corresponde a una sección del tubo. La señal que viaja en la parte superior del filtro de la figura 2.21 (hacia delante) es parcialmente reflejada hacia atrás, mientras que la señal que viaja en la parte inferior (hacia atrás) es parcialmente reflejada hacia delante. De lo anterior los coeficientes es que los k_i reciben su nombre: coeficientes de reflexión.

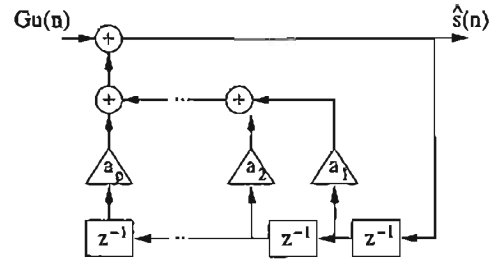


Figura 2.20: *Implementación de la forma directa del filtro del tracto vocal*

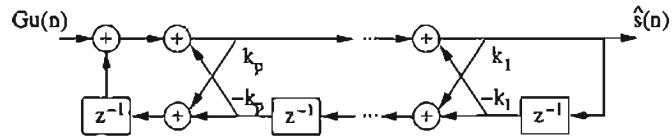


Figura 2.21: *Implementación lattice (rejilla) del filtro del tracto vocal usando PARCORs*

Deénfasis

Para cancelar los efectos del preénfasis aplicado en el analizador LPC se utiliza un filtro de deénfasis de la forma

$$V_{de}(z) = \frac{1}{1 - b_{de}z^{-1}} \tag{2.64}$$

o en el dominio de n:

$$s(n) = s_p(n) + b_{de}s(n-1) \tag{2.65}$$

Aunque las constantes b_{pe} (preénfasis) y b_{de} (deénfasis) usualmente se eligen del mismo valor; sin embargo, pueden usarse valores diferentes, lo que en ocasiones produce una mejor calidad en la voz resultante (por ejemplo, $b_{pe} = 0{,}94$ y $b_{de} = 0{,}74$)

CAPÍTULO 3

Codificadores Híbridos

3.1. Análisis por Síntesis (AbS)

El procedimiento del análisis por síntesis es básicamente una forma de búsqueda exhaustiva dentro de un conjunto de alternativas para encontrar una secuencia óptima de excitación. El procedimiento del análisis de la excitación consiste en la sintetización de la voz a partir de un conjunto de posibles parámetros de la excitación, para de esta forma elegir los parámetros que produzcan la voz sintetizada más similar a la voz original en un sentido perceptual.

Los codificadores de análisis por síntesis pueden considerarse vocoders LPCs con un modelo de excitación más eficiente, aunque una mayor tasa de bits.

Los vocoders basados en el análisis por síntesis usan la mayor parte de la información disponible en la señal de voz para mejorar la calidad y reducir la tasa de bits. Debido a esto, es de esperarse que estos codificadores no sean robustos ante la presencia de errores de bits, ruido acústico, múltiples locutores, o señales que no son de voz. Sin embargo, el modelo del análisis por síntesis ha probado ser muy flexible y más robusto de lo que se podría esperar.

En la figura 3.1 se muestra el diagrama de bloques del procedimiento de análisis para los vocoders de análisis por síntesis.

Este modelo, y por tanto todos los codificadores basados en él, poseen los siguientes componentes principales:

- Generador de la excitación

  El generador de la excitación es capaz de generar K (usualmente 64-1024) secuencias diferentes de excitación, $e_k(n)$. El procedimiento de análisis genera todas las K posibles señales de voz, $\hat{s}_k(n)$, a las que se resta la señal original de voz, para calcular la energía pesada de la señal de error. El analizador debe por tanto realizar K operaciones de síntesis para elegir la secuencia de excitación óptima; por esto el nombre *análisis por síntesis*.

  En términos del modelo descrito por la figura 3.1, los diferentes codificadores de análisis por síntesis se diferencian por el conjunto de secuencias usadas para representar la excitación.

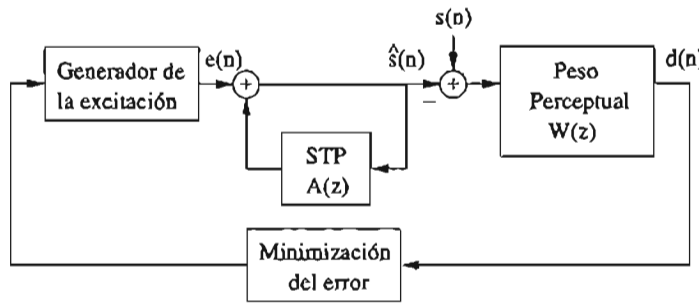


Figura 3.1: *Diagrama de bloques del procedimiento de análisis usado en los codificadores de predicción lineal basados en el análisis por síntesis*

- Predicción lineal de tiempo corto (STP)

  Los parámetros del predictor en tiempo corto se obtienen mediante el análisis de predicción lineal de la voz descrito en el capítulo 2.2.

- Filtro de peso perceptual

  Su principal función es ayudar a enmascarar el ruido de codificación.

- Predicción lineal de tiempo largo (LTP)

  Este predictor es usado para explotar la naturaleza periódica del tono de la voz sonora que no es tomada en cuenta por el predictor de tiempo corto.

3.1.1. Modelo de la excitación

El análisis por síntesis es una técnica de codificación en la que la señal de excitación es determinada por bloques. Se asume que la señal de excitación para cada bloque puede, en general, ser una combinación de diferentes componentes de excitación

$$e(n) = \sum_{k=1}^{M} \beta_k e_k(n) \tag{3.1}$$

donde $e_k(n)$ es la k-ésima componente de la excitación. Las componentes de la excitación pueden ser por ejemplo un pulso, una secuencia de un libro de códigos, o la salida de un predictor de tono (de tiempo largo). El número de componentes de excitación individuales, M, es usualmente pequeño. Un ejemplo típico es CELP, que posee dos componentes (una secuencia de un libro de códigos y un predictor de tono). Cada componente de la excitación es especificado por un índice, γ_k, y una ganancia correspondiente, β_k, los cuales son determinados mediante el análisis por síntesis. Para una k dada y un conjunto dado de K secuencias de excitación, $F_k = f_\gamma(n), \gamma = 1, \ldots, K$, $e_k(n)$ se elige como la $f_{\gamma_k}(n)$ que minimice la diferencia entre la secuencia de voz original y sintética en un sentido medio cuadrático pesado. El índice óptimo, γ_k, y la ganancia asociada, β_k, son transmitidos para que la secuencia de excitación pueda ser reconstruida en el receptor.

Obtención de los parámetros de la excitación. Minimización del error

Asumiendo el modelo de excitación multi-componente de la ecuación (3.1) se usa el análisis por síntesis para obtener el índice y la ganancia de cada componente de la excitación. La figura 3.2 muestra el diagrama de bloques

de este procedimiento de análisis para una clase genérica de codificadores de predicción basados en el análisis por síntesis.

Se observa en esta figura que se han reacomodado los filtros LPC de síntesis y de peso perceptual, esto con el fin de reducir la complejidad en la selección de la excitación en el análisis por síntesis.

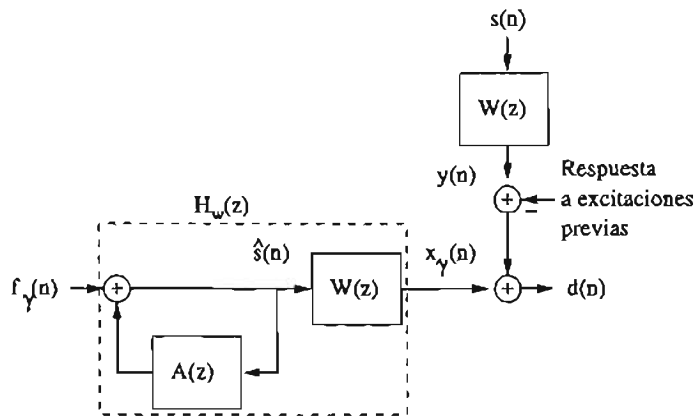


Figura 3.2: *Diagrama de bloques del modelo de análisis de la fuente para una clase genérica de codificadores de predicción basados en análisis por síntesis. $s(n)$ es la señal de voz de entrada. Los filtros de síntesis y de peso perceptual han sido reacomodados*

Cada componente de la excitación se obtiene minimizando la energía de $d(n)$, cuya transformada Z está dada por

$$D(z) = Y(z) - \beta(\gamma)X_\gamma(z) \tag{3.2}$$

$Y(z) = S(z)W(z)$ es la transformada Z de la voz original pesada, y $X_\gamma(z)$ es la transformada Z de la respuesta pesada del sistema a la excitación dada, $f_\gamma(n)$. Para cualquier γ dada, la ganancia correspondiente, $\beta(\gamma)$, puede obtenerse minimizando el error medio cuadrático pesado dado por

$$E_\gamma = \sum_{n=0}^{N-1} d^2(n) = \sum_{n=0}^{N-1} [y(n) - \beta(\gamma)x_\gamma(n)]^2 \tag{3.3}$$

donde

$$x_\gamma(n) = \sum_{i=0}^{\ell-1} h_w(i) f_\gamma(n-i) \tag{3.4}$$

es la respuesta pesada del sistema a la función de excitación dada, $f_\gamma(n)$. La función $h_w(n)$ es la respuesta al impulso pesada. Igualando a cero la derivada de E con respecto a β, la expresión para $\beta(\gamma)$ estará dada por

$$\beta(\gamma) = \frac{\sum_{n=0}^{N-1} y(n)x_\gamma(n)}{\sum_{n=0}^{N-1} x_\gamma^2(n)} \tag{3.5}$$

El error medio cuadrático asociado estará dado entonces por

$$E_\gamma = \left(\sum_{n=0}^{N-1} y(n)^2 \right) - \frac{\left[\sum_{n=0}^{N-1} y(n) x_\gamma(n) \right]^2}{\sum_{n=0}^{N-1} x_\gamma(n)^2} \tag{3.6}$$

Llamado Criterio del Error Medio Cuadrático Pesado (MSPE). El índice óptimo, γ_k, para la componente de la excitación, $e_k(n)$, se obtiene minimizando el error medio cuadrático, E, sobre los posibles valores de γ para el conjunto particular de excitaciones que se use.

Dado cualquier conjunto de excitaciones, las ecuaciones 3.5-3.6 constituyen un método general para la selección de la componente de la excitación, $e_k(n) = \beta_k f_{\gamma_k}(n)$, que minimice la distorsión perceptual pesada.

En la práctica, todos los cálculos se realizan estableciendo condiciones iniciales iguales a cero para los filtros al inicio de cada subtrama. Las condiciones iniciales se obtienen calculando la respuesta a entrada cero de los filtros y sustrayendo esta respuesta a la señal de voz pesada antes de la búsqueda de ciclo cerrado.

Para encontrar la mejor excitación candidata es necesario sintetizar todas las palabras de código de los libros de códigos por lo que la carga computacional es muy elevada. Mientras que el decodificador realiza únicamente una síntesis, el codificador debe realizar K síntesis. Consideremos por ejemplo que se utilizan subtramas de 5 ms, una frecuencia de muestreo de 8000 Hz, y se tiene un libro de códigos con $K = 1024$ excitaciones diferentes; el codificador debe filtrar 1024 señales de excitación de 40 muestras mediante un filtro de orden 10, si no se emplea ningún tipo de algoritmo especial para reducir la carga computacional, ésta técnica conduce a un mínimo de 80 Mops.

Para obtener los parámetros de las componentes de la excitación adicionales, el procedimiento a seguir es remover los efectos de la componente previamente determinada, y realizar el análisis por síntesis en la señal remanente. El motivo del uso de esta técnica secuencial subóptima es la complejidad relacionada con la búsqueda de los parámetros óptimos para todas las componentes de la excitación.

3.1.2. Predicción lineal de tiempo corto

En la predicción lineal de tiempo corto, la señal de voz se asume estacionaria dentro de una ventana de observación. Se usa una respuesta al impulso infinita (IIR) de orden p, llamada LPC, para simular la envolvente espectral de la señal de voz original y emular el efecto de filtrado del tracto vocal. Como se explicó en el capítulo 2.2, el filtro LPC se encuentra determinado por

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{3.7}$$

En el dominio del tiempo, la voz de salida sintetizada, $\hat{s}(n)$ debida a la excitación, $Gu(n)$ (que a partir de ahora llamaremos $r(n)$) se expresa mediante

$$\hat{s}(n) = \sum_{i=1}^{p} a_i s(n-i) + r(n) \tag{3.8}$$

En esta etapa, el bloque correspondiente al generador de la excitación de la figura 3.1 se encuentra deshabilitado; por lo tanto, $r(n)$ simplemente emula una excitación impulso como entrada del filtro LPC.

Como se ha mencionado con anterioridad, el objetivo de la predicción lineal es generar el mejor estimado de los p coeficientes lineales de forma que la voz original, $s(n)$, pueda ser aproximada de manera adecuada por la voz sintética, $\hat{s}(n)$, mediante la excitación $r(n)$ de entrada.

Como los p coeficientes LPC se calculan a partir de la señal de entrada mediante una técnica de ciclo abierto (como el método de la autocorrelación), su complejidad computacional, al compararse con la generación de la excitación, es bastante baja.

3.1.3. Filtro de peso perceptual

Como el error entre las reconstrucciones candidatas y la señal original es la base en el criterio de selección de la excitación más apropiada, una componente importante del codificador de predicción lineal basado en el análisis por síntesis es el filtro de peso del error, $W(z)$, usado para distribuir la energía de la señal de error de codificación.

El filtro de peso perceptual (o filtro de peso del error) hace uso de las propiedades de enmascaramiento del sistema auditivo humano en el dominio de la frecuencia: largos picos en el espectro de una señal pueden enmascarar tonos débiles cercanos de forma que no sean audibles. El moldeado del espectro del error aplicado por el filtro $W(z)$ en los sistemas de análisis por síntesis intenta enmascarar la señal de error de codificación (señal enmascarada) con la señal de voz (señal que enmascara).

La fuerza percibida de la señal de error de codificación es determinada tanto por la potencia de la señal de error como por la distribución espectral de esta señal con respecto a la voz original. Cuando el espectro del ruido es plano (ruido blanco), el ruido percibido se encontrará en las regiones de baja energía del espectro de la señal. Moldeando el espectro del ruido de forma que sea proporcional al espectro de la señal se reduce el ruido percibido y por tanto se mejora la calidad de la voz.

Una forma simple y efectiva de encontrar el filtro de peso perceptual es derivándolo directamente del filtro LPC. El filtro LPC se adapta regularmente a la señal y su modelo describe a la envolvente del espectro de la señal. Por tanto, el filtro de peso perceptual mayormente usado es

$$W_1(z) = \frac{A(z)}{A(z\mu^{-1})} \quad \text{o} \quad W_2(z) = \frac{A(z\mu_1^{-1})}{A(z\mu_2^{-1})} \tag{3.9}$$

donde $0 \leq \mu \leq 1$

o $0 \leq \mu_1 \leq 1 \quad 0 \leq \mu_2 \leq 1 \quad \mu_1 \geq \mu_2$

son las constantes de peso, y $A(z)$ es el predictor en tiempo corto. El valor de la constante de peso μ normalmente se sitúa entre 0.8 y 0.9 para una frecuencia de muestreo de 8 kHz.

La función de peso perceptual del error es pequeña para las frecuencias en que el espectro original posee mayor amplitud, y es mayor en las regiones en que el espectro de la voz es débil. Por lo tanto, el efecto neto del filtro de peso perceptual, $W(z)$, es que las frecuencias correspondientes a los picos del espectro (formantes) son deenfatizadas durante el proceso de selección de la excitación, colocando mayor cantidad de ruido en las regiones en que el espectro $|1/A(z)|^2$ es de alto nivel; mientras que enfatiza el error en los valles del espectro de la voz. Este tipo de peso del error es aplicado en varios algoritmos de codificación LPC. El peso del error disminuye la relación señal a ruido en cierta medida pero aumenta considerablemente la calidad subjetiva.

En la figura 3.3 muestra el espectro LPC de un segmento sonoro de voz y la magnitud de la respuesta del filtro de peso perceptual correspondiente, $|W(f)|$, para $\mu = 0{,}8$.

El filtro de peso perceptual es aplicado en el dominio del tiempo a la diferencia entre la voz original y la sintetizada. Sin embargo, el reacomodo de los filtros mostrado en la figura 3.2 proporciona la ventaja de que la señal de entrada debe ser pesada una sola vez antes de la determinación de la excitación. Además, mediante este reacomodo, los ceros del filtro de peso del error, $W_1(z)$, como se definió en la ecuación (3.9) cancelan los polos del filtro LPC de síntesis. Con lo que la respuesta al impulso del filtro combinado,

$$H_w(z) = H(z)W(z) = \frac{1}{1 - A(z\mu^{-1})} \tag{3.10}$$

referida como filtro pesado LPC de síntesis, puede ser aproximada mediante un filtro FIR relativamente corto,

$$h_w(n) \approx 0, \quad n > \ell \tag{3.11}$$

donde ℓ es típicamente 20. La función $h_w(n)$ se conoce como la respuesta al impulso pesada. Esta aproximación ayuda a disminuir los cálculos computacionales en el proceso de búsqueda del análisis por síntesis.

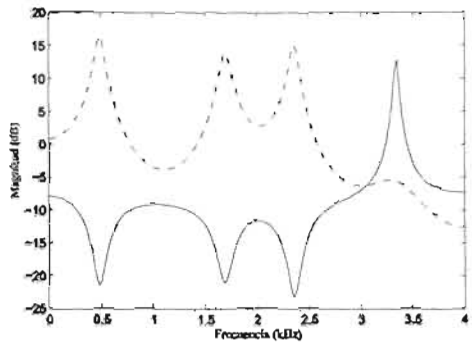


Figura 3.3: *Espectro LPC de un segmento sonoro de voz - - - y la respuesta en frecuencia del filtro de peso del error correspondiente — con* $\mu = 0{,}8$

3.1.4. Predicción lineal de tiempo largo. Libro de códigos adaptable

En los codificadores de predicción lineal, las redundancias de tiempo corto de la señal de voz (aquéllas debidas al efecto acústico del filtro del tracto vocal) son removidas usando un predictor de tiempo corto; como el orden del predictor p usado en los esquemas de predicción lineal generalmente no es muy alto, únicamente se explota la correlación entre muestras cercanas de la señal en el tiempo (correlación en tiempo corto). Sin embargo, como la voz sonora es cuasiperiódica con un periodo L (tono), también posee una correlación entre muestras separadas por distancias mayores (correlación en tiempo largo), lo cual no es explotado cuando el orden del predictor es pequeño. Es entonces claro que la mejor excitación para una subtrama dada debe ser parecida a la mejor excitación obtenida L muestras antes. Si se mantienen en memoria las excitaciones pasadas, es suficiente transmitir al decodificador el valor de L y una ganancia. La memoria que contiene las excitaciones pasadas es llamada Libro de Códigos Adaptable.

La búsqueda en un libro de códigos adaptable, o bien, la predicción en tiempo largo es entonces activada con el fin de reducir la señal residual.

La figura 3.4 muestra el sintetizador (decodificador) LPC con dos filtros de síntesis en cascada.

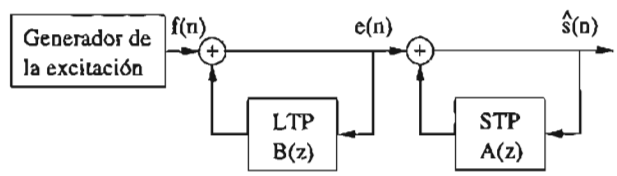


Figura 3.4: *Diagrama de bloques de un sintetizador genérico LPC de análisis por síntesis con predictor de tiempo largo*

El predictor en tiempo corto (STP), $A(z)$, modela la estructura de formantes de la señal de voz, si el filtro del tracto vocal es modelado de forma adecuada, entonces la señal residual representa exactamente la señal de excitación glotal, la cual es periódica por naturaleza. Su periodo es llamado periodo del tono, y el predictor para este periodo del tono es llamado predictor en tiempo largo (LTP), $B(z)$, o simplemente predictor del tono, y por tanto se encargará de modelar la estructura armónica de la voz. La forma general del predictor en tiempo largo es

$$B(z) = \sum_{i=-N_1}^{N_2} \beta_i z^{-\gamma-i} \tag{3.12}$$

donde las β_i's son los coeficientes del predictor en tiempo largo (ganancias), y γ es el retardo del predictor de tiempo largo. El número de coeficientes se elige usualmente entre uno y tres. Típicamente se usa un predictor de primer orden de la forma $\beta z^{-\gamma}$.

El LTP puede ser considerado como una fuente de excitación cuya salida es una componente de la excitación total. La excitación total entonces puede ser expresada como

$$e(n) = \beta e(n - \gamma) + f(n) \tag{3.13}$$

donde $\beta e(n - \gamma)$ es la componente de la excitación generada por el LTP, y $f(n)$ es la suma de todas las demás componentes de la excitación. Los parámetros del LTP pueden ser obtenidos directamente a partir de la señal de voz original. Sin embargo, una forma más eficiente de obtenerlos es usar un método de análisis por síntesis de ciclo cerrado.

El conjunto de búsqueda es un conjunto finito de secuencias de excitación previas que pueden ser representadas por

$$F = e(n - \gamma), \gamma = d_1, \ldots, d_2 \tag{3.14}$$

donde d_1 y d_2 determinan el rango de posibles retardos correspondientes al rango de periodos del tono de la voz, aproximadamente de 5 a 20 ms. La figura 3.5 muestra cómo se construye el conjunto de búsqueda LTP después de cada trama de análisis. Esta estructura es el llamado libro de códigos adaptable. Si el periodo del tono es menor que la longitud de la trama, la sección considerada de la excitación pasada no poseerá la longitud suficiente; en este caso, los últimos γ valores de la excitación pasada son repetidos periódicamente, hasta que se iguale la longitud de la trama.

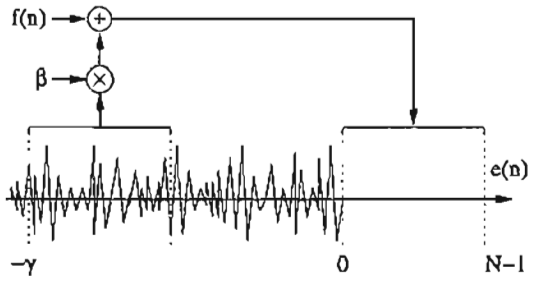


Figura 3.5: *Construcción del conjunto de búsqueda para el LTP. La secuencia óptima es escalada por β y es usada para actualizar el conjunto de búsqueda*

Este conjunto de búsqueda correponde a la memoria del LTP, y cada conjunto está formado por secuencias de N muestras comenzando en la muestra $n = -\gamma$. Por tanto el retardo LTP, γ, es el índice del conjunto cuyas secuencias se encuentran formadas al desplazar una ventana rectangular sobre la memoria del LTP. El criterio de selección del retardo óptimo es el del error pesado mínimo mencionado con anterioridad en la sección 3.1.1.

3.2. CELP

CELP se refiere a una popular familia de algoritmos de codificación de voz que combinan el análisis por síntesis basado en LPC (AbS-LPC) y la cuantización vectorial (VQ). Esta técnica de compresión fue inicialmente introducida por B. Atal en 1985, y desde entonces ha tenido un gran impacto en el campo de la codificación de voz siendo que los conceptos adoptados por CELP han evolucionado para formar la base de una gran variedad de algoritmos de codificación de voz, entre ellos la mayoría de los estándares para telefonía celular de tasa completa y tasa media.

Parte de la investigación actual sobre CELP está orientada a reducir la complejidad y mejorar el desempeño. Otra tendencia actual es el uso de una clasificación de voz, como VAD (*Voice Activity Detection*), para la reducción de la tasa de bits. Los codificadores obtenidos son de tasa variable, con una tasa de bits promedio menor a 3 kbps y la misma calidad que un codificador de tasa fija a 4.8 kbps.

El modelo de excitación por código (CELP) es muy efectivo en el modelado de la excitación con un número muy pequeño de parámetros. CELP es una codificación muy compleja, que requiere 500 millones de operaciones por segundo y entrega una elevada calidad (de *toll* a comunicaciones) con tasas de bits por debajo de los 16 kpbs lo cual es ideal para codificar voz.

A continuación se presentan los pasos básicos de la codificación CELP.

1. Análisis LP de las tramas de la voz original para determinar los coeficientes del filtro todo polos (sección 2.2)
2. Determinación del periodo fundamental (sección 2.2.2)
3. Adaptación del filtro de peso perceptual a la información LP actual y aplicación del filtro adaptado a la señal de voz de entrada (sección 3.1.3)
4. Generación de un filtro en cascada (filtro perceptualmente pesado de síntesis) que consiste en un filtro de síntesis LPC, especificado por los parámetros cuantizados del paso 1, seguido por un filtro de peso perceptual (sección 3.1.1)
5. Sustracción de la respuesta a entrada cero del filtro perceptualmente pesado de síntesis (la respuesta decayente del filtro debida a la entrada anterior) de la señal de voz perceptualmente pesada obtenida en el paso 3
6. Búsqueda en el libro de códigos adaptable (LTP) para encontrar la excitación periódica más adecuada, esto es, cuando el filtro perceptualmente pesado de síntesis es conducido por el mejor vector del libro de códigos adaptable, la salida del filtro en cascada se asemeja más a la señal diferencia obtenida en el paso 5 (sección 3.1.4)
7. Búsqueda en uno o más libros de códigos no adaptables para encontrar los vectores de excitación aleatorios más adecuados que se suman con la mejor excitación periódica obtenida en el paso 6. Esta señal suma resultante conduce al filtro en cascada, produciendo una señal de salida que se asemeja más a la señal diferencia obtenida en el paso 5

Los pasos 1-6 se ejecutan una vez por trama (normalmente de 10-30 ms), mientras que los pasos 7 y 8, que corresponden a los parámetros de la excitación, son ejecutados para cada una de las subtramas (de 2-4 por trama) que constituyen una trama. Durante los pasos 7 y 8 se determinan las ganancias para los vectores que constituyen la excitación. Para cada trama de la señal de entrada, los parámetros del filtro STP y de la excitación se codifican y envían al receptor.

En la figura 3.6 se muestra el diagrama de bloques de un analizador CELP.

Generalmente, el orden del filtro de predicción de tiempo corto es $p = 10$. Durante la obtención de los parámetros LPC, se obtiene también la energía de la trama (generalmente se usa el valor $R(0)$ de la autocorrelación).

Modelo de la excitación. Libros de códigos

La señal de excitación es construida al sumar las salidas pesadas de un número pequeño de libros de códigos (de ahí el nombre *Code Excited*). Típicamente se utilizan de uno a tres libros de códigos que contienen secuencias de la misma duración que la subtrama.

Los libros de códigos usados en el codificador y en el decodificador deben ser los mismos. Los tipos de libros de códigos más comúnmente usados son:

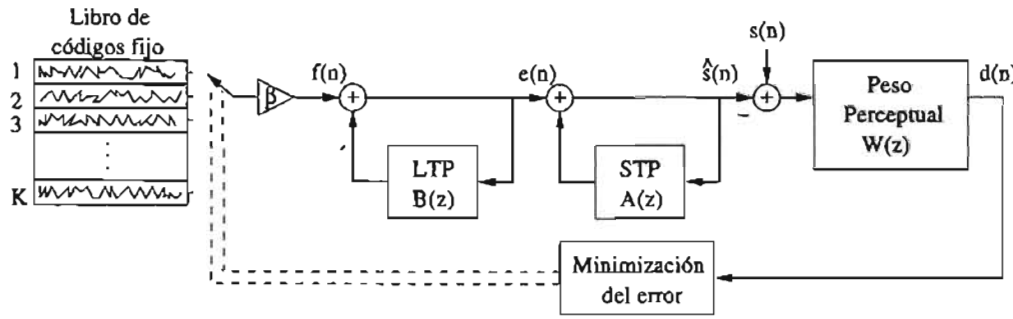


Figura 3.6: *Diagrama de bloques de un analizador CELP*

- Libro de códigos adaptable (sección 3.1.4)

  El contenido de este libro de códigos cambia de una subtrama a otra y depende de la señal de voz que será codificada, por lo que no es conocido a priori. Los valores de d_1 y d_2 que definen el rango de búsqueda son típicamente 20 y 147. Usualmente, L se calcula con una precisión igual a una fracción del intervalo de muestreo, y generalmente se obtiene en dos pasos.

  El primero es el método clásico de estimación del tono de ciclo abierto. El segundo es la búsqueda en el libro de códigos adaptable, que realiza una búsqueda de ciclo cerrado mediante el procedimiento del análisis por síntesis, para mejorar los resultados obtenidos en la búsqueda de ciclo abierto.

- Libro de códigos fijo o estocástico

  Una vez realizado el análisis LPC y la predicción del tono, la señal residual se vuelve aperiódica; la forma tipo ruido de esta señal, aunque carece de información sobre la voz, no se puede pasar por alto ya que en su ausencia, la voz sonará artificial.

  Es por esto que el modelo de excitación por código usa uno o más libros de códigoss (*codebook*) fijos relativamente grandes formados por secuencias de códigos de ruido gaussiano (palabras de código o *codewords*), por lo cual la secuencia seleccionada es llamada comúnmente excitación estocástica.

  Los libros de códigos fijos se encuentran presentes tanto en el codificador como en el decodificador y son conocidos a priori. Deben ser estandarizados para las aplicaciones en telecomunicaciones.

La señal de excitación, $r(n)$, estará dada por la suma de la señal escalada del libro de códigos adaptable (que agrega las periodicidades de tiempo largo durante segmentos sonoros) y una señal escalada a partir de uno o más libros de códigos fijos. Asumiendo un solo libro de códigos estocástico

$$e(n) = \beta_0 e(n - \gamma_0) + \beta_1 v_{\gamma_1}(n) \tag{3.15}$$

donde $v_\gamma[n]$, $\gamma = 1, \ldots, K$, $n = 0, \ldots, N-1$ es la secuencia de N muestras del libro de códigos con el índice γ, donde K es el tamaño del libro de códigos. N es el tamaño de la subtrama de análisis, que usualmente se elige de alrededor de 5 ms.

Típicamente los parámetros del filtro $A(z)$ son determinados en primer lugar y posteriormente se encuentran los índices del libro de códigos, γ_0 y γ_1, así como las ganancias β_0 y β_1. En el transmisor se usa el procedimiento de análisis por síntesis para obtener las palabras de código óptimas. Cada posible entrada del libro de códigos se sintetiza para probar cuál de ellas produce una salida lo más parecida a la voz de entrada en un sentido perceptual. La palabra de código que proporcione la mínima energía pesada de error entre las señales de voz original y reconstruida se selecciona como la secuencia óptima. El índice de la palabra de código óptima (que requiere $log_2(K)$ bits) y el

correspondiente factor de escala son codificados y usados para generar la secuencia de excitación en el sintetizador. En el decodificador la señal de excitación es usada para conducir el filtro de síntesis que modela los efectos del tracto vocal para producir la señal de voz reconstruida, $\hat{s}(n)$. La figura 3.7 muestra el diagrama de bloques de un sintetizador CELP.

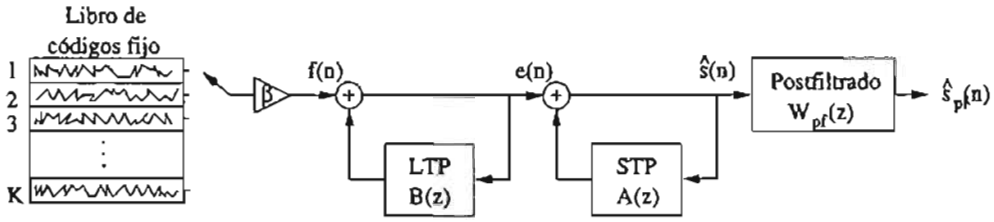


Figura 3.7: *Diagrama de bloques de un sintetizador CELP*

El decodificador recibe entonces de dos a tres tipos de parámetros del codificador:

- Los parámetros de la excitación para cada libro de códigos:
  - El índice (código) del vector en el libro de códigos
  - La ganancia que será aplicada al vector seleccionado
- Los parámetros espectrales, a partir de los cuales se obtienen los coeficientes del filtro de síntesis
- Dependiendo del algoritmo en ocasiones se envía también la energía de la trama

La principal desventaja de la excitación por código es el alto costo computacional del proceso de búsqueda. La mayor parte de la carga computacional en CELP proviene de la búsqueda exhaustiva en los libros de códigos mediante la síntesis de cada una de las secuencias candidatas. Para un libro de códigos de 1024 palabras de longitud 40 se requerirían cerca de 500 millones de operaciones multiplicación-suma por segundo. Por lo anterior se han propuesto varios procedimientos para una búsqueda eficiente del libro de códigos, algunos de ellos obtienen secuencias óptimas ahorrando hasta un orden de magnitud de cálculos computacionales.

Cuando se emplean varios libros de códigos, la búsqueda de la mejor solución es aún más compleja, por ejemplo, para dos libros de códigos de tamaño K, serían necesarias teóricamente K^2 síntesis. En la práctica, se utiliza una solución subóptima, llamada búsqueda iterativa. Consiste en encontrar la mejor solución de un primer libro de códigos, sustrayendo la voz sintetizada de la señal original para obtener una señal diferencia, para posteriormente encontrar la mejor solución de un segundo libro de códigos de forma que se asemeje al vector diferencia, y así sucesivamente. Este método iterativo puede ser mejorado si se ortogonalizan los libros de códigos.

La tabla 3.1 resume los parámetros del codificador CELP y sus valores típicos.

Parámetros	Nombre	Rango	Valores típicos
Orden del predictor	P	1-16	10
Tamaño de la ventana LPC	L	160-360	240
Tamaño de la trama LPC	I	80-240	120
Factor de peso del error	α	0-0.99	0.8
Tamaño de la palabra de código	N	20-60	40

Cuadro 3.1: *Parámetros del análisis y síntesis CELP*

Búsqueda en los libros de códigos. Cálculo de los parámetros de la excitación

En lugar de que los parámetros de los libros de códigos adaptable y fijo sean determinados de manera conjunta para producir un mínimo global en la señal de error pesada, éstos se determinan de manera secuencial.

El retraso y ganancia del libro de códigos adaptable se determinan en primer lugar asumiendo que la señal del libro de códigos fijo es cero. Entonces, dada la señal del libro de códigos adaptable, se encuentran los parámetros del libro de códigos fijo [1]. Esta técnica subóptima se adopta para reducir la complejidad de los codificadores CELP a un nivel razonable. Sin embargo es obvio que debe conducir a una cierta degradación de la voz recontruida.

En primer lugar, el filtro de peso del error en la figura 3.6 se mueve de forma que la entrada y la señal reconstruida, $s(n)$ y $\hat{s}(n)$, sean pesadas de forma separada antes de que se encuentre su diferencia, como se mostró en la figura 3.2. Para un filtro de síntesis todo polos de la forma $H(z) = 1/A(z)$, donde $A(z) = 1 - a_1 z^{-1} - a_2 z^{-2} \ldots a_p z^{-p}$, y filtro de peso del error $A(z)/A(z\mu^{-1})$ donde μ es una constante, el filtro equivalente en cascada es un filtro de síntesis pesado de la forma $1/A(z/\mu^{-1})$. El error pesado $d(n)$ está dado entonces por

$$d(n) = y(n) - \beta(\gamma)x(n) \tag{3.16}$$

donde $y(n)$ es la entrada de voz pesada.

El procedimiento de búsqueda del libro de códigos trata de encontrar los valores de la ganancia β_0 y el retraso γ_0 del libro de códigos adaptable asi como el índice γ_1 y la ganancia β_1 del libro de códigos fijo, que minimicen el error medio cuadrático E_γ sobre la subtrama de longitud N. Este error puede escribirse como

$$E_\gamma = \frac{1}{N}\sum_{n=0}^{N-1}(g^2(n) - T_{\gamma_0\gamma_1}) \tag{3.17}$$

donde

$$T_{\gamma_0\gamma_1} = 2(\beta_0 C_{\gamma_0} + \beta_1 C_{\gamma_1} - \beta_0\beta_1 Y_{\gamma_0\gamma_1}) - \beta_0^2\xi_{\gamma_0} - \beta_1^2\xi_{\gamma_1} \tag{3.18}$$

es el término que debe maximizarse en la búsqueda del libro de códigos. De aquí

$$\xi_{\gamma_0} = \sum_{n=0}^{N-1} e(n-\gamma_0)^2 \tag{3.19}$$

es la energía de la señal del libro de códigos adaptable y

$$C_{\gamma_0} = \sum_{n=0}^{N-1} g(n)e(n-\gamma_0) \tag{3.20}$$

es la correlación entre la señal del libro de códigos adaptable y el objetivo $g(n)$. De manera similar, ξ_{γ_1} es la energía de la señal del libro de códigos fijo, $v_{\gamma_1}(n)$, y C_{γ_1} es la correlación entre ésta y la señal objetivo. Finalmente,

$$Y_{\gamma_0\gamma_1} = \sum_{n=0}^{N-1} e(n-\gamma_0)v_{\gamma_1}(n) \tag{3.21}$$

es la correlación entre las señales de los dos libros de códigos.

[1] Asumiendo que el sistema sólo utiliza un libro de códigos fijo

La técnica usual para encontrar los parámetros del libro de códigos es inicializar $\beta_1 = 0$ en la ecuación (3.18). Entonces para un valor dado de γ_0 puede encontrarse la ganancia óptima β_0 igualando la derivada parcial de $T_{\gamma_0\gamma_1}$ con respecto a β_0 a cero. Usando esto podemos entonces encontrar el valor de $T_{\gamma_0\gamma_1}$ para cada valor de γ_0, y elegir el retraso del libro de códigos adaptable que maximice $T_{\gamma_0\gamma_1}$. Se fijan los parámetros del libro de códigos adaptable y se aplica un procedimiento similar para encontrar los parámetros del libro de códigos fijo, γ_1 y β_1.

Aunque CELP proporciona voz de alta calidad a bajas tasas de bits, el procedimiento secuencial de obtención de los parámetros LPC, y las excitaciones adaptable y estocástica, no es óptimo en términos de SNR. El procedimiento óptimo debería encontrar la mejor de todas las posibles combinaciones de parámetros y las mejores excitaciones adaptable y estocástica. Sin embargo, en la prática es imposible encontrar la mejor combinación debido a que la complejidad es enorme.

Otros puntos a considerar en la codificación CELP

Postfiltrado Es común utilizar en el decodificador CELP un postfiltro adaptable para mejorar la calidad de la voz sintetizada, éste filtro atenúa el ruido de codificación en las regiones perceptualmente sensibles del espectro. La idea del postfiltrado es reforzar los picos de las formantes del espectro de la señal reconstruida con respecto a los valles donde se encuentra presente la mayor cantidad de ruido audible, haciendo que la voz reconstruída sea más clara. El postfiltrado aumenta la calidad subjetiva sin aumentar la tasa de bits.

Dado el filtro de síntesis $1/A(z)$ la función de transferencia del postfiltro es de la forma:

$$H(z) = \frac{A(z\mu_1^{'-1})}{A(z\mu_2^{'-1})} \tag{3.22}$$

y se deriva a partir de 1/A(z).

El postfiltro del decodificador y el filtro de peso perceptual del codificador pueden parecer funcionalmente idénticos. El filtro de peso, sin embargo, influye en la selección de la mejor excitación, mientras que el postfiltro (aplicado únicamente en el decodificador) modela el espectro de la señal sintetizada de forma que sea lo más similar posible al espectro de la voz original, tratando de esconder los efectos de la cuantización tras las formantes de la señal de voz; por lo que un diseño apropiado del mismo reduce la cantidad de ruido audible. A excepción del postfiltrado, las demás operaciones de síntesis del decodificador CELP son duplicadas en el codificador (aunque no al contrario).

Cuantización Vectorial La cuantización vectorial (VQ) consiste en cuantizar de manera conjunta k valores en un solo vector. Si los elementos del vector se encuentran correlacionados, el número de bits requeridos para representarlos se reduce con respecto a la cuantización escalar.

El diagrama de bloques de un cuantizador vectorial simple se encuentra en la figura 3.8. El libro de códigos Y contiene un número K de vectores de código y_i de dimensión N: $y_i = [y_{i1}, y_{i2}, \ldots, y_{iN}]^T$. Cada vector de código es representado de forma única por su índice. La longitud del libro de códigos K, y el número de bits del índice B se encuentran relacionados por $B = log_2 K$.

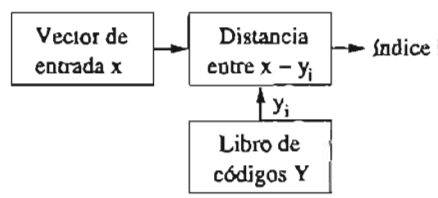


Figura 3.8: *Diagrama de bloques de un cuantizador vectorial simple*

El vector de entrada de N dimensiones $x = [x_1, x_2, \ldots, x_N]^T$ es cuantizado vectorialmente encontrando el vector más similar en el libro de códigos, y representándolo por el índice de dicho vector. El vector más similar es aquél que minimice alguna medida de distorsión, como el error medio cuadrático y el error medio cuadrático pesado.

El número de vectores de código K debe ser lo suficientemente grande de forma que sea posible sustituir cada posible vector de entrada por su vector de códigos más similar sin introducir un error excesivo. Sin embargo, K debe ser limitado para no incrementar excesivamente la complejidad computacional de la búsqueda ni la tasa de bits.

La mayor desventaja de la cuantización vectorial es su alto costo computacional. Comparada con la cuantización escalar, la principal complejidad adicional de VQ radica en la búsqueda del libro de códigos. En una búsqueda completa del libro de códigos, el vector de entrada es comparado con cada uno de los K vectores del libro de códigos, requiriendo K cálculos de distancia computacionalmente costosos.

En la práctica, los sistemas VQ utilizan técnicas subóptimas de búsqueda que reducen los cálculos necesarios sacrificando el funcionamiento del sistema.

En los codificadores CELP, VQ es usada para la cuantización de la señal de excitación, y algunas veces también para modelar la correlación de tiempo largo de la señal de voz (tono) mediante la búsqueda en el libro de códigos adaptable.

Adicionalmente, VQ es usada exitosamente para cuantizar los parámetros espectrales (es decir, cualquier representación de los coeficientes LPC). LSP es la representación más usada para la cuantización de la información espectral. La cuantización vectorial puede explotar efectivamente la correlación intra-trama de los parámetros LSP resultando en una menor distorsión por cuantización que la producida por la cuantización escalar a la misma tasa de bits. Generalmente VQ requiere de 21-26 bits/trama para representar los parámetros LSP, mientras que la cuantización escalar requeriría al menos 32 bits/trama. Sin embargo, a veces se prefiere la cuantización escalar debido a su menor costo computacional, mayor robustez contra variaciones de locutores y ambientes, y puede ser protegida de forma más eficiente contra errores de canal.

La cuantización de los parámetros de los filtros de predicción de tiempo corto y de tiempo largo puede ser realizada usando aproximadamente 80 bits por trama (25 ms), lo que representa una tasa de bits promedio menor a medio bit por muestra para una frecuencia de 8 kHz. La mayor parte de los bits se utiliza en cuantizar la señal de error de predicción, por lo que se puede disminuir significativamente la tasa de bits de la señal de error codificándola por bloques, esto es, aplicando VQ. La longitud del vector no puede ser muy larga, ya que entonces la complejidad aumenta.

Interpolación Aunque la mayor parte del tiempo la voz es cuasiestacionaria por naturaleza, es posible que los coeficientes obtenidos para dos tramas consecutivas sean bastante diferentes, esto es, en los segmentos de transición pueden ocurrir grandes cambios en las características espectrales. Los cambios abruptos en los parámetros LPC entre tramas sucesivas pueden introducir errores (distorsiones audibles) en la voz reconstruída. Para seguir los cambios en el espectro o para suavizar las transiciones espectrales, los coeficientes LPC pueden ser actualizados con mayor frecuencia (disminuyendo la longitud de la trama). Sin embargo, esto puede incrementar la tasa de bits necesaria para codificar los coeficientes. Una alternativa es la interpolación de los coeficientes LPC en el receptor. En el decodificador se interpolan los coeficientes LPC de la trama anterior y actual para producir los coeficientes LPC para cada una de las subtramas que conforman la trama de análisis. Usualmente esta interpolación es lineal, y es realizada en instantes de tiempo de la misma longitud, esto es, a nivel subtrama.

La interpolación no se realiza directamente sobre los coeficientes LPC ya que el filtro de síntesis todo polos puede volverse inestable. De hecho, los problemas de estabilidad en la interpolación son muy similares a los encontrados en la cuantización. La interpolación de los coeficientes de reflexión, LARs y LSPs siempre producen filtros estables. Por tanto, es natural usar la misma representación LPC para la interpolación que la usada para la cuantización. Los parámetros más usados son los LSPs ya que han demostrado funcionar mejor durante la interpolación.

Por tanto, la interpolación de los coeficientes LPC puede proporcionar una calidad de voz mejorada sin tener que transmitir información adicional, esto es, sin aumentar la tasa de bits.

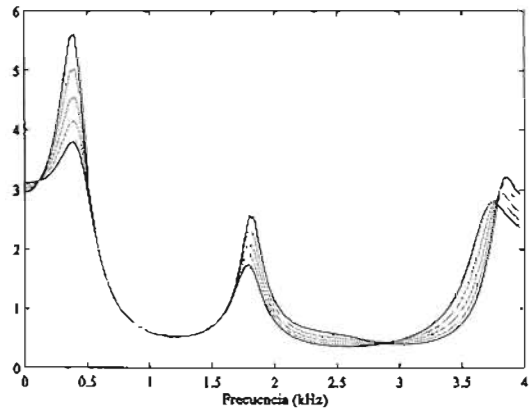


Figura 3.9: *Interpolación de los coeficientes LPC*

Medidas de la calidad Para medir la calidad de la señal sintetizada por un decodificador CELP es posible utilizar el SNR o el SEGSNR. Sin embargo, ninguno de éstos se encuentra específicamente diseñado para modelar los atributos subjetivos de una señal de voz. Aunque son representaciones matemáticas razonablemente confiables de la calidad de la señal, es posible tener dos muestras sintetizadas de voz en las que una de ellas posea un SNR menor que la otra pero una mejor calidad subjetiva. Por lo cual resulta indispensable para la codificación de baja tasa de bits, la inclusión de una medida de calidad subjetiva, como las pruebas de escucha presentadas en la sección 1.7.2.

3.2.1. Estándar CELP FS1016

En 1991 la Administración de Servicios Generales de los Estados Unidos publicó el Estándar Federal 1016, el cual es un algoritmo CELP que opera a 4.8 kbps, originalmente empleado en comunicaciones seguras. Tiene la ventaja sobre otros estándares similares como G.723.1 y G.729 publicados por la ITU, de ser una patente libre.

El flujo de voz de entrada es segmentado en tramas de 30 ms de duración. Cada trama es a su vez dividida en cuatro subtramas de 7.5 ms. De acuerdo con una tasa de muestreo de 8 kHz, hay 240 muestras de voz por trama, y 60 muestras por subtrama. Cuando la voz es analizada, primero se realiza la predicción lineal de tiempo corto sobre toda la trama de voz para extraer 10 coeficientes de predicción lineal. Posteriormente, se realizan las búsquedas en los libros de códigos adaptable (tiempo largo) y estocástico, de manera secuencial, para cada una de las cuatro subtramas. El procedimiento de síntesis de la voz es simplemente el inverso del análisis de la misma. En la tabla 3.2 se muestran los parámetros principales del algoritmo FS1016, así como la asignación de bits correspondiente.

Además de la cantidad de bits mostrada en la tabla 3.2 se usan 200 bps de la siguiente forma: 1 bit por trama para sincronización, 4 bits por trama para corrección de errores y 1 bit por trama para futura expansión del codificador. Lo que nos da una tasa total de

$$1133{,}3 \text{ bps} + 1600 \text{ bps} + 1866{,}67 \text{ bps} + 200 \text{ bps} = 4800 \text{ bps} \tag{3.23}$$

El bit de sincronía (144) comienza en 0 en la primera trama y posteriormente se alterna entre 1 y 0 en las siguientes tramas.

Preprocesamiento

Se aplica un filtrado paso altas como precaución contra componentes indeseables de baja frecuencia. También puede escalarse la señal, dividiéndola por un factor (2) para reducir la posibilidad de sobrecargas de la señal.

	Predictor de tiempo corto	Predictor de tiempo largo	Libro de códigos fijo
Actualización	30 ms 240 muestras	30/4=7.5 ms 60 muestras	30/4=7.5 ms 60 muestras
Orden	10	256×60 1 ganancia	512×60 1 ganancia
Análisis	Autocorrelación Lazo abierto Hamming de 30 ms Sin preénfasis δ = 0,994 Expansión de 15 Hz	MSPE VQ Lazo cerrado peso de 0.8 Rango de 20-147 (con fracciones)	MSPE VQ Lazo cerrado peso de 0.8 Desplazamientos de 2 muestras
Bits por trama	34 LSP [3444433333]	(ndice: 8+6+8+6 ganancia(-1,2):5×4	índice: 9×4 ganancia(+/-):5×4
Tasa de bits	1133.3 bps	1600 bps	1866.67 bps

Cuadro 3.2: *Características de codificación del algoritmo FS1016*

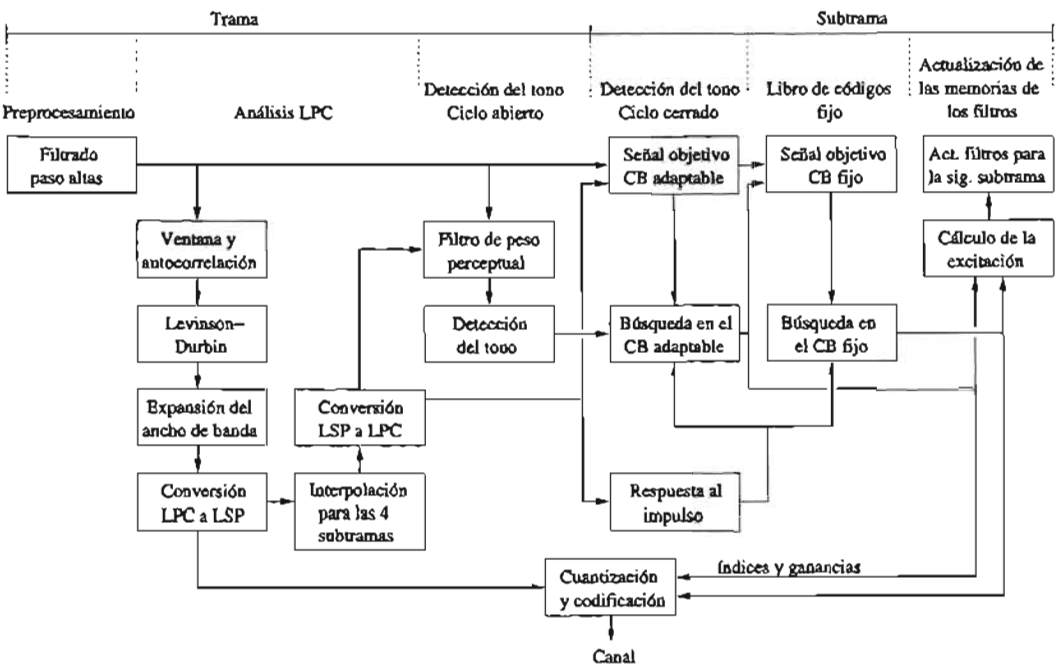


Figura 3.10: *Diagrama de bloques de un analizador CELP FS1016*

Predicción lineal de tiempo corto

La dificultad en esta etapa se encuentra en las pérdidas por cuantización. Como sólo se reserva un número limitado de bits para cada coeficiente, inevitablemente se introducen errores en la señal residual. Además, los coeficientes LPC no garantizan la estabilidad del filtro IIR resultante.

Con el fin de disminuir estos problemas, FS1016 elige cuantizar los coeficientes LSF, que representan un mapeo uno a uno de los coeficientes LPC. Los LSFs se localizan dentro del círculo unitario en el dominio de z. Por tanto, poseen la misma amplitud y diferentes fases. La cuantización aplicada a los LSFs sólo afecta por tanto a las fases resultantes. Como resultado, la estabilidad del sistema es menos vulnerable a la cuantización.

Expansión del ancho de banda

El análisis LPC no estima de manera precisa la envolvente espectral para voz con tonos altos debido al espaciamiento armónico, el cual es bastante amplio. Un método para superar este problema es la expansión del ancho de banda, en la cual cada coeficiente LPC es multiplicado por un factor δ^k ($\delta < 1$), con lo que se mueven los polos del filtro $H(z)$ hacia dentro por un factor δ y se expanden los anchos de banda de todos los polos en la misma cantidad ΔB, dada por

$$\Delta B = -\frac{F_s}{\pi} ln(\delta) \tag{3.24}$$

donde F_s es la frecuencia de muestreo. El filtro de síntesis LPC expandido está dado por

$$H'(z) = \frac{1}{1 - A'(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k \delta^k z^{-k}} \tag{3.25}$$

La expansión del ancho de banda es comúnmente usada en los codificadores de voz, con valores típicos de δ entre 0.996 y 0.988, a una frecuencia de muestreo de 8 kHz, lo que corresponde a una expansión de 10 a 30 Hz. FS1016 emplea una $\delta = 0{,}994$, lo que corresponde a una expansión de $\Delta B = 15Hz$.

Interpolación de los coeficientes LPC

Los coeficientes LPC expandidos son convertidos a un conjunto de parámetros LSP, los cuales son cuantizados usando 34-bits, mediante tablas de cuantización escalar no uniforme independiente.

FS1016 especifica una interpolación pesada para los coeficientes LSF entre dos tramas consecutivas. Como se ilustra en la figura 3.11, los coeficientes LPC de la trama i (anterior) y la trama $i+1$ (actual) se transforman en los coeficientes LSF equivalentes para su transmisión. En la etapa de decodificación, los coeficientes LSF respectivos para las subtramas de la trama de análisis se derivan a partir de

$$\begin{aligned}
\text{LSF para la subtrama 1} &= (7/8) f_i^j + (1/8) f_{i+1}^j \\
\text{LSF para la subtrama 2} &= (5/8) f_i^j + (3/8) f_{i+1}^j \\
\text{LSF para la subtrama 3} &= (3/8) f_i^j + (5/8) f_{i+1}^j \\
\text{LSF para la subtrama 4} &= (1/8) f_i^j + (7/8) f_{i+1}^j
\end{aligned} \tag{3.26}$$

para $j = 1, \ldots, 10$.

Dentro de los 10 coeficientes LSF, 4 de ellos son perceptualmente más sensibles para el oído humano. De acuerdo con esto, FS1016 reserva 4 bits para cada uno de estos 4 coeficientes LSF, y 3 bits para cada uno de los 6 coeficientes restantes. Esto da un total de 34 bits para los 10 coeficientes LSF, lo que consume un ancho de banda de 34 bits/30 ms = 1.133 kbps.

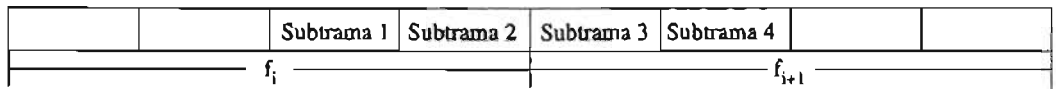


Figura 3.11: *Interpolación de los coeficientes LSF entre tramas consecutivas*

Análisis de ciclo abierto del tono

Para reducir la complejidad de la búsqueda del retardo óptimo del libro de códigos adaptable, el rango de búsqueda es limitado alrededor de un retardo T_{opt} obtenido de un análisis de ciclo abierto del tono. Este análisis es realizado una vez por trama usando la señal de voz pesada.

Predicción lineal de tiempo largo

En FS1016, la señal de excitación que corresponde al periodo del tono óptimo para cada subtrama se selecciona de un libro de códigos adaptable mediante un esquema de estimación de ciclo cerrado que minimize el error pesado medio cuadrático entre la voz original y la reconstruida. El procedimiento es adaptable a la $e(n)$ anterior, esto es, la salida combinada de la búsqueda del libro de códigos estocástico y del libro de códigos adaptable debida a la subtrama anterior. La relación entre la $e(n)$ anterior y la salida del predictor del tono actual [2] es caracterizada mediante el filtro LTP

$$M(z) = \frac{1}{1 - \beta_0 z^{-\gamma_0}} \tag{3.27}$$

donde γ_0 es el periodo del tono y β_0 es la ganancia del tono. La expresión en el dominio del tiempo debida a la entrada $e_{inicial}$ es por lo tanto

$$e(n) = e_{inicial} + \beta_0 e(n - \gamma_0) \tag{3.28}$$

FS1016 busca entonces el mejor estimado del periodo del tono γ_0, de entre 256 candidatos, y su ganancia correspondiente β_0 de forma que se minimice el error medio cuadrático

$$\begin{aligned} \sum_n d^2(n) &= \sum_n [w(n) * (s(n) - \hat{s}(n))]^2 \\ &= \sum_n \left[w(n) * \left(s(n) - \beta_0 e(n - \gamma_0) - \sum_{i=1}^{10} a_i \hat{s}(n - i) \right) \right]^2 \end{aligned} \tag{3.29}$$

Las 256 palabras de código empleadas para minimizar 3.29 son pre-calculadas de acuerdo con un vector 147-dimensional $(\ell_{-147}, \ell_{-146}, \ell_{-146}, \ldots, \ell_{-1})$ que se actualiza a partir de la $e(n)$ anterior. Específicamente, el procedimiento de actualización es remover los 60 componentes más "viejos" de la $e(n)$ anterior, esto es,

$$\ell_{-61} = \ell_{-1},\ \ell_{-62} = \ell_{-2}, \ldots,\ \ell_{-147} = \ell_{-87} \tag{3.30}$$

y

$$\ell_{-1} = r_{59},\ \ell_{-2} = r_{58}, \ldots,\ \ell_{-60} = r_0 \tag{3.31}$$

[2] En esta etapa la búsqueda del libro de códigos estocástico está deshabilitada, por lo que la señal $v(n)$ actual es igual a cero.

El valor inicial de este vector 147-dimensional es simplemente cero.

Las primeras 88 palabras de código con retardos enteros y 60 muestras de longitud se encuentran formadas por

$$\begin{matrix} \ell_{-147} & \ell_{-146} & \dots & \ell_{-88} \\ \ell_{-146} & \ell_{-145} & \dots & \ell_{-87} \\ \vdots & \vdots & \vdots & \vdots \\ \ell_{-60} & \ell_{-59} & \dots & \ell_{-1} \end{matrix} \tag{3.32}$$

Las 40 palabras de código restantes con retardos enteros y de 60 muestras de longitud se forman de la siguiente manera

$$\begin{matrix} \ell_{-59} & \ell_{-58} & \dots & \ell_{-3} & \ell_{-2} & \ell_{-1} & \ell_{-59} \\ \ell_{-58} & \ell_{-57} & \dots & \ell_{-2} & \ell_{-1} & \ell_{-58} & \ell_{-57} \\ \ell_{-57} & \ell_{-56} & \dots & \ell_{-1} & \ell_{-57} & \ell_{-56} & \ell_{-55} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ \ell_{-20} & \ell_{-19} & \dots \ell_{-1}\, \ell_{-20}\, \ell_{-19} \dots \ell_{-1} & \ell_{-20} & \ell_{-19} & \dots & \ell_{-1} \end{matrix} \tag{3.33}$$

Además se obtienen 128 palabras de código adicionales con retardos fraccionarios interpolando las dos palabras de código con retardo entero más cercanas. La ganancia del tono se encuentra en el rango de -1 a 2, y es cuantizada utilizando el mismo número de bits para cada subtrama.

En FS1016, se utiliza un procedimiento diferente para encontrar el retardo del tono en una subtrama par que en una subtrama non. Con base en el criterio de máxima igualdad, m_{opt}, el retardo de tono óptimo, T_{opt} para cada subtrama non es primero seleccionado de los 128 retardos enteros. Posteriormente, FS1016 prueba si la máxima igualdad correpondiente a los retardos $(1/2)T_{opt}$, $(1/3)T_{opt}$ y $(1/4)T_{opt}$ se encuentra a 1 dB de m_{opt}. Si esto ocurre, se actualizan T_{opt} y m_{opt} con el respectivo retraso del tono y coeficiente de igualdad. Finalmente, se examinan los coeficientes de igualdad de las palabras de código con retardos fraccionarios cuyos retardos de tono se encuentran entre $I-3$ e $I+3$, donde I es el índice de la palabra de código correpondiente a T_opt, y se actualiza el retardo óptimo del tono una vez que que se localiza una mayor igualdad que la anteriormente encontrada. El procedimiento anterior se encuentra especificado en FS1016 como modo de búsqueda incompleta. Si en su lugar se usa el modo de búsqueda completa, las 256 palabras de código son examinadas.

Para las subtramas pares, se intenta localizar el desplazamiento del retardo óptimo del tono con relación al retardo óptimo del tono de la trama anterior. Específicamente, si la palabra de código óptima de la subtrama non posee el índice i, FS1016 busca únicamente en las palabras de código cuyos índices se encuentren en el rango $j = min[max(i-31,1),193], j+1, \dots, j+63$ para la subtrama par actual. De nuevo, en el modo de búsqueda incompleta, sólo se prueban los retardos enteros del tono (de entre los 64 candidatos), lo que produce un retardo entero de tono óptimo, T_{opt}, con un coeficiente de igualdad, $m*$. Posteriormente se examinan los retardos fraccionarios en el rango $I-3$ e $I+3$, y se elige la palabra de código con la máxima igualdad de entre los examinados. En este caso, no se examinan los retardos submúltiplos del retardo óptimo.

Una vez que se ha determinado el tono de las subtramas, se calcula el vector del libro de códigos adaptable $e(n-\gamma_0)$ interpolando la señal de excitación pasada para el retardo entero dado y la fracción del mismo. Un retardo fraccionario es representado por su parte entera $int(\gamma_0)$, y su parte fraccionaria frac$= -1, 0, 1$.

En total, FS1016 distribuye 48 bits para los retardos del tono y sus ganancias para las cuatro subtramas de una trama, de los cuales 8 bits y 6 bits son reservados para los retardos en la subtrama non y la subtrama par, respectivamente. La tasa de transmisión resultante es por tanto de 48 bits/30 ms = 1.6 kbps.

Búsqueda en el libro de códigos estocástico

En FS1016 se emplea un libro de códigos estocástico para aproximar la señal de innovación (la señal residual resultante posterior a la aplicación de STP y LTP). Cada palabra de código de este libro de códigos posee su propio índice. Existen un total de 512 palabras de código en el libro. Para acelerar el proceso de búsqueda, también se han especificado libros de códigos de 256, 128 y 64 palabras. Sin embargo, se logra una mejor calidad de voz al usar un libro de códigos con mayor número de palabras.

De manera análoga a la búsqueda en el libro de códigos adaptable, la búsqueda en el libro de códigos estocástico se realiza por subtrama en una operación de ciclo cerrado. También se adopta en este caso el criterio del mínimo error cuadrático. Para facilitar la búsqueda, las 512 palabras de código se obtienen apartir de un arreglo unidimensional de valores (-1, 0, +1) con aproximadamente 77 % ceros. Las palabras de código consecutivas se traslapan excepto en la primera y última de sus componentes. El diseño de un libro de códigos de este tipo posee varias ventajas:

- Sólo se requieren dos bits para representar los valores ternarios +1, -1 y 0
- La multiplicación por +1 y -1 puede ser reemplazada por un cambio de signo, lo que reduce en gran medida la complejidad computacional
- La suma del producto entre un término y cero es equivalente a dejar inalterada la cantidad original, y la posibilidad de encontrar un cero es del 77 %
- Cuando se realizan convoluciones para dos palabras de código consecutivas, la convolución de la segunda palabra de código puede retener los resultados obtenidos a partir de su parte traslapada con la palabra de código anterior, lo que reduce la complejidad computacional

En total, FS1016 distribuye 56 bits para la búsqueda en el libro de códigos estocástico. El índice de la palabra de código seleccionada requiere de 9 bits por cada subtrama. La ganancia estocástica se encuentra entre -1330 y 1330, y cada una de las cuatro ganancias consume 5 bits. Esto resulta en una tasa de transmisión de 56 bits/30 ms = 1.866 kbps.

Actualización de las memorias de los filtros

Es necesario actualizar los filtros de síntesis y de peso perceptual para calcular la señal objetivo de la siguiente subtrama. Una vez que las ganancias han sido cuantizadas, la señal de excitación, $e(n)$, de la trama presente se obtiene usando:

$$e(n) = \hat{\beta}_0 e(n - \gamma_0) + \hat{\beta}_1 v_{\gamma_1}(n) \quad n = 0, \ldots, 39 \tag{3.34}$$

donde $\hat{\beta}_0$ y $\hat{\beta}_1$ son las ganancias cuantizadas de los libros de códigos adaptable y fijo, respectivamente; $e(n - \gamma_0)$ es el vector del libro de códigos adaptable, y $v_{\gamma_1}(n)$ es el vector del libro de códigos fijo. Los estados de los filtros pueden ser actualizados filtrando la señal $r(n) - e(n)$ (la diferencia entre el residuo y la excitación) a través de los filtros $1/\hat{A}(z)$ y $A(z\mu_1^{-1})/A(z\mu_2^{-1})$ para la subtrama de 40 muestras, y guardando los estados de los filtros. Este procedimiento requeriría tres operaciones de filtrado.

Una forma más simple de actualizar los estados de los filtros es la siguiente. Se calcula la señal de voz reconstruída $\tilde{s}(n)$ filtrando la señal de excitación a través de $1/\hat{A}(z)$. La salida del filtro debida a la entrada $r(n) - e(n)$ es equivalente a $s(n) - \hat{s}(n)$. Por lo que el estado del filtro $1/\hat{A}(z)$ está dado por $s(n) - \hat{s}(n)$, $n = 30, \ldots, 39$. La actualización del estado del filtro $A(z\mu_1^{-1})/A(z\mu_2^{-1})$ se puede llevar a cabo mediante el filtrado de la señal de error $s(n) - \hat{s}(n)$ a través de este filtro para encontrar el error perceptualmente pesado E_γ.

3.2.2. Estándares ACELP y VSELP

En esta sección se presentarán las características básicas de codificación de voz (fuente) de algunos de los estándares de tasa completa (FR) basados en algoritmos pertenecientes a la familia CELP:

- ACELP
  - ITU G.729 (CS-ACELP)

    Los codificadores de voz de tasa fija ITU-T G.729(A/B) son usados para comunicaciones multimedia e Internet (VoIP). Usan la codificación ACELP a 8 kbps.
  - ETSI GSM 6.60 (GSM EFR)

    Desarrollado por Nokia y la Universidad de Sherbrooke (USH), meses antes (1996) el mismo codificador fue seleccionado para el sistema US PCS 1900. Este codificador proporciona una mejor calidad que el GSM FR y GSM HR usando la misma tasa de bits que GSM FR y manteniendo la complejidad con relación a GSM HR. Esta calidad mejorada es similar a la de la telefonía fija ante la mayoría de las condiciones típicas de errores.

    La tasa de codificación de voz (fuente) es de 12.2 kbit/s; mientras que para la codificación de canal (protección contra errores) se usan 10.6 kbit/s adicionales, lo que resulta en una tasa total de 22.8 kbit/s. El codificador EFR emplea 0.8 kbit/s más para protección contra errores que el codificador FR en el que la tasa de codificación de la voz es de 13.0 kbit/s.
  - TIA/EIA IS-641 (TIA EFR)

    Codificador de tasa completa mejorado (EFR) más recientemente estandarizado (1996) para el sistema celular digital TDMA (IS-136), fue desarrollado de manera conjunta por Nokia y la Universidad de Sherbrooke (USH). Se encuentra basado en la misma tecnología que el codificador mejorado propuesto por Nokia/USH para los sistemas PCS 1900 EFR y GSM EFR.

    Este codificador utiliza 7.4 kbps para la codificación de voz (fuente) y 5.6 kbps para la codificación de canal (protección contra errores) resultando en una tasa total de canal de 13.0 kbps. Ofrece calidad de voz similar a la de la telefonía fija (ADPCM G.726 32 kbit/s como referencia) y proporciona una mejora sustancial en la calidad de la voz ante una gran variedad de condiciones de ruido y errores. El codificador EFR proporciona no únicamente una mejor calidad sino una mayor protección contra errores que el codificador de tasa completa usando la misma tasa de bits.
- VSELP
  - TIA/EIA IS-136 (TIA FR)

    La TIA adoptó en 1990 el algoritmo VSELP de tasa completa como el estándar de codificación para las comunicaciones digitales celulares en los Estados Unidos. Motorola fue el responsable del diseño y desarrollo del algoritmo VSELP. Adicionalmente, Motorola es propietario de los detalles de la implementación del este algoritmo.

    El vocoder VSELP descrito en el estándar IS-136 (antes IS-54) codifica la voz a una tasa de bits de 7950 bps. Además se utilizan 5050 bps para protección contra errores y sincronización, lo que resulta en una tasa total de bits de 13 kbps.

La representación de los coeficientes LPC de orden 10 mediante LSP's es usada en la mayoría de los estándares de codificación de voz que operan a tasas de bits por debajo de los 16 kbps, como los tres estándares ACELP que se muestran en la tabla 3.3, entre otros, como CELP FS1016 (4.8 kbps), ITU-T G.723.1 (5.3/6.3 kbps) y TIA IS-96 QCELP (tasa variable).

Los estándares antiguos, como GSM 6.10 e IS-54/136, usan los coeficientes de reflexión y LAR para cuantizar la información espectral. Estos codificadores han sido remplazados por los estándares más recientes que usan la

representación LSP, GSM 6.60 e IS-641 respectivamente. En estos nuevos estándares, la representación LSP permite una cuantización más eficiente de la información espectral, usando menos bits y con una mejor calidad. El ahorro en la tasa de bits es usado para mejorar la calidad de la voz, mediante una mejor representación de otros parámetros del codificador y la asignación de más bits para la protección contra errores.

Modelo de la excitación. Estructura del libro de códigos fijo Los codificadores basados en CELP difieren principalmente en la forma en que se generan las excitaciones candidatas (estructura del libro de códigos fijo), además de la tasa de bits, como se observa en la tabla 3.3.

ACELP El libro de códigos fijo se encuentra basado en una estructura algebraica que usa un diseño ISPP. En este libro de códigos, cada vector contiene n pulsos diferentes de cero. Cada pulso puede tener una amplitud +1 ó -1, y puede aparecer en las posiciones mostradas en la tablas 3.4 y 3.5.

El vector del libro de códigos $v(n)$, se construye mediante un vector de ceros de dimensión 40, y colocando los cuatro pulsos unitarios en las posiciones encontradas en las tablas 3.4 y 3.5, multiplicados por sus signos correspondientes.

La búsqueda en el libro de códigos fijo se basa en la minimización del error medio cuadrático entre la señal de entrada pesada y la voz reconstruida pesada.

VSELP El algoritmo VSELP reduce la carga computacional de la búsqueda en el libro de códigos estocástico evitando filtrar todas las posibles excitaciones. El libro de códigos estocástico de tamaño $K = 2^n$ posee una estructura especial. Las palabras de código del libro de códigos estocástico son combinaciones lineales de $n = 7$ vectores base, siendo los pesos de la combinación lineal $+1$ o -1.

Los vectores de los libros de códigos se describen mediante:

$$u_{k,i}(n) = \sum_{m=1}^{M} \theta_{im} v_{k,m}(n) \tag{3.35}$$

donde $k = 1$ ó 2 es el índice del libro de códigos correspondiente, $v_{k,m}$ es el $m^{ésimo}$ vector base del libro de códigos k y $u_{k,i}$ es el $i^{ésimo}$ vector del libro de códigos k. Cada uno de los vectores del libro de códigos posee un índice i, siendo $m = 1$ el bit menos significativo del índice y $m = M$ el bit más significativo, entonces θ_{im} puede definirse como:

Si (el bit m del índice $i = 1$) entonces $\theta = +1$
Si (el bit m del índice $i = 0$) entonces $\theta = -1$

En este caso, únicamente es necesario filtrar los n vectores base, ya que el libro de códigos completo filtrado (2^n vectores) puede ser deducido a partir de los vectores base filtrados usando una pequeña cantidad de cálculos (aproximadamente 10 veces menos que un filtrado completo).

Como VSELP utiliza dos conjuntos de vectores base para generar un espacio de "vectores candidatos", la búsqueda en un libro de códigos fijo CELP corresponde a dos búsquedas en VSELP. Por lo tanto, se tienen 3 fuentes de excitación para el filtro de síntesis LPC. Una de ellas es la correspondiente al libro de códigos adaptable. Las restantes corresponden a las seleccionadas de los 2 conjuntos de códigos de 128 vectores cada uno.

La búsqueda en los 3 libros de códigos se realiza de manera secuencial. Primero se realiza la búsqueda en el libro de códigos adaptable asumiendo que las ganancias de los libros de códigos fijos γ_1 y γ_2 son cero, posteriormente se realiza la búsqueda en uno de los libros de códigos fijos, y finalmente en el segundo libro de códigos fijo, de la misma forma en que se describió en la sección 3.2. Las 3 señales se escalan por sus ganancias para obtener la amplitud adecuada y se suman para ingresar al filtro de síntesis. El resultado se utiliza para actualizar al código adaptable.

		ACELP			VSELP
		G.729	GSM 6.60	IS-641	IS-136
Frecuencia de muestreo		8 kHz			
Longitud de la trama	ms	10	20		
	muestras	80	160		
Longitud de la subtrama	ms	5			
	muestras	40			
Predictor de tiempo corto	Orden	10			
	Análisis	Autocorrelación			FLAT
		Ciclo Abierto			
		Ventana asimétrica de 30 ms			
		Expansión de 60 Hz			Expansión de 80 Hz
	Parámetros LP	LSP			PARCOR
	Bits/trama	18	$[a1_{1-2}\ a2_{1-2}]$=7 $[a1_{3-4}\ a2_{3-4}]$=8 $[a1_{5-6}\ a2_{5-6}]$=9 $[a1_{7-8}\ a2_{7-8}]$=8 $[a1_{9-10}\ a2_{9-10}]$=6	$[a_{1-3}]$=8 $[a_{4-6}]$=9 $[a_{7-10}]$=9	[6554433332]
	Tasa de bits (bps)	1800	1900	1300	1900
Filtro de peso perceptual		$\alpha = 0{,}94 - 0{,}98$ $\beta = 0{,}4 - 0{,}7$	$\alpha = 0{,}9$ $\beta = 0{,}6$	$\alpha = 0{,}94$ $\beta = 0{,}6$	$\gamma = 0{,}8$
Predictor de tiempo largo	Análisis	Ciclo abierto/Ciclo cerrado MSPE VQ			
	Rango	$[19\frac{1}{3}\ 84\frac{2}{3}]$ $\cup$ [85 143]	$[17\frac{3}{6}\ 94\frac{3}{6}]$ $\cup$ [95 143] (Subt. 1 y 3) $T_1+[-5\frac{3}{6}\ 4\frac{3}{6}]$ (Subt. 2 y 4)	$[19\frac{1}{3}\ 84\frac{2}{3}]$ $\cup$ [85 143] (Subt. 1 y 3) $T_1+[-5\ 4]$ (Subt. 2 y 4)	[20 146]
	Tamaño	256 × 40	512 × 40	256 × 40	128 × 40
	Bits/trama (índice)	8+5	9+6+9+6	8+5+8+5	7+7+7+7
	Tasa de bits (bps)	1300	1500	1300	1400
Libro de códigos fijo	Análisis	Ciclo cerrado MSPE VQ			
	Estructura	Algebraica			2 conjuntos de 7 vectores base
		4 pulsos/vector	10 pulsos/vector	4 pulsos/vector	
	posición amplitud	3+3+3+4 1+1+1+1	(2)(3+3+3+3+3) 1+1+1+1+1	3+3+3+4 1+1+1+1	
	Bits/trama	17+17	140	17+17+17+17	2(7+7+7+7)
	Tasa de bits (bps)	3400	7000	3400	2800
Ganancias de los libros de códigos [a]	Bits/trama	7+7	4+4+4+4 5+5+5+5	7+7+7+7	8+8+8+8
	Tasa de bits (bps)	1400	1800	1400	1600
Otros	Bits/trama	1 Paridad			5 Energía
	Tasa de bits (bps)	100	0	0	250
Tasa total de bits (bps)		8000	12200	7400	7950

[a] A excepción de GSM 6.60, las ganancias de los libros de códigos son cuantizadas de manera conjunta usando VQ

Cuadro 3.3: *Características de codificación de algunos estándares CELP de tasa completa*

Pulso	Signo	Posiciones
p_0	$s_0 = \pm 1$	$m_0 = 0, 5, 10, 15, 20, 25, 30, 35$
p_1	$s_1 = \pm 1$	$m_1 = 1, 6, 11, 16, 21, 26, 31, 36$
p_2	$s_2 = \pm 1$	$m_2 = 2, 7, 12, 17, 22, 27, 32, 37$
p_3	$s_3 = \pm 1$	$m_3 = 3, 8, 13, 18, 23, 28, 33, 38$
		$4, 9, 14, 19, 24, 29, 34, 39$

Cuadro 3.4: *Estructura del libro de códigos fijo de CS-ACELP e IS-641*

Pulsos	Signo	Posiciones
p_0, p_1	$s_0 = \pm 1$	$0, 5, 10, 15, 20, 25, 30, 35$
p_2, p_3	$s_2 = \pm 1$	$1, 6, 11, 16, 21, 26, 31, 36$
p_4, p_5	$s_4 = \pm 1$	$2, 7, 12, 17, 22, 27, 32, 37$
p_6, p_7	$s_6 = \pm 1$	$3, 8, 13, 18, 23, 28, 33, 38$
p_8, p_9	$s_8 = \pm 1$	$4, 9, 14, 19, 24, 29, 34, 39$

Cuadro 3.5: *Estructura del libro de códigos fijo de GSM EFR*

CAPÍTULO 4

Simulaciones en Matlab. Resultados

Los resultados que se muestran a continuación se obtuvieron a partir de simulaciones en Matlab de cada uno de los codificadores presentados en los capítulos anteriores.

Como señal de entrada se utilizó un archivo de audio .wav, con la palabra "señales", la cual fue codificada mediante PCM de 16 bits y muestreada a 8 kHz, por lo que la tasa de bits correspondiente es de

$$F_S B = 8000 \text{ muestras/s} \cdot 16 \text{ bits/muestra} = 128 \text{ kbps} \tag{4.1}$$

Con el fin de poder apreciar el funcionamiento de cada uno de los codificadores se mostrarán las gráficas de algunas etapas en el análisis y síntesis de la señal de voz para una trama de la misma. También podrán observarse las gráficas de la señal completa de entrada y la de salida con sus respectivos espectros para proporcionar un medio de comparación visual. Además de las gráficas anteriores se medirá la calidad objetiva usando el SNR, y para el caso de CELP también la calidad subjetiva mediante MOS.

4.1. Codificación Diferencial

4.1.1. DPCM

Se simuló un codificador DPCM con predictor fijo de primer orden y cuantizador uniforme fijo. Los parámetros de este codificador se encuentran en la tabla 4.1

Se hicieron pruebas para 6, 7 y 8 bits, lo que representa 64, 128 y 256 niveles de cuantización, y una tasa bits de 48, 56 y 64 kbps respectivamente. Como la tasa de bits original es de 128 kbps, la tasa de reducción de información es de

$$\begin{aligned} 48/128 &= 0{,}375 \text{ o una reducción de información del } 62{,}5\,\% \\ 56/128 &= 0{,}4375 \text{ o una reducción de información del } 56{,}25\,\% \\ 64/128 &= 0{,}5 \text{ o una reducción de información del } 50\,\% \end{aligned}$$

Parámetros	Nombre	Valor
Orden del predictor	P	1
Coeficiente del predictor	a	0.5
Número de bits	B	6-8
Máximo nivel del cuantizador	S_{max}	32767
Tipo de cuantizador	-	*midtread* uniforme

Cuadro 4.1: *Parámetros del DPCM implementado*

Tasa de bits (kbps)	SNR (dB)	SEGSNR (dB)	Compresión (%)
48	18.22	4.76	62.5
56	24.24	10.77	56.25
64	30.05	16.65	50

Cuadro 4.2: *SNR para el DPCM implementado*

Los resultados en SNR y SEGSNR al variar el número de bits empleados por el codificador DPCM, utilizando el archivo señales.wav, son los siguientes:

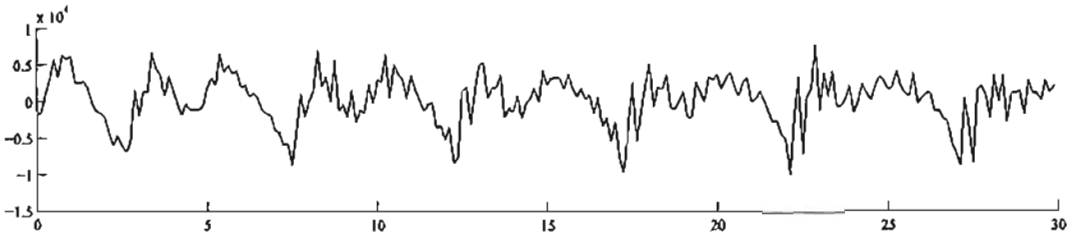
Para comparar estos resultados con la codificación PCM, se aplicó una compresión a la señal original, esto es, se redujo el número de niveles del cuantizador, usando el mismo número de bits que para DPCM. Los resultados fueron:

Tasa de bits (kbps)	SNR (dB)	SEGSNR (dB)	Compresión (%)
48	13.80	0.69	62.5
56	20.06	6.76	56.25
64	25.44	11.94	50

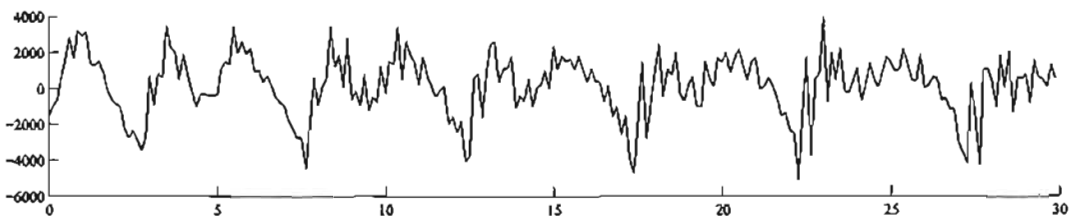
Cuadro 4.3: *SNR para PCM*

En las tablas 4.2 y 4.3 se puede observar claramente el incremento de 6 dB por cada bit agregado, correspondiente a los sistemas PCM. Además, la configuración diferencial en este caso produjo una mejora en el SNR de alrededor de 4 dB con respecto a PCM.

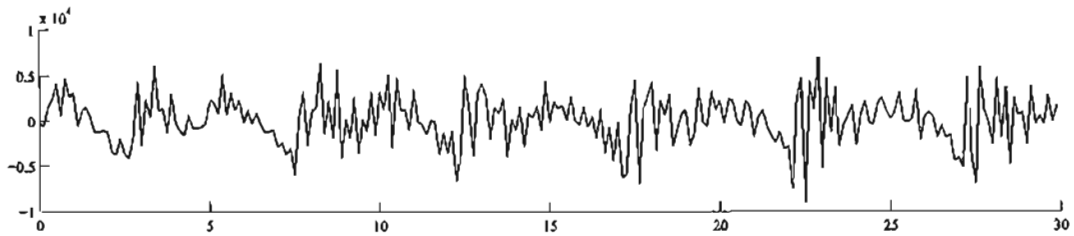
A continuación se muestran algunas gráficas de un segmento de la señal utilizada junto con algunos resultados parciales de la aplicación del codificador DPCM de 6 bits (figuras 4.1). Además se muestra la señal original completa y la señal resultante junto con sus espectros (figura 4.2).



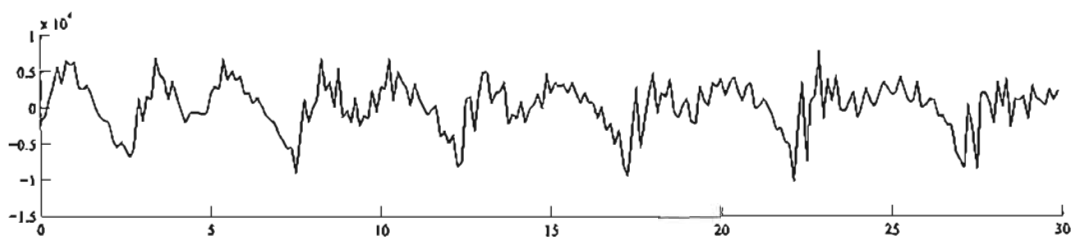
(a) Señal original



(b) Señal estimada

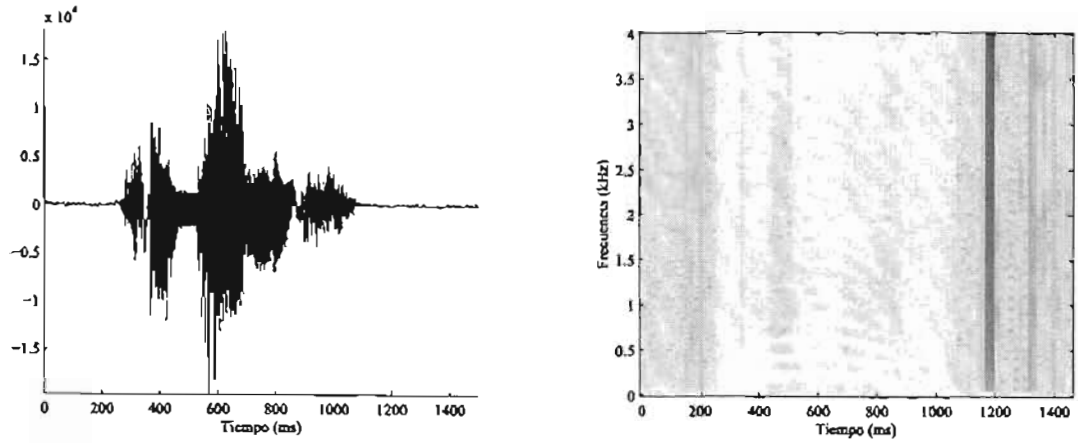


(c) Señal diferencia

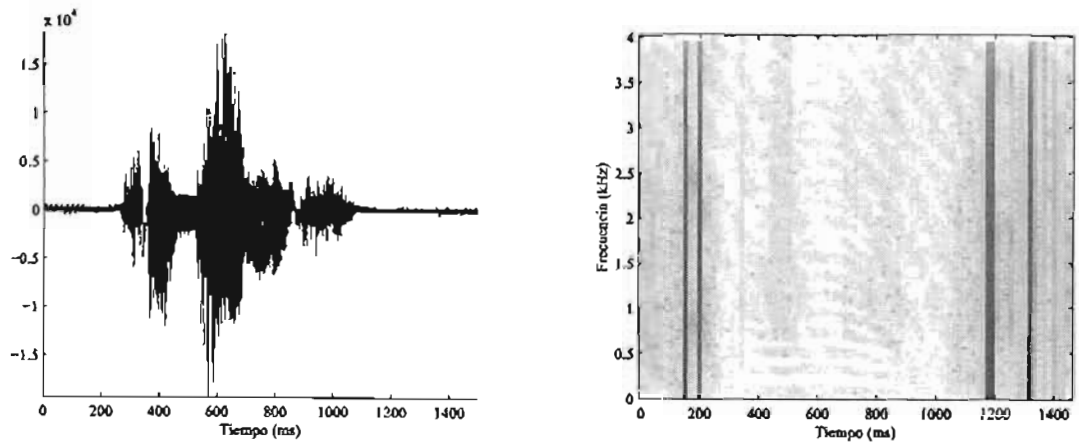


(d) Señal sintetizada

Figura 4.1: *Algunas etapas del DPCM implementado*



(a) Señal original



(b) Señal sintetizada

Figura 4.2: *Resultado de la aplicación del DPCM implementado*

Código 4.1: *Archivo de Matlab DPCM.m*

```
%-----------------------------------------------------
%-----------------------------------------------------
% CODIFICADOR DPCM
%-----------------------------------------------------
%-----------------------------------------------------
archivo = input('Nombre del archivo de audio .wav:','s');
niveles = input('Niveles de cuantizacion:');
fid = fopen(strcat(archivo,'.wav'),'r');
header = 44;
x_temp = fread(fid,inf,'int16');
x = x_temp(header+1:end);
fs = x_temp(13);
offset = mean(x);
x = x-offset;
bits = x_temp(9);
Smax = 2^(bits-1)-1;

%-----------------------------------------------------
% COEFICIENTE DEL PREDICTOR LINEAL DE 1er ORDEN
%-----------------------------------------------------
a = 0.5;

%-----------------------------------------------------
% CUANTIZADOR FIJO
%-----------------------------------------------------
diferencia = filter([1 -a],1,x);
Smax = max(abs(diferencia));
tamano_paso = Smax*2./(niveles);

particion = -Smax+tamano_paso/2:tamano_paso:Smax-tamano_paso/2;
sig = (-Smax+tamano_paso):tamano_paso:(Smax-tamano_paso);

%-----------------------------------------------------
% CODIFICACION DPCM
%-----------------------------------------------------
x_estimada(1) = x(1);

for n=1:length(x)
    d(n) = x(n)-x_estimada(n);
    [indice(n),d_cuantizada(n)] = quantiz(d(n),sig,particion);
    x_cuantizada(n) = d_cuantizada(n)+x_estimada(n);
    x_estimada(n+1) = a.*x_cuantizada(n);
end


%-----------------------------------------------------
% Se envia al decodificador:
% indice -- El indice del cuantizador: respresenta la Diferencia Cuantizada
%-----------------------------------------------------
d_cuantizada_salida = particion(indice+1);
x_salida = filter(1,[1 -a],d_cuantizada_salida);

[SNR,SEGSNR] = SNRs(x,x_salida,fs)
```

4.1.2. ADPCM

Se simuló un codificador ADPCM-FF con predictor fijo de primer orden y cuantizador uniforme fijo con adaptación de ganancia. Los parámetros de este codificador se encuentran en la tabla 4.4

Parámetros	Nombre	Valor
Orden del predictor	P	1
Coeficiente del predictor	a	0.5
Tamaño de la ventana rectangular	M	100
Número de bits	B	6-8
Máximo nivel del cuantizador	S_{max}	32767
Tipo de cuantizador	-	*midriser* uniforme
Escalamiento de la ganancia	G_0	-

Cuadro 4.4: *Parámetros del DPCM implementado*

De la misma forma que con DPCM, se varió el número de bits utilizado por el cuantizador, con el fin de mostrar las variaciones en el SNR.

Como se trata de un codificador ADPCM-FF, se debe enviar al decodificador la información sobre la adaptación de la ganancia, $G(n)$, por lo que la tasa de bits será mayor a la usada por el DPCM presentado con anterioridad. El aumento en la tasa de bits será de aproximadamente el 1 % de la tasa total, por tanto, para una frecuencia de muestreo de 8 kHz

Si usamos 10 bits para codificar la ganancia, $G(n)$, como ésta debe enviarse cada $M = 100$ muestras, el aumento en la tasa de bits sería:

$$\frac{8000 \text{ muestras/s}}{100 \text{ muestras/G}} \cdot 10\text{bits/G} = 800 \text{ bps} \tag{4.2}$$

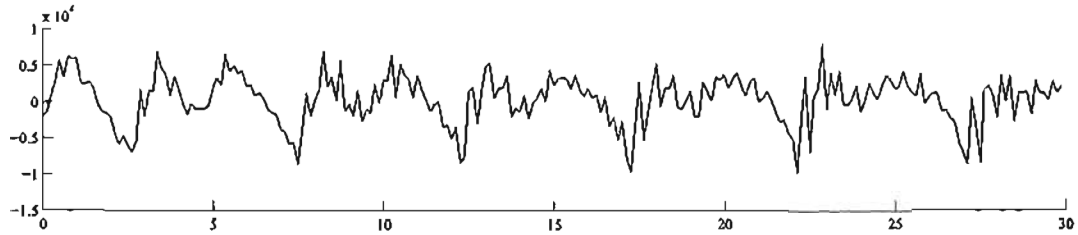
Lo cual representa menos del 1 % de la tasa empleada por DPCM, por lo que la tasa de compresión será aproximadamente igual. La tasa total de información será entonces de 48.8, 56.8 y 64.8 kbps para 6, 7 y 8 bits repsectivamente

En la tabla 4.5 de nuevo se observa el incremento de 6 dB por bit agregado mas un aumento de >5 dB en SNR con respecto a DPCM, debido a la adaptación de la ganancia; por tanto, ADPCM mejora en 9 dB el SNR de PCM. Observando ahora el SEGSNR, tenemos que el incremento con respecto a DPCM es aún mayor, >18 dB, lo que indica que no sólo mejora el funcionamiento global del sistema, sino que la adaptación mejora considerablemente la operación del codificador tanto en segmentos de alta como de baja energía. De lo anterior se concluye que el ADPCM-FF de n bits implementado funciona aproximadamente igual que PCM de $n + 2$ bits, y que DPCM de $n + 1$ bit.

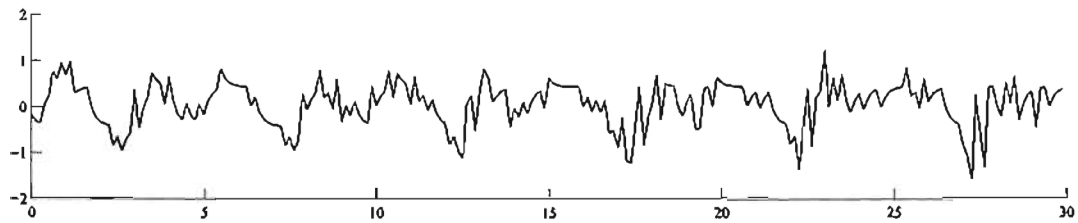
A continuación se muestran algunas gráficas de un segmento de la señal utilizada junto con algunos resultados parciales de la aplicación del codificador ADPCM-FF de 4 bits (figuras 4.1). Posteriormente se muestra la señal original completa y la señal resultante junto con sus espectros (figura 4.2).

Tasa de bits (kbps)	SNR (dB)	SEGSNR (dB)
32.8	11.58	11.63
40.8	17.60	17.54
48.8	23.68	23.95
56.8	29.57	29.18
64.8	35.62	35.71

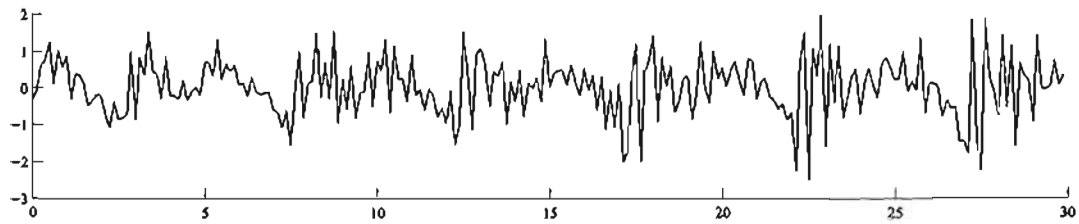
Cuadro 4.5: *SNR para el ADPCM implementado*



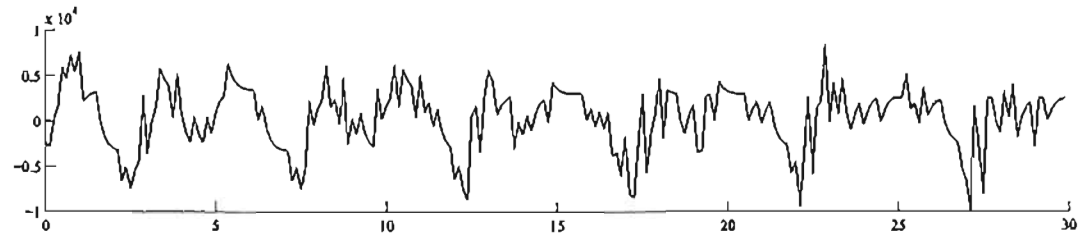
(a) Señal original



(b) Señal estimada

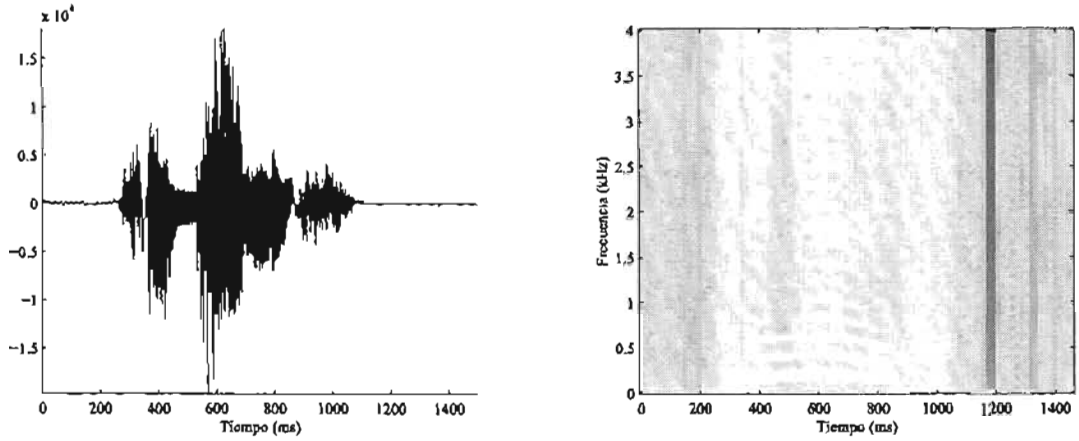


(c) Señal diferencia

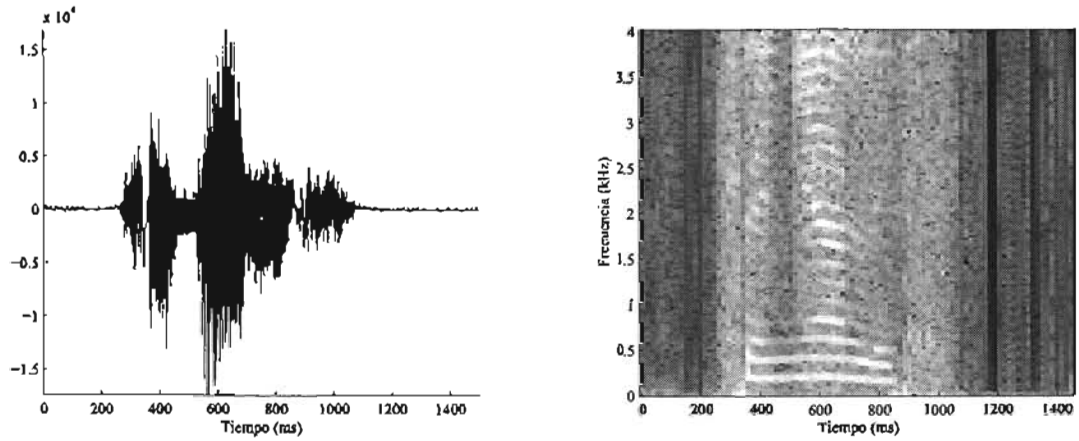


(d) Señal sintetizada

Figura 4.3: *Algunas etapas del ADPCM implementado*



(a) Señal original



(b) Señal sintetizada

Figura 4.4: *Resultado de la aplicación del ADPCM implementado*

Código 4.2: *Archivo de Matlab ADPCMFF.m*

```
%-----------------------------------------------------------
%-----------------------------------------------------------
% CODIFICADOR APCM
%-----------------------------------------------------------
%-----------------------------------------------------------
archivo = input('Nombre del archivo de audio .wav:','s');
niveles = input('Niveles de cuantizacion:');
fid = fopen(strcat(archivo,'.wav'),'r');
header = 44;
x_temp = fread(fid,inf,'int16');
x = x_temp(header+1:end);
fs = 8000;
offset = mean(x);
x = x-offset;
Smax = 2^15-1;

%-----------------------------------------------------------
% CALCULO DE LA VARIANZA
%-----------------------------------------------------------
m=100;

for i=1:ceil(length(x)/m)
    inicio=m*(i-1)+1;
    fin=inicio+m-1;
    if fin<length(x)
        dst(i)=sqrt(sum(x(inicio:fin).^2)./m);
    else
        dst(i)=sqrt(sum(x(inicio:end).^2)./m);
    end
end

%-----------------------------------------------------------
% CODIFICACION APCM
%-----------------------------------------------------------
G = 1./dst;

%-----------------------------------------------------------
% COEFICIENTE DEL PREDICTOR LINEAL DE 1er ORDEN
%-----------------------------------------------------------
a = 0.5;

%-----------------------------------------------------------
% CODIFICACION DE LA GANANCIA
%-----------------------------------------------------------
nivelesG = 2^10;
tamano_pasoG = (max(G)-min(G))./(nivelesG-1);
particionG = min(G):tamano_pasoG:max(G);
sigG = min(G)+tamano_pasoG:tamano_pasoG:max(G)-tamano_pasoG;

for n=1:length(G)
    [indiceG(n),G_cuantizada(n)] = quantiz(G(n),sigG,particionG);
    inicio = (n-1).*m+1;
    fin = inicio+m-1;
    if fin>length(x)
        fin=length(x);
    end
    xG(inicio:fin) = G_cuantizada(n).*x(inicio:fin);
end

%-----------------------------------------------------------
% CUANTIZADOR FIJO
%-----------------------------------------------------------
```

```
diferencia = filter([1 -a],1,xG);
Smax = max(abs(diferencia));
tamano_paso = Smax*2./(niveles);

particion = -Smax+tamano_paso/2:tamano_paso:Smax-tamano_paso/2;
sig = (-Smax+tamano_paso):tamano_paso:(Smax-tamano_paso);

%-----------------------------------------
% CODIFICACION ADPCM
%-----------------------------------------
x_estimada(1) = 0;

for n=1:length(x)
    ind = cell(n/m);
    d(n) = G_cuantizada(ind).*x(n)-x_estimada(n);
    [indice(n),d_cuantizada(n)] = quantiz(d(n),sig,particion);
    x_cuantizada(n) = d_cuantizada(n)+x_estimada(n);
    x_estimada(n+1) = a.*x_cuantizada(n);
end


%-----------------------------------------
% Se envia al decodificador:
% indice - Diferencia Cuantizada
% indiceG - Ganancia
%-----------------------------------------
G_salida = particionG(indiceG+1);
d_cuantizada_salida = particion(indice+1);

x_salida(1) = d_cuantizada_salida(1)./G_salida(1);
ind_ant = 1;

for n=2:length(x)
    ind = cell(n/m);
    x_salida(n) = (d_cuantizada_salida(n)+a.*x_salida(n-1).*G_cuantizada(ind_ant))./G_cuantizada(ind);
    ind_ant = ind;
end

[SNR,SEGSNR]= SNRs(x,x_salida,fs)
```

4.2. Vocoder LPC

Se simuló un vocoder LPC con los parámetros mostrados en la tabla 4.6

Parámetros	Nombre	Valor
Orden del predictor	P	10
Tipo de ventana		Hamming
Longitud de la ventana	N	30 ms
Longitud de la trama	I	15 ms
Factor de preénfasis	b_{pe}	0.9375
Factor de deénfasis	b_{de}	0.8

Cuadro 4.6: *Parámetros del vocoder LPC implementado*

De lo anterior se observa un traslape entre tramas sucesivas del 50 %: $\frac{I}{N} = \frac{15\text{ms}}{30\text{ms}} = 0{,}5$

En este caso la tasa de información es mucho menor que la usada por los codificadores DPCM, ya que únicamente se envían los coeficientes LPC, la ganancia, la frecuencia fundamental y la clasificación sonoro/sordo de la trama de análisis. Para codificar los coeficientes LPC típicamente se utilizan alrededor de 40 bits por trama; para codificar la ganancia de la trama se usan 5 bits, mientras que el interruptor sonoro/sordo utiliza únicamente 1 bit (0 ó 1); además la frecuencia fundamental de la trama se representa mediante 7 bits. Lo que resulta en una tasa de bits de:

$$40 + 5 + 1 + 7 = 53 \text{ bits/trama} \cdot \frac{1 \text{ trama}}{30 \text{ ms}} = 1766{,}6 \text{ bps} \tag{4.3}$$

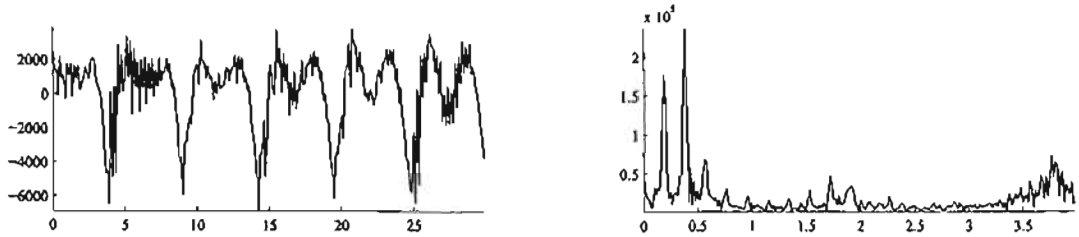
Lo que representa una tasa de reducción de información de

$$1{,}77/128 = 0{,}0138 \text{ o una reducción de información del } 98{,}62\,\% \tag{4.4}$$

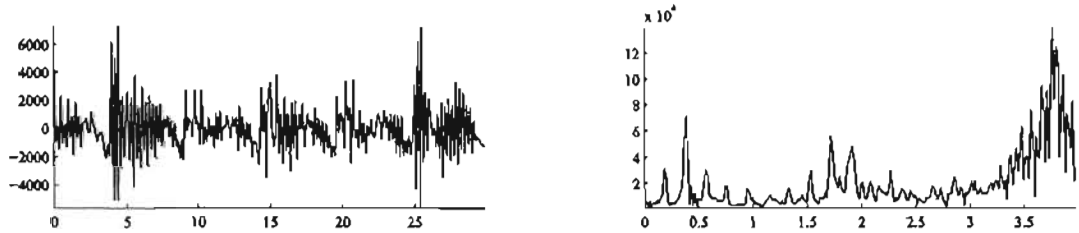
Claramente se observa la mayor compresión obtenida mediante este codificador. Sin embargo, como es de esperarse, la calidad de la voz sintetizada es mucho menor también (tipo sintética). De hecho la relación señal a ruido en este caso es negativa, pero como la tasa de bits es demasiado baja, el SNR no nos proporciona una medida adecuada para la calidad de la voz sintetizada.

Podría pensarse que para este tipo de vocoder sería posible mejorar la calidad de la voz reconstruida aumentando el orden del predictor lineal. Sin embargo, éste se encarga de modelar la estructura de formantes de la voz, y el principal motivo por el que la calidad es tan baja se debe al modelo de la excitación, esto es, al responsable de modelar la estructura armónica de la voz en un vocoder LPC.

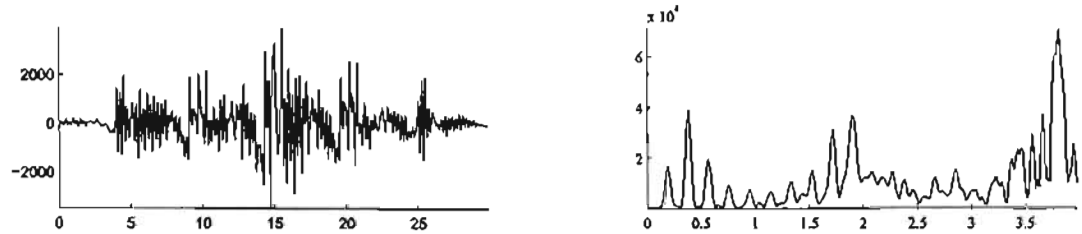
A continuación se presentan algunas gráficas obtenidas mediante el codificador LPC descrito, para una trama de la señal de voz utilizada. Así como la gráfica en el tiempo y el espectro de la señal total de salida. En estas gráficas se puede observar que el codificador no genera una representación en el tiempo tan buena como lo haría un codificador DPCM ya que ese no es su objetivo; sin embargo, opera de manera bastante adecuada en la frecuencia, es por esto que la señal de salida mantiene su inteligibilidad pero se escucha bastante sintética.



(a) Segmento original



(b) Aplicación de preénfasis



(c) Aplicación de la ventana de Hamming

Figura 4.5: *Preprocesamiento de un segmento de voz para el vocoder LPC implementado*

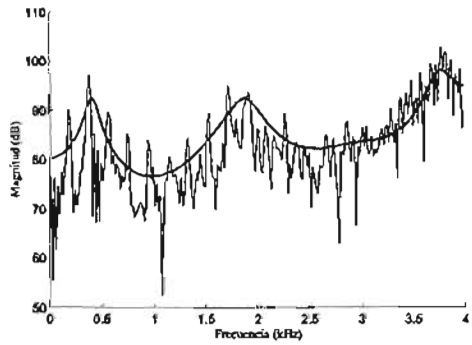


Figura 4.6: *Espectro LPC*

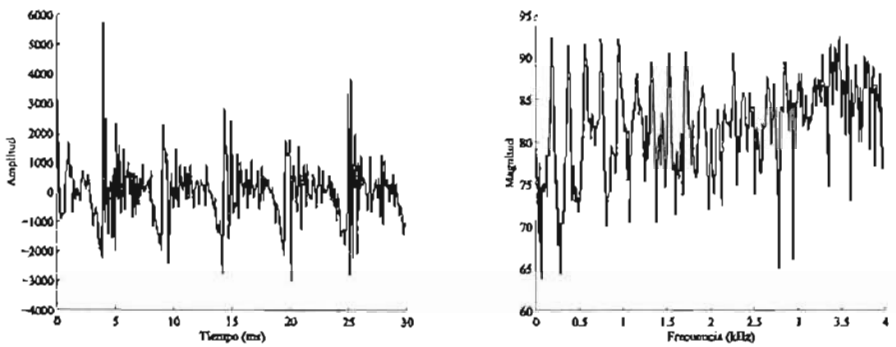
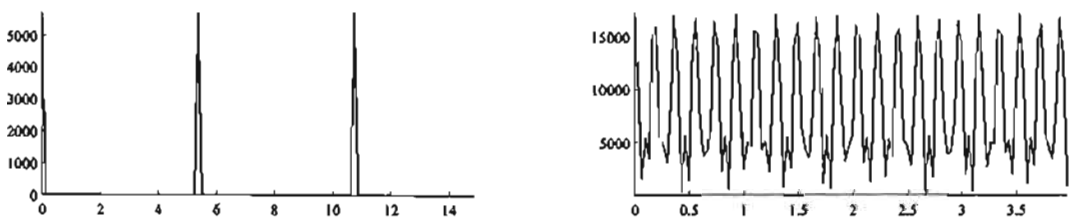
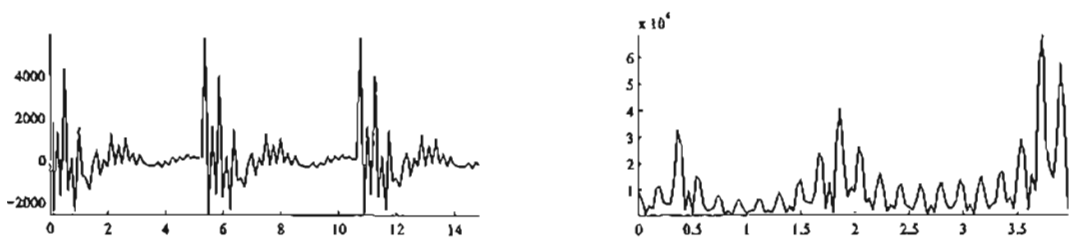


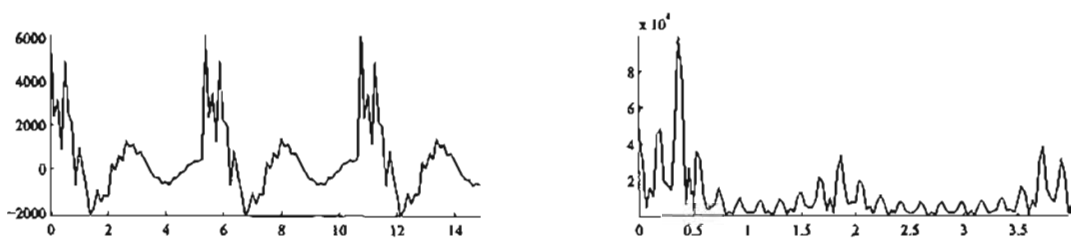
Figura 4.7: *Residuo del segmento de voz*



(a) Excitación $Gu(n)$

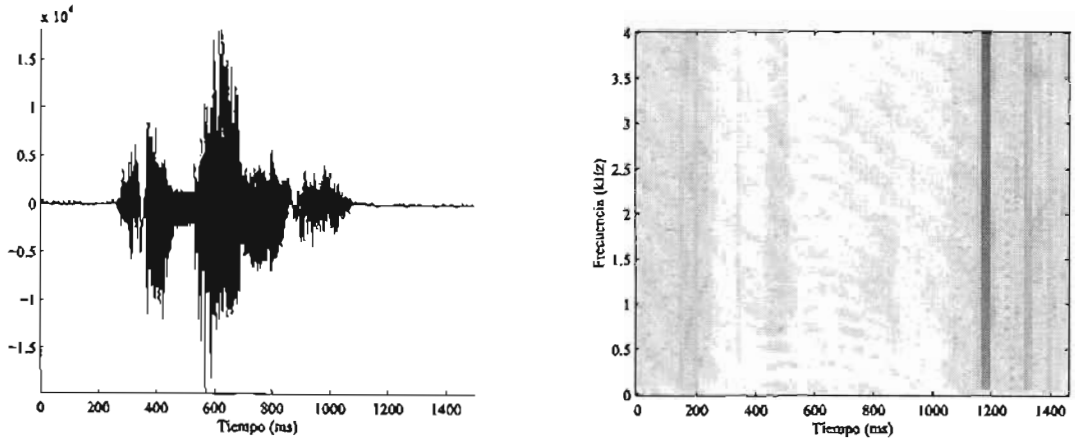


(b) Segmento de voz sintetizado

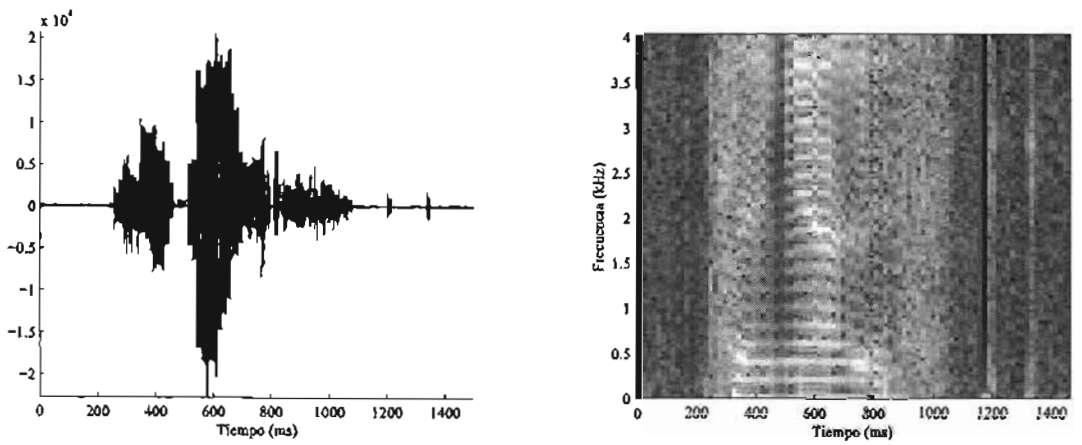


(c) Aplicación de deénfasis

Figura 4.8: *Reconstrucción de un segmento de voz para el vocoder LPC implementado*



(a) Señal original



(b) Señal sintetizada

Figura 4.9: *Resultado de la aplicación del vocoder LPC implementado*

Código 4.3: *Archivo de Matlab vocoderlpcs.m*

```
clear all
%-----------------------------------------------
%-----------------------------------------------
% ANALIZADOR LPC
%-----------------------------------------------
%-----------------------------------------------
archivo = input('Nombre del archivo de audio wav:','s');
fid = fopen(strcat(archivo,'.wav'),'r');
header = 44;
Y_temp = fread(fid,inf,'int16');
Y = Y_temp(header+1:end);
fs = Y_temp(13);
offset = mean(Y);
Y = Y-offset;

%-----------------------------------------------
% LONGITUD DE TRAMA = 30 ms
% PERIODO DE TRAMA = 15 ms
%-----------------------------------------------
long_trama = round(fs.*0.03);
periodo_trama = round(fs*0.015);
n_tramas = length(Y)/(long_trama-periodo_trama);
completar_long = ceil(n_tramas)*(long_trama-periodo_trama)+periodo_trama;
Y_comp = [Y; zeros(completar_long-length(Y),1)];

%-----------------------------------------------
% FILTRO PASO BAJAS
%-----------------------------------------------
[B,A] = butter(2,0.99);
Zi = zeros(1,2);

%-----------------------------------------------
% FILTRO PREENFASIS
%-----------------------------------------------
a = 0.9375;
Zip = 0;

%-----------------------------------------------
% DIVISION POR TRAMAS
% APLICAMOS FILTRADO PASO BAJAS Y PREENFASIS
%-----------------------------------------------
for i=1:round(n_tramas)
    inicio = (i-1)*(long_trama-periodo_trama)+1;
    fin = inicio+long_trama-1;
    [trama Zi] = filter(B,A,Y_comp(inicio:fin),Zi);
    tramas(i,:) = trama';
    [trama Zip] = filter([1 -a],1,trama,Zip);
    Y2(i,:) = trama';
end

tamano=size(Y2);

%-----------------------------------------------
% VENTANA DE HAMMING
%-----------------------------------------------
hamm = hamming(long_trama);
Y_vent = Y2.*(repmat(hamm',tamano(1),1));

%-----------------------------------------------
% ORDEN DEL ANALISIS LPC
%-----------------------------------------------
p = input('Orden:');
```

```
corr = [];
lpcs = [];
%-----------------------------------------------
% AUTOCORRELACION
%-----------------------------------------------
for g=1:tamano(1)
    corr_temp = xcorr(Y_vent(g,:),p);
    corr(g,:) = corr_temp(round(length(corr_temp)/2):end);
end

%-----------------------------------------------
% ANALISIS LPC
% OBTENCION DE LOS COEFICIENTES LPC MEDIANTE
% EL ALGORITMO LEVINSON-DURBIN
%-----------------------------------------------
for g=1:tamano(1)
    lpc(g,:) = levinson(corr(g,:),p);
    %-----------------------------------------
    % COEFICIENTES PARCOR O DE REFLEXION
    %-----------------------------------------
    ks(g,:) = poly2rc(lpc(g,:))';
end

%-----------------------------------------------
% VERIFICAMOS ESTABILIDAD DE LOS COEFICIENTES
% DE REFLEXION
%-----------------------------------------------
no_estable = ks>1;
estable = not(no_estable);
lpc(:,2:end) = lpc(:,2:end).*estable+no_estable;
lpcs = lpc;

%-----------------------------------------------
% GANANCIA DE LA TRAMA DE ANALISIS
%-----------------------------------------------
G = sqrt(corr(:,1)-sum(corr(:,2:end).*(-lpcs(:,2:end)),2))';
Zir = zeros(1,p);
for i=1:tamano(1)
    [residuo(i,:) Zir] = filter(lpcs(i,:),G(i),Y2(i,:),Zir);
    G(i) = max(abs(residuo(i,:))).*G(i);
end

%-----------------------------------------------
% DETECCION DEL TONO MEDIANTE AUTOCORRELACION
%-----------------------------------------------
[B,A] = butter(10,2*900/fs);
Zi2 = zeros(1,length(A)-1);

for i=1:tamano(1)
    [Y_filt(i,:) Zi2] = filter(B,A,residuo(i,:),Zi2);
end

Y_vent = Y_filt.*(hamm*ones(1,tamano(1)))';
tamano = size(Y_vent);

tercio = round(tamano(2)/3);
As = [max(Y_vent(:,1:tercio),[],2) max(Y_vent(:,2*tercio:end),[],2)];
Cl = 0.7*min(As,[],2);
Clmat = Cl*ones(1,size(Y_vent,2));

mayorCl = Y_vent>=Clmat;
menorCl = -1.*(Y_vent<=(-Clmat));
```

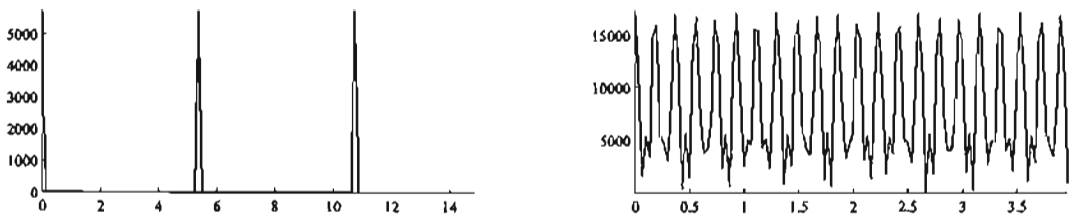
```
Y_clip = mayorCl+menorCl;
Y_clip2 = reshape(Y_clip',1,prod(size(Y_vent)));
Y3 = Y_clip;

corr_clip = [];
for g=1:tamano(1)
    corr_clip(g,:)= xcorr(Y3(g,:));
end

tc = size(corr_clip);
corr_clip2 = corr_clip(:,((tc(2)+1)/2):end);

%-----------------------------------------------
% Encontramos el maximo pico de la autocorrelacion
%-----------------------------------------------
tam_corr = size(corr_clip2);
maximo = corr_clip2(:,1);

pico = zeros(1,tam_corr(1));
ind_max = ones(1,tam_corr(1));

for j=1:tam_corr(1)
    for i=20:147
        if corr_clip2(j,i)>corr_clip2(j,i+1) & corr_clip2(j,i)>corr_clip2(j,i-1)...
            & corr_clip2(j,i)>pico(j) & corr_clip2(j,i)>0.3.*(maximo(j))
            pico(j)=corr_clip2(j,i);
            ind_max(j)=i;
        end
    end
end

pitch = fs./ind_max;
sonoros = (pitch<fs);

%-----------------------------------------------
% Se envian al sintetizador los parametros:
% lpcs - Coeficientes LPC
% G - Ganancia de la Trama
% sonoros - Sonoro:1 / Sordo:0
% pitch - Tono de las tramas sonoras
%-----------------------------------------------

%-----------------------------------------------
%-----------------------------------------------
% SINTETIZADOR LPC
%-----------------------------------------------
%-----------------------------------------------

%-----------------------------------------------
% PREDICTOR LINEAL
%-----------------------------------------------
Az = lpcs;
sordos = not(sonoros);

%-----------------------------------------------
% FUENTE DE RUIDO ALEATORIO
%-----------------------------------------------
uno = wgn(tamano(1),long_trama-periodo_trama,0);
uno = uno./max(max(abs(uno)));

%-----------------------------------------------
% GENERADOR DE IMPULSOS
```

```
%-----------------------------------------------------
dos_a = cos(2*pi.*pitch'*(0:(tamano(2)-periodo_trama-1))./fs);
dos = dos_a>=1;

%-----------------------------------------------------
% APLICAMOS LA GANANCIA CORRESPONDIENTE A LAS
% TRAMAS
%-----------------------------------------------------
sonoros2 = (G.*sonoros)'*ones(1,long_trama-periodo_trama);
sordos2 = (G.*sordos)'*ones(1,long_trama-periodo_trama);

%-----------------------------------------------------
% EXCITACION TOTAL
%-----------------------------------------------------
uno2 = uno.*sordos2;
dos2 = dos.*sonoros2;
excitacion = uno2+dos2;

CondI = zeros(1,size(Az,2)-1);
CondIp = 0;

%-----------------------------------------------------
% GENERAMOS LA SALIDA Y APLICAMOS DEENFASIS
%-----------------------------------------------------
b = -0.8;

for i=1:tamano(1)
[salida1 CondI] = filter(1,Az(i,:),excitacion(i,:),CondI);
[salida CondIp] = filter(1,[1 b],salida1,CondIp);
X_sint(i,:) = salida1;
X_de(i,:) = salida;
end

X = reshape(X_de',1,prod(size(X_de)));
X2 = reshape(X_sint',1,prod(size(X_sint)));

[B,A] = butter(2,0.99);
Pi = specgram(filter(B,A,Y));
Po = specgram(X);

sd_temp = SD(abs(Pi(:,1:min(size(Pi,2),size(Po,2)))'),abs(Po(:,1:min(size(Pi,2),size(Po,2)))'),size(Pi,1));
sd = mean(sd_temp)
```

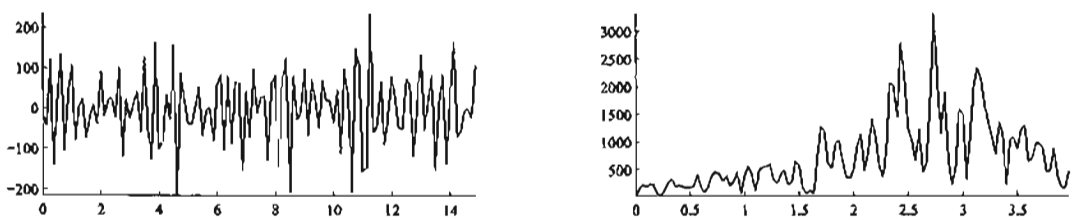
4.2.1. Coeficientes de Reflexión

Empleando el mismo codificador que en la sección anterior, pero esta vez usando los coeficientes de reflexión, obtenemos las siguientes gráficas para la misma trama de voz que la presentada en las gráficas 4.8.

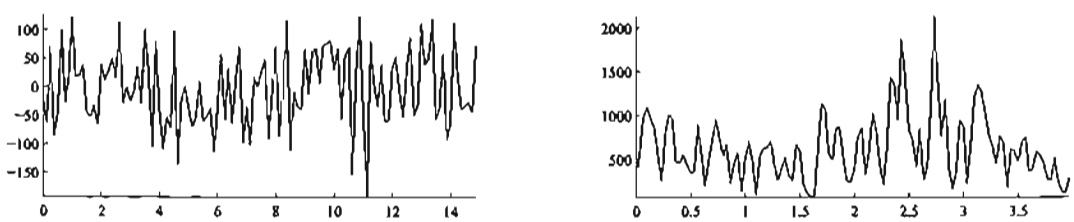
Aunque podría pensarse que los coeficientes LPC y los de reflexión operan de manera distinta, en realidad obtenemos resultados bastante similares. La diferencia en el uso de diferentes tipos de coeficientes radica, como se ha mencionado con anterioridad, en las propiedades que facilitan su cuantización.



(a) Excitación $Gu(n)$

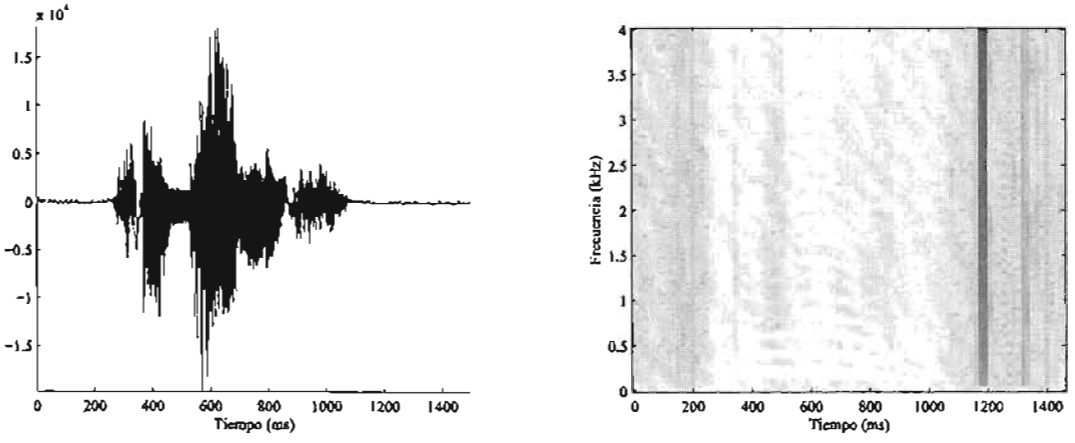


(b) Segmento de voz sintetizado

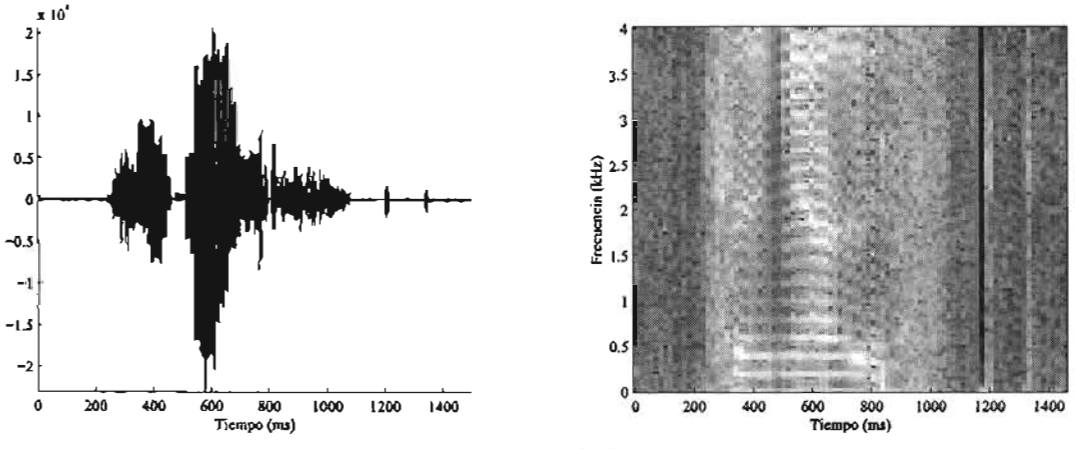


(c) Aplicación de deénfasis

Figura 4.10: *Reconstrucción de un segmento de voz para el vocoder LPC implementado usando los coeficientes de reflexión*



(a) Señal original



(b) Señal sintetizada

Figura 4.11: *Resultado de la aplicación del vocoder LPC implementado usando los coeficientes de reflexión*

4.3. CELP

Después de analizar las diversas técnicas de codificación y la aplicación en particular de la compresión CELP en los estándares para telefonía celular, se prosiguió finalmente a la simulación en Matlab de un codificador de voz basado en CELP.

A continuación se muestran los motivos a partir de los cuales se determinó el diseño de los codificadores CELP que se presentarán más adelante. Estos motivos se basan principalmente en la información presentada en las tablas 3.2 y 3.3, que proporcionan información acerca de las tasas de bits de los diferentes estándares.

Motivo	Decisión
La mayor porción de la tasa de bits es utilizada para codificar la excitación estocástica. La codificación de los parámetros espectrales y del tono utilizan tasas de bits similares de un estándar a otro, pero no así la tasa de bits utilizada para la codificación de la excitación correspondiente al libro de códigos fijo.	Enfocar el diseño en la generación de la excitación estocástica.
ACELP utiliza una mayor cantidad de bits para este propósito, ya que debe codificar las posiciones y amplitudes de una cantidad fija de pulsos. VSELP usa una menor cantidad debido a que únicamente requiere codificar el índice del vector de excitación, de la misma forma que el estándar FS1016.	Utilizar un libro de códigos fijo predefinido en lugar de asignar una cantidad n de pulsos a cada vector de excitación estocástica.
El algoritmo FS1016 emplea una tasa de bits de 4.8 kbps para codificar la voz, mientras que los estándares ACELP utilizan de 7.4 a 12.2 kbps y VSELP 7.95 kbps.	Basar el diseño en el algoritmo CELP FS1016.

Gracias la baja tasa de bits que maneja el algoritmo FS1016 es posible experimentar diversas modificaciones al mismo sin que la tasa de bits se incremente demasiado. Además, muchos de los estándares de codificación de voz actuales se han basado en el CELP original, manteniendo los principios básicos y modificando algunos cuantos aspectos, por lo que parece ser apropiado seguir esta metodología.

Por lo anterior, el primer paso fue la simulación del algoritmo FS1016 en Matlab. Se conservaron los principios básicos del algoritmo como se presentaron en la sección 3.2.1, aunque simplificando algunas de las etapas, como la búsqueda del retardo óptimo y las cuantizaciones de los parámetros. Una vez programado este algoritmo, como se determinó enfocar el diseño en el proceso de búsqueda del libro de códigos fijo, el segundo paso fue precisamente hacer modificaciones tanto al libro de códigos como al procedimiento de búsqueda de la excitación estocástica.

Durante esta etapa se obtuvieron varios codificadores de mala calidad. Por ejemplo, uno de los intentos fallidos consistía en modificar el vector que genera el libro de códigos estocástico en FS1016 de forma que en lugar de tener únicamente amplitudes +1 y -1, se tuvieran dos niveles más, +0.5 y -0.5, incluyendo un par de pulsos por cada pulso unitario, con lo que se redujo el número de ceros del vector; otro intento consistía también en modificar el vector del libro de códigos esta vez sin aumentar el número de pulsos, sino modificando las amplitudes de los pulsos existentes. Sin embargo, estas modificaciones no lograron mejorar la calidad del codificador CELP simulado.

Como fue imposible obtener un codificador de mejor calidad sin aumentar la tasa de bits, esto es, modificando únicamente el libro de códigos fijo, se recurrió a otras opciones que inevitablemente aumentaron la cantidad de bits.

Una de las opciones fue realizar dos búsquedas en el libro de códigos fijo, como VSELP, pero en lugar de emplear vectores base para generar el conjunto de vectores de excitación se utilizó el mismo libro de códigos estocástico

Parámetros	Nombre	Valor
Orden del predictor LPC	p	10
Tipo de ventana		Hamming
Longitud de la trama	N	30 ms
Longitud de la subtrama	I	7.5 ms
Tamaño del libro de códigos adaptable		128×60
Rango del retardo		20-147 (sin fracciones)
Tamaño del libro de códigos fijo		512×60
Filtro de peso perceptual	μ	0.8
Expansión del ancho de banda	ΔB	15 Hz
Coeficiente de expansión de ΔB	δ	0.994
Postfiltro	μ_1'	0.8
	μ_2'	0.5

Cuadro 4.7: *Parámetros de los codificadores CELP implementados*

empleado en el codificador FS1016, por lo que se duplicó la tasa de bits correspondiente a la excitación estocástica, resultando en una tasa de bits de

$$1133{,}3 + 1600 + 2(1866{,}67) + 200 = 6{,}7kbps \tag{4.5}$$

La segunda modificación propuesta consistió en subdividir a la mitad las subtramas de 7.5 ms para encontrar la mejor excitación estocástica, esto es, se realizó la búsqueda del libro de códigos estocástico para segmentos de 3.75 ms. Por lo tanto la tasa de bits se incrementa de la misma forma, a 6.7 kbps, que en el codificador anterior propuesto.

En la tabla 4.7 se encuentran los parámetros de los tres codificadores mencionados.

En las figuras 4.13-4.15 se muestran algunos resultados parciales del análisis realizado por los codificadores CELP sobre una trama de la señal utilizada (figura 4.12). Las líneas verticales indican la división de la trama en subtramas.

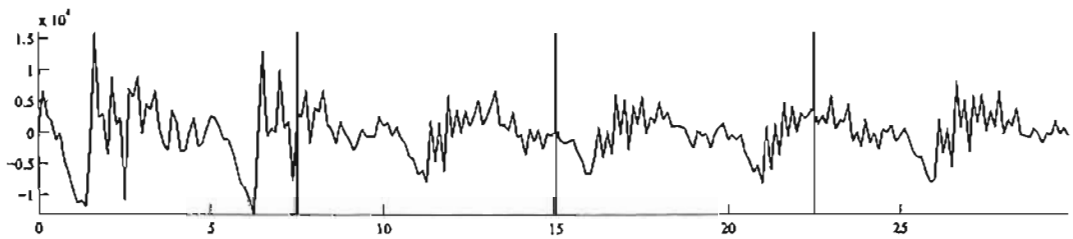


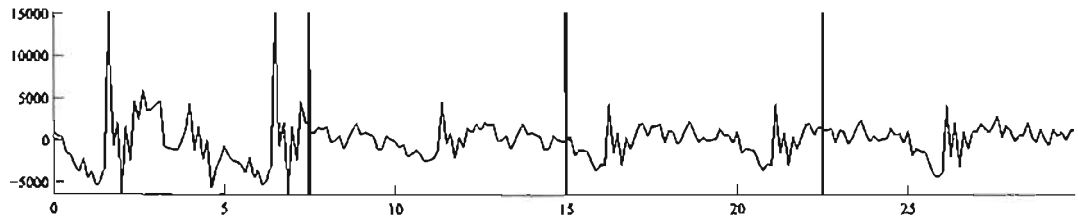
Figura 4.12: *Trama de voz original*

Posteriormente, en las figuras 4.16-?? se muestra la síntesis de la trama de voz realizada por los codificadores CELP a partir de los parámetros extraídos en el análisis de la trama de la señal utilizada (figura 4.12).

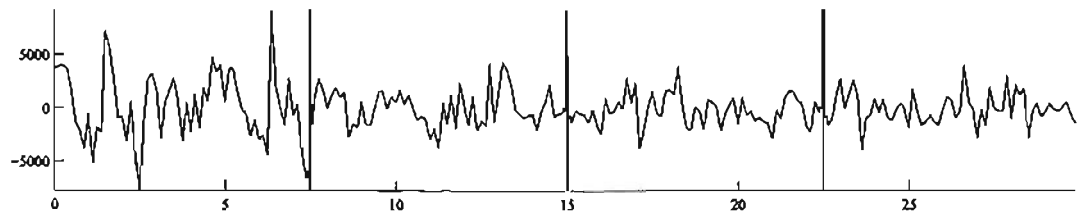
Finalmente, en las figuras ?? se observan las gráficas en el tiempo así como los espectros tanto de la señal de entrada como las salidas de los codificadores implementados.

4.3.1. Tasa de bits y SNR

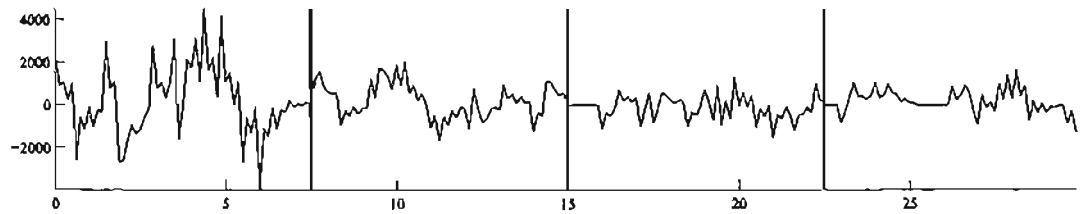
Como se ha visto, para realizar un estudio de los codificadores simulados fueron definidas dos métricas de comparación de los algoritmos, éstas son:



(a) Vector de código seleccionado del libro de códigos adaptable escalado por su correspondiente ganancia



(b) Residuo

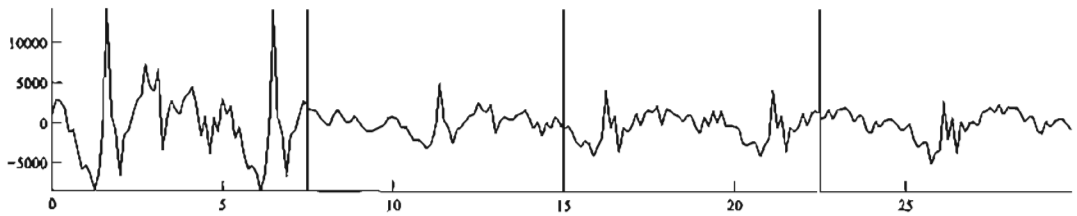


(c) Vector de código seleccionado del libro de códigos estocástico escalado por su correspondiente ganancia

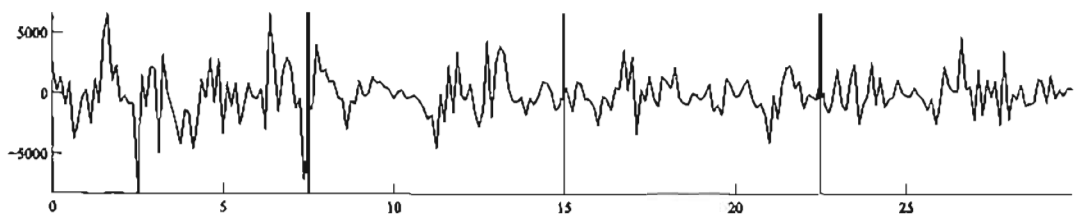
Figura 4.13: *Procedimiento de búsqueda de la excitación en CELP FS1016*

Algoritmo	Tasa de bits (kbps)	SNR (dB)	SEGSNR (dB)	Compresión (%)
CELP FS1016	4.8	3.8	3.3	96.3
VSELP modificado	6.7	5.1	4.3	94.8
CELP FS1016 modificado	6.7	5.3	4.6	94.8

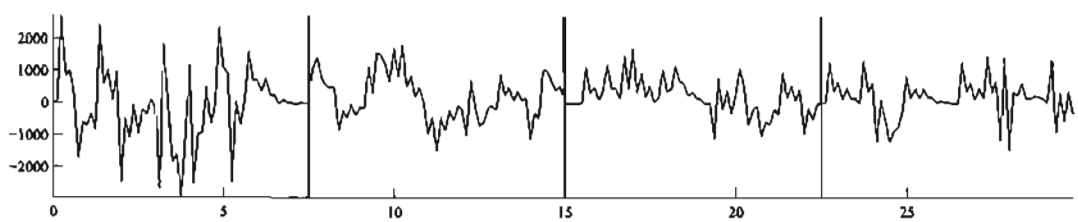
Cuadro 4.8: *Tasa de bits y SNR de los codificadores CELP implementados*



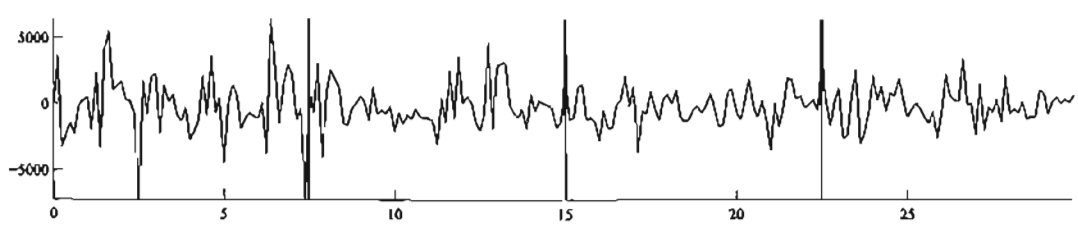
(a) Vector de código seleccionado del libro de códigos adaptable escalado por su correspondiente ganancia



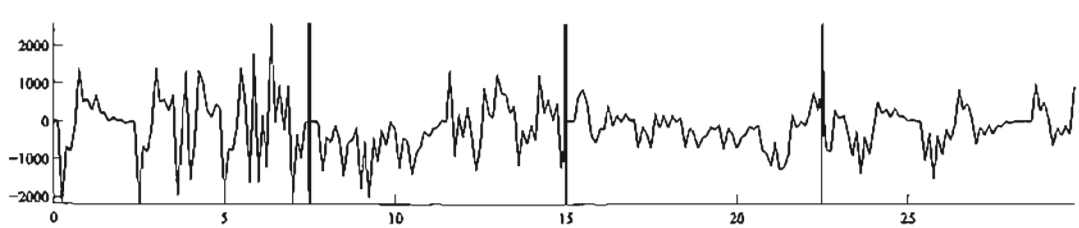
(b) Residuo



(c) Vector de código seleccionado del libro de códigos estocástico escalado por su correspondiente ganancia

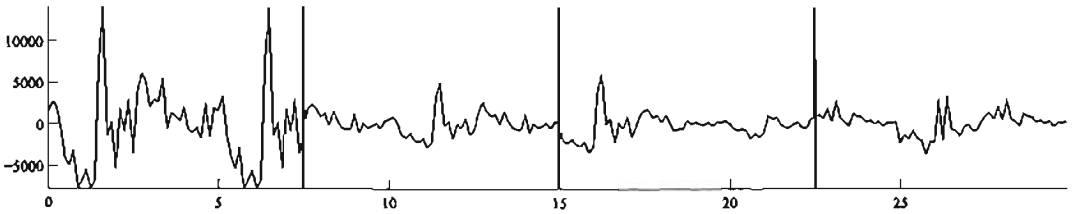


(d) Residuo

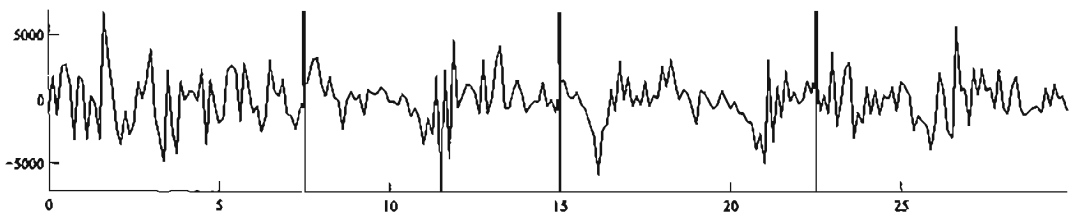


(e) Vector de código seleccionado del libro de códigos estocástico escalado por su correspondiente ganancia

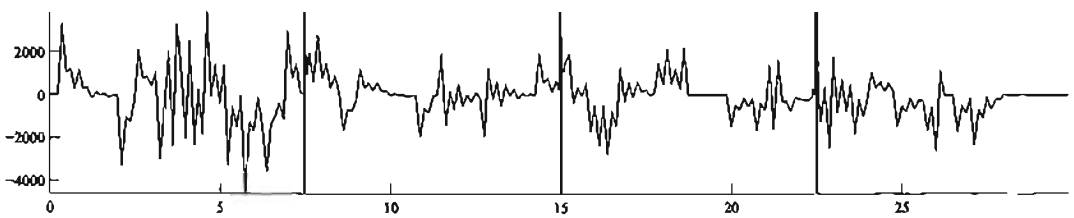
Figura 4.14: *Procedimiento de búsqueda de la excitación en VSELP modificado*



(a) Vector de código seleccionado del libro de códigos adaptable escalado por su correspondiente ganancia

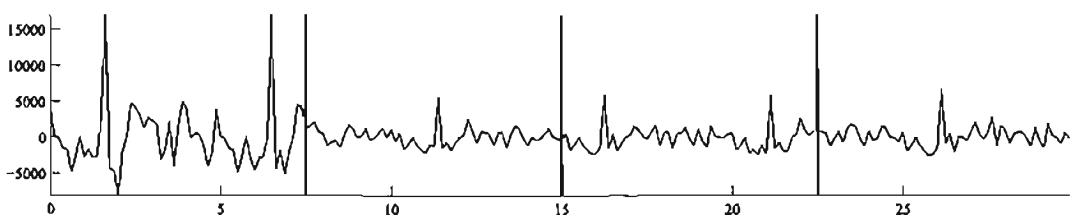


(b) Residuo

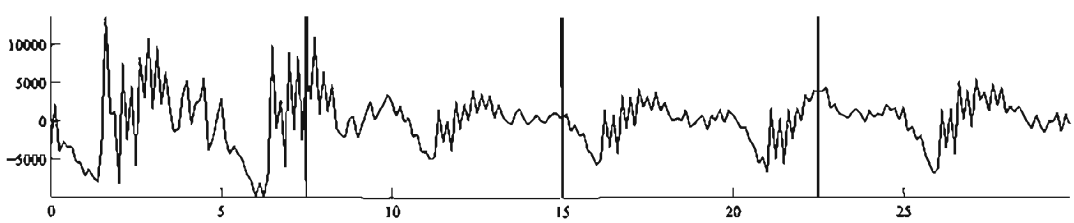


(c) Vector de código seleccionado del libro de códigos estocástico escalado por su correspondiente ganancia

Figura 4.15: *Procedimiento de búsqueda de la excitación en CELP FS1016 modificado*

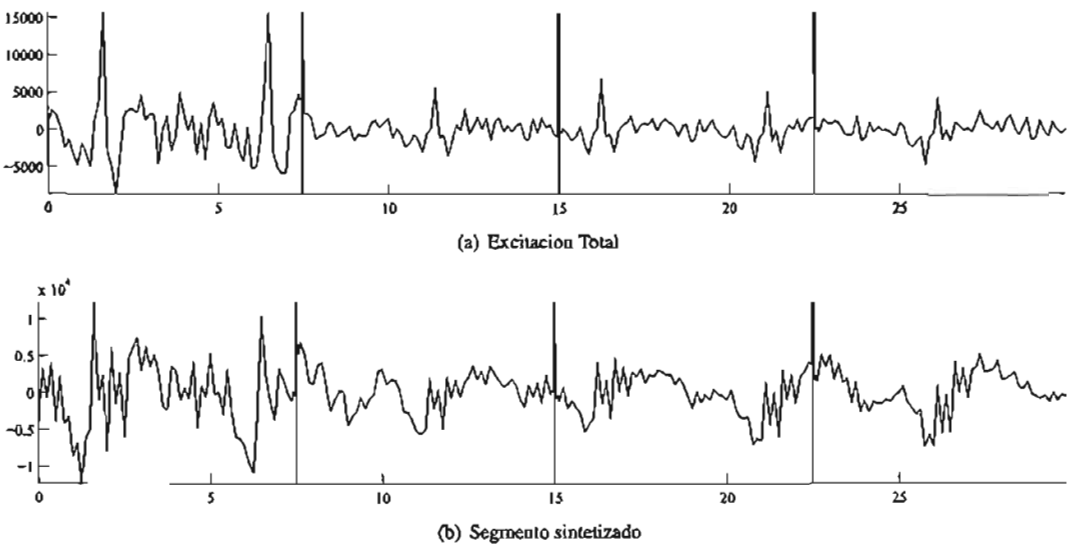


(a) Excitacion Total



(b) Segmento sintetizado

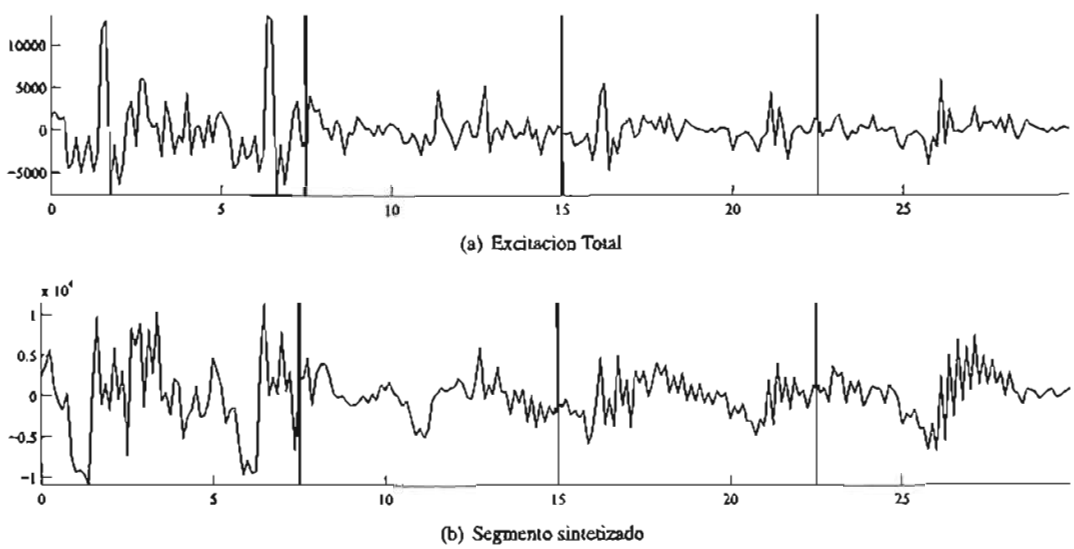
Figura 4.16: *Síntesis del segmento de voz de la figura 4.13*



(a) Excitacion Total

(b) Segmento sintetizado

Figura 4.17: *Síntesis del segmento de voz de la figura 4.14*



(a) Excitacion Total

(b) Segmento sintetizado

Figura 4.18: *Síntesis del segmento de voz de la figura 4.15*

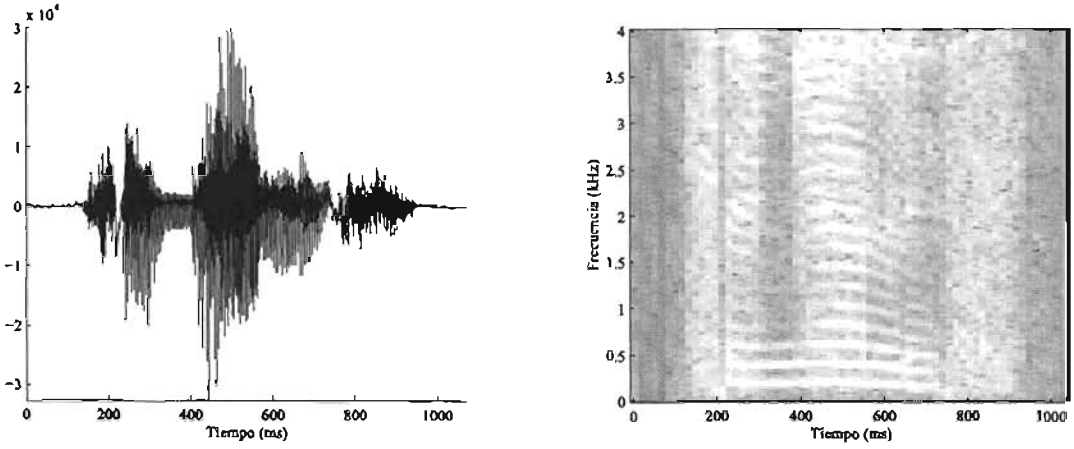


Figura 4.19: *Señal original*

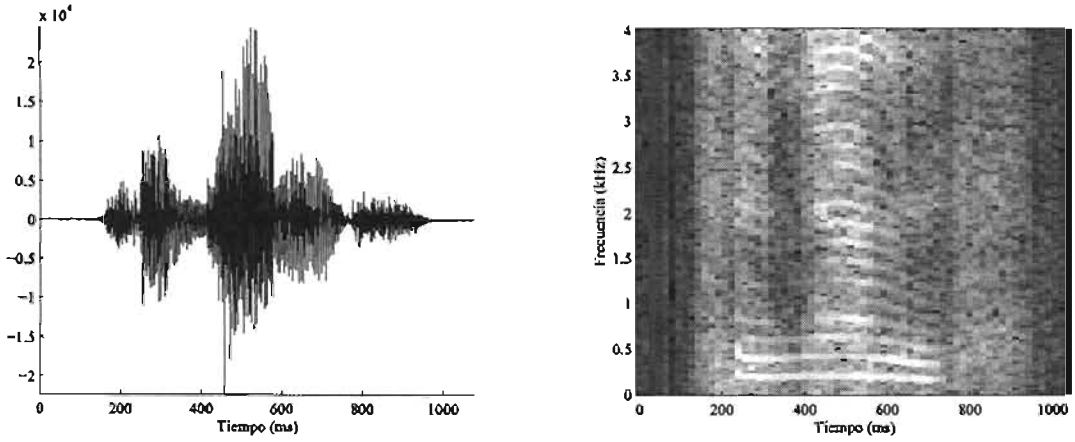


Figura 4.20: *Señal sintetizada. Resultado de la aplicación del vocoder CELP FS1016 implementado*

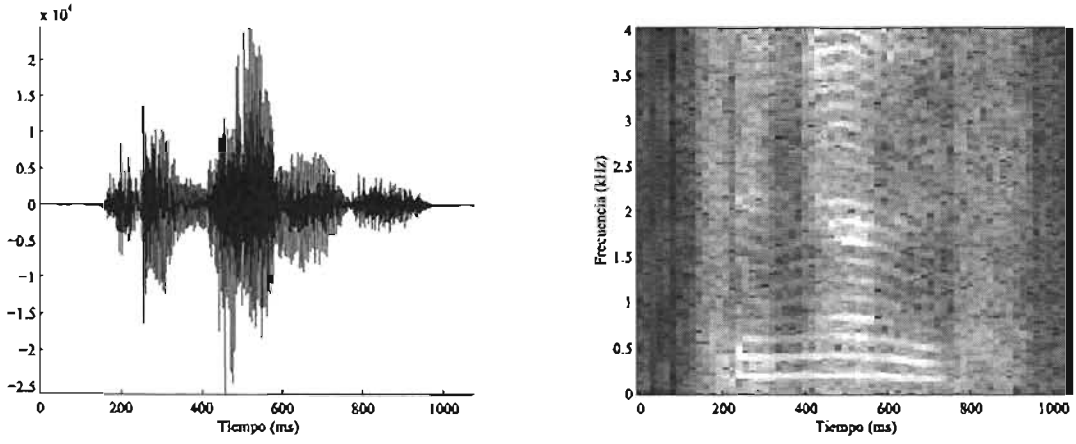


Figura 4.21: *Señal sintetizada. Resultado de la aplicación del vocoder VSELP implementado*

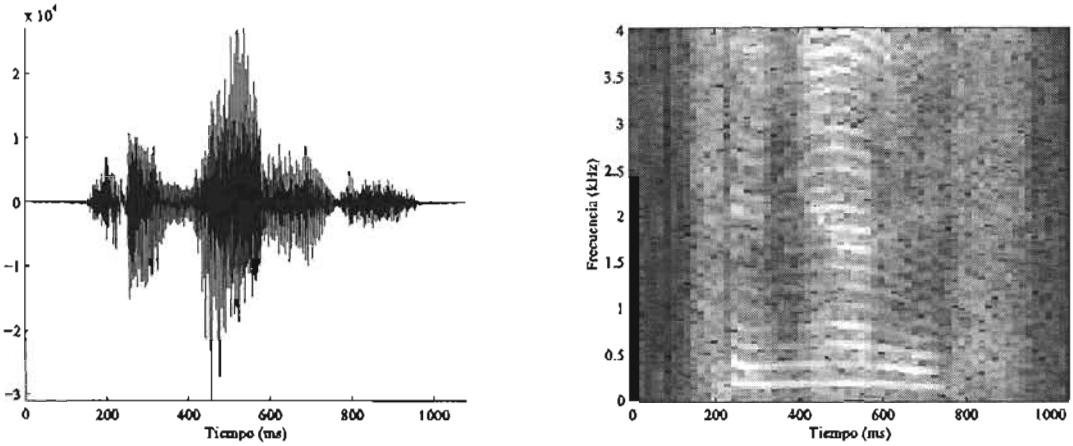


Figura 4.22: *Señal sintetizada. Resultado de la aplicación del vocoder CELP FS1016 modificado implementado*

- Capacidad de medio de transmisión necesaria para cada algoritmo.
- Calidad de reconstrucción de la señal.

De acuerdo con los SNR obtenidos (tabla 4.8) podríamos comparar las versiones modificadas de VSELP y de CELP FS1016 con un ADPCM, como el presentado en la sección 4.1.2, de 3 bits, lo que corresponde a una tasa de bits de 24 kbps, la cual es mayor al triple de la tasa manejada por los codificadores CELP. Además, duplicando la tasa de bits manejada por el vocoder LPC, obtenemos una calidad mucho mejor en los codificadores CELP. En la figura ?? se muestran los SNR correspondientes a cada trama de voz obtenidos por la aplicación de los tres codificadores CELP.

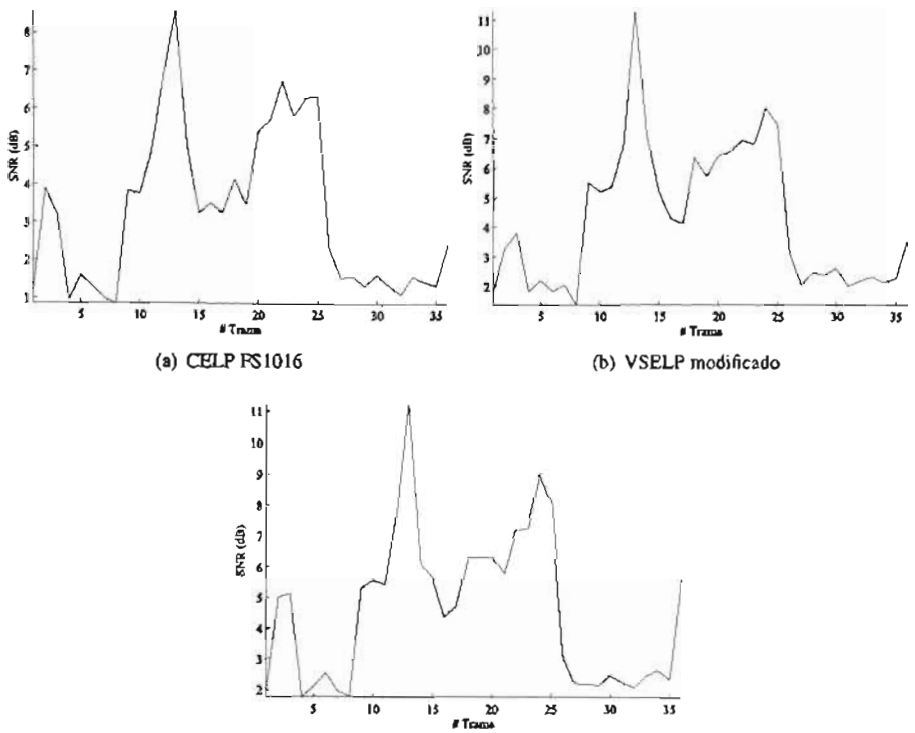
4.3.2. Pruebas de escucha

Como se ha mencionado anteriormente las pruebas subjetivas son de extrema importancia en la evaluación de un codificador de voz a bajas tasas de bits. Por lo tanto se requirió a un grupo de personas que evaluaran, con base en la tabla 4.9, una serie de grabaciones resultado de la aplicación de las codificaciones CELP descritas con anterioridad.

Nota	Calidad	Esfuerzo de escucha	Degradación
5	Excelente	Posible relajación completa, no requiere ningún esfuerzo	Inaudible
4	Buena	Atención necesaria, no requiere esfuerzo apreciable	Audible pero no molesta
3	Aceptable	Se necesita esfuerzo moderado	Ligeramente molesta
2	Mediocre	Se necesita esfuerzo considerable	Molesta
1	Mala	Cualquier esfuerzo no permite comprender	Muy Molesta

Cuadro 4.9: *Descripción de las notas MOS para las pruebas de escucha de los codificadores CELP implementados*

Debido a que las condiciones tanto de grabación como de escucha no son las ideales, fue necesario realizar la evaluación sobre los tres codificadores al mismo tiempo, de forma que los participantes de las pruebas pudieran evaluar los resultados de los tres codificadores de manera comparativa.



(a) CELP FS1016

(b) VSELP modificado

(c) CELP FS1016 modificado

Figura 4.23: *SNR para cada trama de voz*

Para el estudio se utilizaron 14 frases, 11 en español y 3 inglés, 5 de ellas de voz femenina y 9 masculina. Además algunas de ellas presentaban altos niveles de ruido debido a la grabación. Así que podría decirse que se evaluaron los codificadores para diversas condiciones de ruido y locutores.

El test fue realizado por 15 personas, en la tabla 4.11 se muestran todas las notas proporcionadas por estas personas y en la tabla 4.10 los promedios de las mismas.

Frase	CELP FS1016	CELP FS1016 modificado	VSELP modificado
buzz	4.0	4.4	4.1
caramba	3.6	3.8	3.6
cine	3.1	3.8	3.3
force	4.4	4.3	3.9
hasta	4.0	4.0	3.8
ingenieria	3.7	4.2	4.0
offer	3.7	4.1	3.9
risa	4.4	4.4	4.4
ronquido	4.5	4.6	4.2
señales	3.5	3.7	3.5
tarzan	3.6	4.0	4.2
telcel	3.6	3.8	3.3
universidad	4.2	4.2	4.3
wav	2.9	3.1	2.6
MOS	3.8	4.0	3.8

Cuadro 4.10: *Promedio MOS para los codificadores AbS implementados*

Aunque el codificador VSELP modificado y FS1016 modificado emplean la misma tasa de bits, tanto la medida objetiva como la subjetiva de la calidad fueron mejores en el segundo caso. Particularmente el aumento de 0.2 en el MOS con respecto a los otros dos codificadores indica un aumento considerable en la calidad de la voz reconstruida.

Frase	Codificador																	
buzz	FS1016	4.3	4.5	4.0	3.5	4.0	4.0	4.6	4.5	4.8	4.0	2.0	2.0	5.0	4.0	3.0	4.0	5.0
"To infinity and beyond"	FS1016 modificado	4.7	4.5	4.5	4.0	5.0	4.5	4.6	4.5	4.8	4.0	4.0	4.0	5.0	3.0	3.0	5.0	5.0
	VSELP modificado	3.8	4.0	4.0	3.0	4.5	4.0	4.6	4.5	4.8	4.0	5.0	3.0	5.0	4.0	2.0	4.0	5.0
caramba	FS1016	4.3	4.2	4.0	2.5	4.0	4.7	4.5	4.0	3.7	4.0	4.0	2.0	4.0	3.0	1.0	3.0	4.5
"Ay Caramba"	FS1016 modificado	4.3	4.2	4.0	2.5	5.0	4.5	4.5	4.0	3.7	3.0	2.0	3.0	4.0	4.0	3.0	4.0	4.5
	VSELP modificado	3.8	3.8	4.0	3.0	5.0	4.5	4.5	4.0	3.7	2.0	3.0	3.0	3.0	4.0	2.0	4.0	4.5
cine	FS1016	4.5	4.0	4.0	2.5	4.0	4.0	4.0	3.0	3.6	2.0	1.0	3.0	3.0	2.0	1.0	3.0	4.5
"Vamos al cine"	FS1016 modificado	4.3	4.0	4.0	3.5	5.0	4.6	4.0	3.0	3.6	4.0	4.0	4.0	3.0	3.0	3.0	3.0	4.5
	VSELP modificado	4.0	3.8	4.0	2.5	4.0	4.4	4.0	2.5	3.6	3.0	3.0	3.0	2.0	3.0	2.0	2.0	4.5
force	FS1016	4.8	5.0	4.0	5.0	4.5	4.5	5.0	2.5	4.5	4.0	4.0	3.0	5.0	4.0	5.0	5.0	5.0
"The force will be with	FS1016 modificado	4.2	4.2	4.0	3.5	5.0	4.8	5.0	3.5	4.5	4.0	4.0	5.0	5.0	3.0	4.0	5.0	5.0
you, young Skywalker"	VSELP modificado	3.8	4.0	3.5	4.0	4.0	4.7	4.0	2.5	4.5	3.0	3.0	4.0	5.0	3.0	4.0	4.0	5.0
hasta	FS1016	4.4	4.5	4.0	4.0	4.5	4.4	4.5	3.5	4.2	4.0	4.0	3.0	4.0	4.0	3.0	4.0	4.5
"Hasta la vista, baby"	FS1016 modificado	4.8	4.5	4.0	4.5	5.0	4.7	4.5	3.5	4.2	3.0	2.0	4.0	4.0	2.0	4.0	4.0	4.5
	VSELP modificado	4.5	4.5	4.5	4.0	4.0	4.8	5.0	3.5	4.3	2.0	1.0	2.0	5.0	4.0	3.0	3.0	5.0
ingeniería	FS1016	4.7	4.8	4.5	3.5	5.0	4.6	5.0	4.8	4.7	3.0	1.0	2.0	5.0	2.0	1.0	3.0	4.5
"Ingeniería"	FS1016 modificado	4.5	4.8	4.5	3.5	4.5	4.9	5.0	4.8	4.7	5.0	3.0	3.0	5.0	2.0	3.0	4.0	4.5
	VSELP modificado	4.6	4.8	4.5	3.5	4.5	5.0	5.0	4.8	4.7	5.0	2.0	2.0	5.0	2.0	3.0	3.0	4.5
offer	FS1016	4.2	4.7	4.5	4.5	4.0	4.5	4.9	3.9	4.0	3.0	3.0	3.0	3.0	3.0	1.0	4.0	4.5
"I will make him an offer	FS1016 modificado	4.6	4.6	4.0	4.5	4.5	4.9	3.8	3.9	4.0	3.0	5.0	3.0	4.0	4.0	5.0	3.0	4.5
he can't refuse"	VSELP modificado	4.0	4.3	4.5	5.0	5.0	4.7	4.9	3.9	4.0	3.0	4.0	3.0	4.0	3.0	2.0	3.0	4.5
risa	FS1016	3.8	5.0	5.0	4.0	5.0	4.6	4.7	4.1	4.3	5.0	5.0	3.0	4.0	4.0	4.0	4.0	5.0
"Jaajajaaa"	FS1016 modificado	4.0	4.8	5.0	4.5	5.0	5.0	4.7	4.1	4.3	5.0	3.0	3.0	5.0	5.0	4.0	4.0	5.0
	VSELP modificado	3.8	4.6	5.0	4.0	4.5	5.0	4.7	4.1	4.3	5.0	4.0	2.0	5.0	3.0	5.0	5.0	5.0
ronquido	FS1016	4.4	5.0	4.5	4.5	5.0	4.5	4.8	4.0	4.1	5.0	5.0	3.0	5.0	3.0	5.0	4.0	5.0
	FS1016 modificado	4.8	5.0	4.5	4.5	5.0	4.8	4.8	4.0	4.1	5.0	3.0	4.0	5.0	4.0	5.0	5.0	5.0
	VSELP modificado	4.3	4.7	4.5	3.5	4.5	4.0	4.8	4.0	4.1	3.0	5.0	3.0	4.0	3.0	5.0	5.0	4.5
señales	FS1016	4.5	4.5	4.0	2.5	4.5	3.7	4.7	3.8	3.8	3.0	3.0	2.0	3.0	3.0	2.0	3.0	5.0
"Señales"	FS1016 modificado	4.2	4.5	4.0	2.5	5.0	3.8	4.7	3.8	3.8	3.0	2.0	3.0	4.0	3.0	4.0	2.0	5.0
	VSELP modificado	4.2	4.5	4.0	2.0	4.5	4.6	4.7	3.8	3.8	2.0	1.0	3.0	3.0	3.0	4.0	2.0	5.0
tarzan	FS1016	4.3	4.8	4.0	2.5	3.5	2.5	4.6	4.0	4.5	5.0	1.0	3.0	4.0	3.0	3.0	3.0	4.5
"Ahaaaaha"	FS1016 modificado	4.5	4.8	4.0	3.0	4.5	3.3	4.6	4.0	4.5	5.0	3.0	4.0	4.0	3.0	4.0	3.0	4.5
	VSELP modificado	4.3	4.5	4.0	3.5	4.0	3.5	5.0	4.3	4.7	3.0	5.0	4.0	3.0	5.0	5.0	3.0	5.0
telcel	FS1016	4.0	4.3	4.0	3.5	4.0	4.0	4.8	4.0	4.4	3.0	4.0	4.0	2.0	3.0	2.0	2.0	5.0
"Telefonía celular"	FS1016 modificado	4.5	4.5	4.0	3.0	4.5	4.5	4.8	4.0	4.4	2.0	4.0	3.0	2.0	4.0	4.0	3.0	5.0
	VSELP modificado	3.8	4.2	3.5	3.0	4.5	4.7	4.8	4.0	4.4	3.0	2.0	2.0	1.0	2.0	3.0	2.0	4.5
universidad	FS1016	3.8	4.0	4.0	4.0	5.0	3.0	4.8	4.8	4.6	5.0	4.0	5.0	5.0	4.0	3.0	3.0	5.0
"Universidad"	FS1016 modificado	4.0	4.0	4.0	3.5	4.0	4.5	4.8	4.8	4.6	5.0	2.0	4.0	5.0	3.0	5.0	4.0	5.0
	VSELP modificado	4.2	4.5	4.0	4.0	4.5	3.5	4.5	4.5	4.6	4.0	3.0	4.0	5.0	5.0	4.0	4.0	5.0
wav	FS1016	4.0	4.3	3.0	2.0	4.0	2.0	3.0	2.5	3.2	4.0	1.0	4.0	2.0	2.0	1.0	4.0	4.0
"Los ficheros wav son el	FS1016 modificado	3.8	4.2	3.5	2.0	4.0	3.8	3.5	2.5	3.2	2.0	5.0	3.0	2.0	3.0	2.0	2.0	4.0
formato creado por MS"	VSELP modificado	3.5	3.8	4.0	1.5	4.0	2.7	3.5	2.5	2.0	2.0	2.0	2.0	1.0	2.0	1.0	2.0	4.0

Cuadro 4.11: *Resultados MOS para los codificadores AbS implemeniados*

Código 4.4: *Archivo de Matlab CELP.m*

```
%-------------------------------------------------------
%-------------------------------------------------------
% CELP FS1016 con diccionario adaptable y gaussiano
%-------------------------------------------------------
%-------------------------------------------------------
clear all;
archivo = input('Nombre del archivo .wav:','s');
fid = fopen(strcat(archivo,'.wav'),'r');
header = 44;
Y_temp = fread(fid,inf,'int16');
Y = Y_temp(header+1:end);
fs = Y_temp(13);
offset = mean(Y);
Y = Y-offset;

%-------------------------------------------------------
% TRAMA : 30 ms --> 240 muestras
% SUBTRAMAS : 7.5 ms --> 60 muestras
%-------------------------------------------------------
long_trama = 240;
long_subtrama = 60;
n_subtramas_por_trama = 4;

n_tramas = ceil(length(Y)/long_trama);
completar_long = n_tramas*long_trama;
Y_completa = [Y; zeros(completar_long-length(Y),1)];
tramas = reshape(Y_completa',long_trama,n_tramas)';

%-------------------------------------------------------
% VENTANA DE HAMMING
%-------------------------------------------------------
hamm = hamming(long_trama)';

%-------------------------------------------------------
% FILTRO PASO ALTAS
%-------------------------------------------------------
B = [0.946,-1.892,0.946];
A = [1.0,-1.889033,0.8948743];
Zi = zeros(1,2);

trama_anterior = zeros(1,long_trama);

for i=1:round(n_tramas)
    %-------------------------------------------------------
    %FILTRO PASO ALTAS A 100Hz
    %-------------------------------------------------------
    [tramas(i,:),Zi] = filter(B,A,tramas(i,:),Zi);
    %-------------------------------------------------------
    %TRAMA DE ANALISIS: MEDIA TRAMA ANTERIOR Y MEDIA ACTUAL
    %-------------------------------------------------------
    trama_analisis(i,1:round(long_trama/2)) = trama_anterior(1,round(long_trama/2)+1:end);
    trama_analisis(i,round(long_trama/2)+1:long_trama) = tramas(i,1:round(long_trama/2));
    trama_anterior = tramas(i,:);
    %-------------------------------------------------------
    %VENTANA DE HAMMING
    %-------------------------------------------------------
    tramas_ventana(i,:) = tramas(i,:).*hamm;
end

[n_tramas long] = size(tramas);
Y2 = tramas_ventana;
```

```
%-----------------------------------------------
%SUBDIVIDIMOS LA TRAMA DE ANALISIS EN SUBTRAMAS
%-----------------------------------------------
n_subtramas = n_subtramas_por_trama.*n_tramas;
subtramas = reshape(trama_analisis',long_subtrama,n_subtramas)';

global p e0 e02 h alpha idb z lpcs_subtrama
%-----------------------------------------------
% NO. COEFICIENTES LPC
%-----------------------------------------------
p = 10;
%-----------------------------------------------
%FILTRO DE PESO PERCEPTUAL W(z)
%-----------------------------------------------
alpha = 0.8;
%-----------------------------------------------
%COEFICIENTE DE EXPANSION DE LOS LPC (15 Hz)
%-----------------------------------------------
omega = 0.994127;
%-----------------------------------------------
% CODEBOOK ESTOCASTICO
%-----------------------------------------------
load stochcb.dat
vector_base = reshape(stochcb',1,prod(size(stochcb)));
point = 1024-2;

for j=1:512
    codebook(j,:) = vector_base(1,point+1:point+long_subtrama);
    point = point-2;
end

corr=[];
lpcs=[];
%-----------------------------------------------
% AUTOCORRELACION
%-----------------------------------------------
for g=1:n_tramas
    temp = xcorr(Y2(g,:),p);
    corr(g,:) = temp(round(length(temp)/2):end);
end

%-----------------------------------------------
% ANALISIS LPC
% OBTENCION DE LOS COEFICIENTES LPC MEDIANTE
% EL ALGORITMO LEVINSON-DURBIN
%-----------------------------------------------
for g=1:n_tramas
    lpcs1(g,:) = levinson(corr(g,:),p);
    %-----------------------------------------------
    %EXPANSION DE LOS LPC
    %-----------------------------------------------
    lpcs(g,:) = lpcs1(g,:).*(omega.^(0:p));
end

%-----------------------------------------------
% LPC'S PARA LAS SUBTRAMAS
% INTERPOLACION LINEAL DE LOS LPC
%-----------------------------------------------
n = 1:(n_tramas-1);

for j=1:n_subtramas_por_trama
    a = (j-1)./n_subtramas_por_trama;
    lpcs_subtrama(4.*n-4+j,2:p+1) = (1-a).*lpcs(n,2:end)+a.*lpcs(n+1,2:end);
```

```
    lpcs_subtrama(4.*n_tramas-4+j,2:p+1)= (1-a).*lpcs(n_tramas,2:end);
end

lpcs_subtrama(:,1) = 1;

%-----------------------------------------------------------
% CODEBOOK ADAPTABLE
%-----------------------------------------------------------
d1 = 20;
d2 = 147;
%-----------------------------------------------------------
% CONDICIONES INICIALES
%-----------------------------------------------------------
d22 = zeros(1,p);
d32 = zeros(1,p);
d23 = zeros(1,p);
d33 = zeros(1,p);
MAXPA = long_subtrama+d2+22;
d1a = zeros(1,MAXPA);
d1b = zeros(1,MAXPA);
idb = 209;
memoria_pitch = zeros(1,idb);
long_memoria_pitch = d2+long_subtrama*2+p+2;
buffer = zeros(1,long_memoria_pitch);

DA = zeros(length(d1:d2),long_subtrama);
inicio = long_memoria_pitch-long_subtrama+1;
v0low = long_memoria_pitch - idb - long_subtrama + 1;
v0high = v0low + idb - 1;

global indice_pitch beta_pitch
beta_pitch(1) = 0;
indice_pitch(1) = d1;

for z=1:n_subtramas
    e0 = zeros(1,long_subtrama);
    %-----------------------------------------------------------
    %ERROR INICIAL
    %-----------------------------------------------------------
    fctemp = lpcs_subtrama(z,:).*(alpha.^(0:p)) ;
    e0 = zeros(1,long_subtrama);
    e0 = filter(1,lpcs_subtrama(z,:),e0,d22);
    e0 = subtramas(z,:)-e0;
    e0 = filter(lpcs_subtrama(z,:),fctemp,e0,d32);
    %-----------------------------------------------------------
    %RESPUESTA AL IMPULSO
    %-----------------------------------------------------------
    h = filter(1,fctemp,[1 zeros(1,long_subtrama-1)]);
    %-----------------------------------------------------------
    %DICCIONARIO ADAPTABLE
    %-----------------------------------------------------------
    memoria_pitch = d1b;

    if z~=1
        buffer(v0low:v0high)= memoria_pitch(1:idb);
        for j=d1:d2
            lag = inicio-j;
            DA(j-d1+1,:) = buffer(lag:lag+long_subtrama-1);
        end
        for j=1:size(DA,1)
            DAW_temp(j,:) = conv(h(1:30),DA(j,:));
        end
```

```
        DAW = DAW_temp(:,1:long_subtrama);
        arg_temp = DAW;

    for j=d1:long_subtrama-1
        arg_temp(j-d1+1,j+1:long_subtrama) = DAW(j-d1+1,j+1:long_subtrama)+DAW(j-d1+1,1:long_subtrama-j);
        If j<fix(long_subtrama/2)
            imin = (2*j)+1;
            imax = long_subtrama;
            arg_temp(j-d1+1,imin:imax) = arg_temp(j-d1+1,imin:imax)+DAW(j-d1+1,1:long_subtrama-(2*j) );
        end
    end

    DAW = arg_temp;

    %-----------------------------------------------
    % GANANCIA DEL CODEBOOK ESTOCASTICO = 0
    % OBTENEMOS GANANCIA DEL CODEBOOK ADAPTABLE
    %-----------------------------------------------
    arg = ones(size(DAW,1),1)*e0.*DAW;
    %-----------------------------------------------
    % CORRELACION
    %-----------------------------------------------
    num = sum(arg,2);
    %-----------------------------------------------
    % ENERGIA
    %-----------------------------------------------
    den = sum(DAW.^2,2);
    If not(all(den))
        den=den+(den==0);
    end

    G1(:,z) = num./den;
    Talpha = G1(:,z).*num;
    [bp,ip] = max(Talpha);
    beta_pitch(z) = G1(ip,z);
    indice_pitch(z) = ip+d1-1;
    end

    e0 = zeros(1,long_subtrama);
    %-----------------------------------------------
    % ACTUALIZACION DE LA MEMORIA DEL FILTRO LTP
    %-----------------------------------------------
    [e0,d1a] = ltp(e0,long_subtrama,d1a,idb,[indice_pitch(z) beta_pitch(z)]);
    [e0 d22] = filter(1,lpcs_subtrama(z,:),e0,d22);
    e0 = subtramas(z,:)-e0;
    [e0 d32] = filter(lpcs_subtrama(z,:),fctemp,e0,d32);

    %-----------------------------------------------
    % CODEBOOK ESTOCASTICO
    %-----------------------------------------------
    for j=1:size(codebook,1)
        DGW_temp(j,:) = conv(h(1:30),codebook(j,:));
    end
    DGW = DGW_temp(:,1:long_subtrama);

    %-----------------------------------------------
    % OBTENEMOS LA GANANCIA DEL DICCIONARIO ESTOCASTICO
    %-----------------------------------------------
    %-----------------------------------------------
    % CORRELACION
    %-----------------------------------------------
    arg1 = sum(ones(size(DGW,1),1)*e0.*DGW,2);
    num2 = sum(arg1,2);
```

```
    %-----------------------------------------------
    % ENERGIA
    %-----------------------------------------------
    den2 = sum(DGW.^2,2);
    if not(all(den2))
        den2 = den2+(den2==0);
    end

    G2(:,z) = num2./den2;
    Talpha2 = G2(:,z).*num2;
    [bp,ip] = max(Talpha2);
    beta_vocal(z) = G2(ip,z);
    indice_vocal(z) = ip;

    v = beta_vocal(z).*codebook(indice_vocal(z),:);
    e0 = v;
    [e0,d1b] = ltp(e0,long_subtrama,d1b,idb,[indice_pitch(z) beta_pitch(z)]);
    [e0 d23] = filter(1,lpcs_subtrama(z,:),e0,d23);
    e0 = subtramas(z,:)-e0;
    [e0 d33] = filter(lpcs_subtrama(z,:),fetemp,e0,d33);

    d22 = d23;
    d32 = d33;
    d1a = d1b;
end

%-----------------------------------------------
% Se envian al sintetizador los parametros:
% lpcs - Coeficientes LPC
% beta_pitch - Ganancia del libro de codigos adaptable
% indice_pitch - Indice del libro de codigos adaptable
% beta_vocal - Ganancia del libro de codigos estocastico
% indice_vocal - Indice del libro de codigos estocastico
%-----------------------------------------------

memoria_pitch = zeros(1,idb);
dps = zeros(1,MAXPA);
dss = zeros(1,p);
%-----------------------------------------------
%GENERAMOS LA EXCITACION TOTAL
%-----------------------------------------------
excitacion1 = codebook(indice_vocal,:);

for i=1:size(excitacion1,1)
    vcb1(i,:) = beta_vocal(i).*excitacion1(i,:);
    [vcb1(i,:),dps] = ltp(vcb1(i,:),long_subtrama,dps,idb,[indice_pitch(i) beta_pitch(i)]);
end

for i=1:size(vcb1,1)
    [vcb(i,:) dss] = filter(1,lpcs_subtrama(i,:),vcb1(i,:),dss);
end

%-----------------------------------------------
% POSTFILTRADO
%-----------------------------------------------
global ipZ opZ TC

ip = 0;
op = 0;
dp1 = zeros(1,p+1);
dp2 = zeros(1,p+1);
dp3 = zeros(1,2);
ipZ = 0;
```

```
opZ = 0;

TC = 0.01;
ALPHA = 0.8;
BETA = 0.5;

salida = vcb;

for i=1:size(vcb,1)
    [vcb(i,:),ip,op,dp1,dp2,dp3] = postfiltro(vcb(i,:),long_subtrama,ALPHA,BETA,ip,op,dp1,dp2,dp3,lpcs_subtrama(i,:),p);
end

%-----------------------------------------------
% SNR y SEGSNR
%-----------------------------------------------
X = reshape(vcb',1,prod(size(vcb)));
X1 = reshape(salida',1,prod(size(salida)));
X2 = reshape(subtramas',1,prod(size(subtramas)));

[SNR, SEGSNR_prom] = SNRs(X2',X1,fs);
SEGSNRs = SEGSNR(X2',X1,fs);


soundsc(Y,fs)
pause(3)
soundsc(X,fs)

Xwav=X./max(abs(X));
wavwrite(Xwav,fs,char([archivo 'OUT1' '.wav']));
```

Conclusiones

El estudio de diversas técnicas de codificación es fundamental para la comprensión de los algoritmos de codificación de voz que se emplean en la actualidad. La mayoría de ellos hacen uso de conceptos que fueron desarrollados e implementados hace ya varias décadas. Debido a los requerimientos cada vez más exigentes de las aplicaciones para las telecomunicaciones, no pueden ser usados los codificadores de altas tasas de bits, como ADPCM, aunque producen una calidad de voz muy buena; y tampoco pueden ser usados los codificadores de muy bajas tasas de bits, como los vocoders LPC, ya que la calidad, que ya es menor a una calidad de red, se vería terriblemente degradada por el medio de transmisión.

El algoritmo estándar de mejor rendimiento es por tanto CELP, lo cual ha quedado demostrado al ser elegido como algoritmo de codificación de la telefonía celular digital. Esto da indicios de que el estudio en esta área no ha avanzado en más de 20 años (ya que CELP fue ideado en la década de los 80) y que durante ese tiempo las investigaciones sólo se han dedicado al mejoramiento de éste sistema

Debido a que el estudio sobre la codificación de voz se encuentra bastante desarrollada, resulta complicado proponer técnicas innovadoras que realmente impliquen una mejoría en el funcionamiento de un codificador de voz. Sin embargo, para lograr este cometido es primero necesario poseer un entendimiento profundo del tema desde sus orígenes, conocer el trabajo realizado a la fecha y las tendencias en la investigación.

Aunque la tendencia actual es desarrollar codificadores de tasas de bits variables, los principios básicos son similares a los usados por los codificadores de velocidad completa. Por esto se decidió optar por el análisis de los estándares de tasa completa para identificar alguna etapa en la codificación que pudiera modificarse con facilidad sin afectar las demás etapas del codificador.

El diseño de un codificador de voz implica la determinación de una serie de parámetros que caracterizarán tres etapas principales: el análisis espectral de la voz, el análisis en tiempo largo de la voz y el modelado del residuo. Sería muy difícil tratar de investigar de manera conjunta estas tres etapas, plantear diversas soluciones, y a la vez obtener un codificador funcional; es por esto que el diseño presentado en este trabajo se basa principalmente en un estándar, el CELP FS1016.

El mayor problema que posee el algoritmo CELP es el tratamiento de la señal residual, por eso la mayoría de las investigaciones realizadas se han orientado a atacar ese punto y los algoritmos que de ahí se produjeron, como ACELP

y VSELP, son una sólo una variación de CELP. Consecuentemente, las modificaciones realizadas al codificador CELP FS1016 estuvieron orientadas en esta dirección. Proponiendo dos codificaciones similares pero diferentes en cuanto a la obtención de la excitación correspondiente al libro de códigos fijo, se obtuvieron mejoras en SNR en comparación con el SNR obtenido para el algoritmo original, a cambio de aumentar en cierta medida la tasa de bits, pero manteniéndola por debajo de los 7 kbps.

Lo más importante en este estudio es la relación entre calidad de reconstrucción del algoritmo y la capacidad necesaria para la transmisión. La importancia de esta relación se basa en que es inútil un algoritmo que logre una gran compresión si la calidad de la señal se pierde, o viceversa, un algoritmo que pueda reconstruir la señal de forma idéntica a la original, pero que posea una mala compresión.

La tasa de bits se puede incrementar en muy diversas formas, sin embargo no todas de ellas producen una calidad de voz mejorada. Por lo tanto, el estudio de un codificador de voz requiere del conocimiento de las propiedades de la voz así como del oído de forma que puedan ser explotados de la mejor manera posible.

Antes de llegar a las modificaciones realizadas sobre el algoritmo FS1016, se realizaron otras muchas modificaciones básicas para encontrar las que podrían funcionar mejor. Por ejemplo, se probó el incremento en el número de coeficientes LPC, sin embargo éste no mejora la calidad de la voz reconstruida en una medida notable. También se redujo el tamaño de las tramas, por ejemplo a 20 ms, lo cual proporciona una mejoría en la calidad de la voz menor a 2 dB incrementando la tasa de bits 2.4 kbps. Agregando un bit al libro de códigos fijo, esto es, incrementando el tamaño del libro de códigos a 1024 vectores candidatos, la mejora es de alrededor de 0.2 dB sobre el SNR de CELP FS1016 que usa 512 vectores, y la tasa de bits aumenta 133.3 bps. Además se realizaron otras modificaciones al libro de códigos fijo con el fin de mantener la tasa de bits usada por FS1016 e incrementar la calidad, sin embargo no se obtuvieron mejoras audibles de la calidad de la voz reconstruida.

Finalmente, las modificaciones adoptadas se basaron en tratar de reducir el residuo de la señal mediante dos métodos: el primero, reduciendo el tamaño de la subtrama para la cual se obtendría un vector del libro de códigos; y el segundo, mediante dos búsquedas en el libro de códigos fijo, usando un método similar a VSELP.

El método llamado FS1016 modificado probó comportarse de forma bastante aceptable tanto en SNR como en MOS, aunque los SNR de ambos codificadores fueron similares, la evaluación subjetiva determinó que el método llamado VSELP modificado presentaba una mayor degradación en la señal reconstruida que el FS1016 modificado. Las notas obtenidas reflejan únicamente una comparación entre los tres codificadores simulados, ya que las condiciones en que se realizaron las pruebas no fueron las óptimas, sin embargo se procuró que se apegaran lo más posible a la metodología estandarizada por la ITU para este tipo de pruebas.

A pesar de que la propuesta presentada para el diseño de un codificador CELP podría considerarse bastante sencilla y básica, el procedimiento para la búsqueda de una alternativa que proporcione calidad de voz mejorada, aunque aumentando en cierta medida la tasa de bits, no es tarea sencilla. Y aún más pensar que un cierto codificador propuesto pueda ser considerado como estándar para telefonía celular debe ser mucho más complicado. Sin embargo, para poder llegar a diseñar completamente un codificador de voz se debe empezar diseñando alguna de sus etapas de codificación.

La transmisión de la voz siempre será un asunto de suma importancia para las telecomunicaciones, pues como es bien sabido, es el principal medio de comunicación entre los humanos. Es por esto que los estudios relacionados con el procesamiento de la voz poseen un gran potencial en cuanto a su desarrollo. Seguramente en años posteriores habrá nuevos algoritmos para la codificación de voz, pues la investigación al respecto sigue avanzando; sin embargo, por el momento parece necesario utilizar las técnicas que han probado ser exitosas en ese aspecto a lo largo de muchos años y seguir trabajando sobre ellas.

Acrónimos

ACELP Algebraic CELP

ACR Absolute Category Rating

ADM Adaptive Delta Modulation

ADPCM Adaptive Differential Pulse Code Modulation

ADPCM-FB ADPCM-FeedBack

ADPCM-FF ADPCM-FeedForward

APC Adaptive-Predictive Coding

APCM Adaptive Pulse Code Modulation

ATC Adaptive Transform Coding

CCITT International Telephone and Telegraph Consultative Committee

CDMA Code Division Multiple Access

CELP Code-Excited Linear Prediction

CS-ACELP Conjugate Structure ACELP

DDVPC U.S. Department of Defense Digital Voice Processor Consortium

DM Delta Modulation

DPCM Differential Pulse Code Modulation

DSP Digital Signal Processing

EFR Enhanced Full Rate

ETSI European Telecommunications Standards Institute

FB FeedBack

FF FeedForward

FFT Fast Fourier Transform

FIR Finite Impulse Response

FR Full Rate

GSM Global System for Mobile Communications

HMM Hidden Markov Models

HR Half Rate

IFFT Inverse Fast Fourier Transform

IIR Infinite Impulse Response

ISO International Organization for Standardization

ISPP Interleaved Single Pulse Permutation

ITU International Telecommunications Union

JDC Japanese Digital Cellular

LAR Log Area Ratio

LD-CELP Low Delay CELP

LPC Linear Predictive Coding

LPC-AS LPC-Analysis by Synthesis

LSF Line Spectrum Frequencies

LSP Line Spectrum Pairs

LTP Long Term Predictor

MOS Mean Opinion Score

MPEG Moving Picture Experts Group

MPLPC MultiPulse-Excited Linear Prediction Coding

MSPE Minimum Squared Prediction Error

PARCOR PARtial CORrelation

PCM Pulse Code Modulation

PSTN Public Switched Telephone Network

RELP Residual-Excited Linear Prediction

RPE-LTP Regular Pulse Excitation with Long-Term Prediction

SBC SubBand Coding

SD Spectral Distortion

SEGSNR SEGmental SNR

SEV Self-Excited Vocoder

SNR Signal-to-Noise Ratio

STFT Short-Time Fourier Transform

STP Short Term Predictor

SVD Singular Value Decomposition

TDMA Time Division Multiple Access

TIA Telecommunications Industry Association

VLSI Very Large Scale Integrated

VQ Vector Quantization

VSELP Vector Sum-Excited Linear Prediction

Bibliografía

[1] Barnwell, Thomas P.; Nayebi, Kambiz and Richardson, Craig H., *Speech Coding: A Computer Laboratory Textbook,* The Georgia Tech Digital Signal Processing Laboratory Series. John Wiley & Sons, Inc., 1996

[2] Saito, Shuzo and Nakata, Kazuo, *Fundamentals of Speech Signal Processing.* Academic Press, Inc., 1985

[3] Rabiner, L. R. and Shafer, R. W., *Digital Processing of Speech Signals,* Prentice-Hall Signal Processing Series. Prentice Hall, Inc., 1978

[4] Rabiner, L. R. and Juang, *Fundamentals of Speech Recognition.* Prentice Hall, Inc., 1993

[5] Papamichalis, Panos E., *Practical approaches to speech coding,* Prentice-Hall and Texas Instruments Digital Signal Processors Series. Prentice-Hall, Inc., 1987

[6] Unión Internacional de Telecomunicaciones, *Voz codificada digitalmente en el servicio móvil terrestre,* Recomendación ITU-R M.1309, 1997.

[7] Woodard, J.P. and Hanzo, L., "Improvements to the analysys-by-synthesis loop in celp codecs", In *Proceedings of the Radio Receivers and Associated Systems Conference, IEE RRAS'95,*Bath, UK: Sept. 1995
`http://www-mobile.ecs.soton.ac.uk/papers/papers.html`

[8] International Telecommunications Union. *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited-Linear-Prediction (CS-ACELP),* ITU-T Recommendation G.729, 1996

[9] Unión Internacional de Telecomunicaciones. *Métodos de determinación subjetiva de la calidad de transmisión,* Recomendación ITU-T P.800, 1996

[10] Hasegawa-Johnson, Mark and Alwan, Abeer., "Speech Coding: Fundamentals and Applications." University of Illinois at Urbana-Champaign. Image Formation and Processing Group
`http://www.ifp.uiuc.edu/ifp_home/index.html`

[11] Robinson, A. J., "Speech Analysis." University of Cambridge, Department of Engineering: Machine Intelligence Laboratory
`http://mi.eng.cam.ac.uk/~ajr/SpeechAnalysis/`

[12] Spanias, Andreas S., "Speech coding: A tutorial review," In *Proceedings of the IEEE,***volume**82,Oct. 1994

[13] Flanagan, J., *Speech Analysis, Synthesis and Perception,* 2nd Edition. Springer-Verlag, 1972

[14] Mak, M.W., "FS1016 CELP codec." The Hong Kong Polytechnic University, Department of Electronic and Information Technology: Center for Multimedia Signal Processing, 2000
www.en.polyu.edu.hk/~mwmak/mypage.htm

[15] Islam, Tamanna and Kabal, Peter. "Partial-Energy weighted interpolation of linear prediction coefficients", In *Proceedings of the IEEE Workshop Speech Coding.* McGill University, Electrical and Computer Engineering, Sept. 2000

[16] Painter, Ted and Spanias, Andreas. "Matlab Simulation of NSA FS1016 CELP v3.2." Copyright 1996-2002

[17] Koestoer, Nanda P. *Robust Linear Prediction Analysis for Low Bit-Rate Speech Coding,* PhD Thesis. Griffith University, 2002

[18] Chan, W.; Gerson, I. and Miki,T. "Half-Rate Standards," *Mobile Communications Handbook.* Suthan S. Suthersan, 1999

[19] Herrera Camacho, Abel. "Procesamiento digital de voz." Apuntes del curso. Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2004

[20] Chien-Kuang Lin. *Low-Complexity Codebook Searching Algorithms for FS1016,* MSc Thesis. National Chiao Tung University, Department of Commmunications Engineering, 2002

[21] Erkelens, Johan S., *Autoregressive Modelling for Speech Coding: Estimation, Interpolation and Quantisation.* Ph.D. dissertation. Publications of the Signals, Systems and Control Group. Delft University of Technology, Delft Center for Systems and Control.Holland, 1996
http://www.dcsc.tudelft.nl/Research/PubSSC/

[22] Bustos Jiménez, Javier A., *Estudio de Sistemas de compresión de voz digital orientado a telefonía celular.*Universidad de Concepción, Departamento de Ingeniería Informática y Ciencias de la Computación. Chile, 2002
http://www.inf.udec.cl

[23] Ares, Roberto. *Manual de las Telecomunicaciones,* Compresión de señales vocales, audio y sonido. Argentina, 2000
http://www.rares.com.ar/

[24] Macres, Jason V. *Theory and Implementation of the Digital Cellular Standard Voice Coder: VSELP on the TMS320C5x,* Application Report. Texas Instruments, 1994

[25] Baudoin, G.; Blaha, P. *Developing a Low Bit Rate Speech Coder Based on the Half Rate GSM Standard with the TMS320C30 DSP.* Texas Instruments. Paris, 1996

[26] Telecommunications Industry Association. *TDMA Cellular/PCS - VSELP.* TIA/EIA Standard 136-420, 1999.

[27] Telecommunications Industry Association. *TDMA Cellular/PCS - Radio Interface Enhanced Full-Rate Voice Codec.* TIA/EIA Standard 136-410, 1999.

[28] *Digital cellular telecommunications system; Half rate speech (GSM 06.20).* European Standard (Telecommunications series) ETSI EN 300 969 V8.0.1. European Telecommunications Standards Institute, 2000

[29] European Telecommunications Standards Institute. *Digital cellular telecommunications system; Enhanced Full Rate speech transcoding (GSM 06.60).* European Standard (Telecommunications series) ETSI EN 300 726 V8.0.1., 2000

[30] Campbell, Joseph P. Jr.; Welch, Vanoy C. and Tremain, Thomas E. *An Expandable Error-Protected 4800 bps CELP Coder (U.S. Federal Standard 4800 bps Voice Coder)*, in Proceedings of ICASSP, 1989

[31] Jarvinen, K. et al., "GSM Enhanced Full Rate Speech Codec", in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing.* ICASSP'97. Vol.2, pp.771-774, 1997.

[32] Honkanen, T. et al., "Enhanced Full Rate Speech Codec for IS-136 Digital Cellular System", in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing.* ICASSP'97. Vol.2, pp.731-734, 1997.

[33] Fang ZHENG, Zhanjiang SONG, Ling LI, Wenjian YU Fengzhou ZHENG and Wenhu WU. "The Distance Measure for Line Spectrum Pairs Applied to Speech Recognition", in *International Conference on Spoken Language Processing.* ICSLP'98. Vol. 3 pp. 1123-1126, 1998.

[34] Grassi, Sara. *Optimized Implementation of Speech Processing Algorithms*, PhD Thesis, University of Neuchatel, Institute of Microtechnology, February 1998.
`http://www-imt.unine.ch/www/grp_pe/publications/thesis.htm`