Sistema de Síntesis de Voz en Español de México

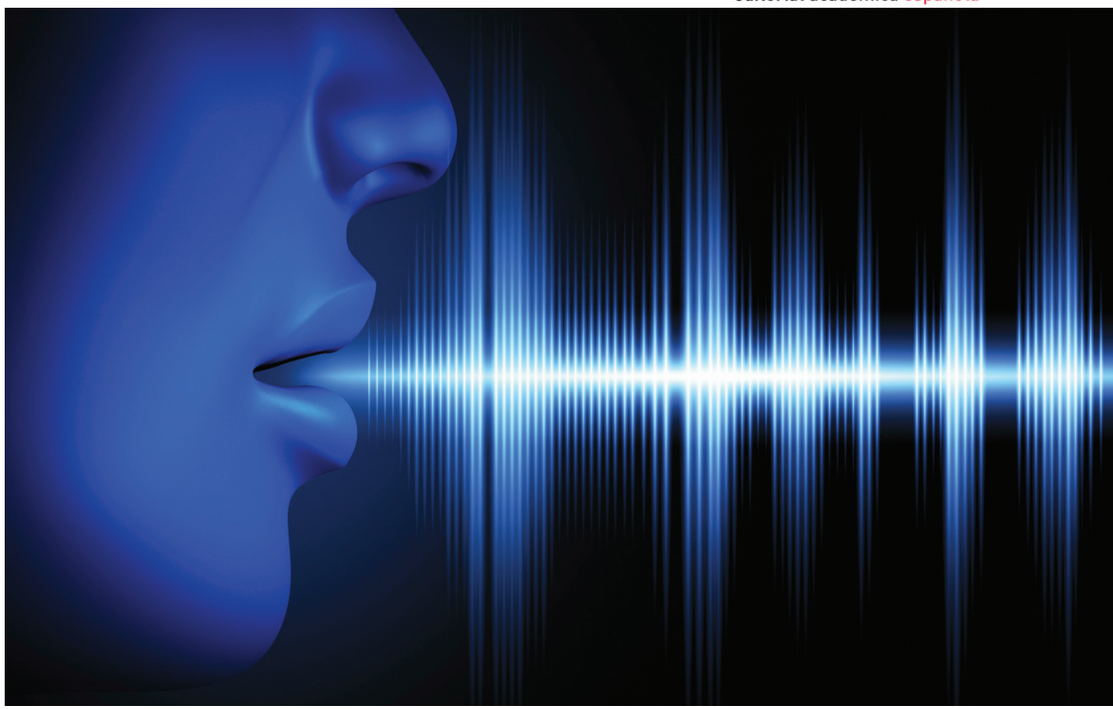
Se realizó la implementación de un sintetizador de voz con acento mexicano como parte de los proyectos de últimos cinco años en el Laboratorio de Tecnologías del Lenguaje de la UNAM. Este libro presenta un resumen de las diferentes teorías de síntesis de voz así como la parametrización de una señal de voz usando Par Lineal Espectral LSP para ser implementada en un sintetizador HTS. Se menciona también en el libro el princpio de funcionamiento de Modelos Ocultos de Markov y los resultados de las pruebas realizadas al sistema.

Carlos A. Franco (Pue, México 1977) hizo sus estudios de licenciatura en la BUAP, maestría en la Universidad de York Inglaterra y doctorado en la UNAM. Su área de especialidad en los últimos años es la Síntesis de Voz. Actualmente labora como Profesor-Investigador en la BUAP y colabora con los co-autores en proyectos de investigación en la UNAM.

978-620-2-16538-9

editorial académica española

editorial académica española



Carlos Franco · Abel Herrera · Boris Escalante

Sistema de Síntesis de Voz en Español de México

Sistema basado en Hidden Markov Models as Text to Speech Synthesis

Carlos Franco
Abel Herrera
Boris Escalante

Sistema de Síntesis de Voz en Español de México

Carlos Franco
Abel Herrera
Boris Escalante

Sistema de Síntesis de Voz en Español de México

Sistema basado en Hidden Markov Models as Text to Speech Synthesis

Editorial Académica Española

Imprint
Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: www.ingimage.com

Publisher:
Editorial Académica Española
is a trademark of
International Book Market Service Ltd., member of OmniScriptum Publishing Group
17 Meldrum Street, Beau Bassin 71504, Mauritius

Printed at: see last page
ISBN: 978-620-2-16538-9

Contenido

Capítulo 1

Los actuales sistemas de síntesis de voz gozan de distintos usos, por ejemplo: Ayuda para gente con problemas de habla, auxiliar de lectura para invidentes, instrucciones de GPS para automovilistas o indicaciones verbales en cajeros automáticos entre muchas otras. Todos estos sistemas son normalmente bastante eficientes en términos de inteligibilidad, sin embargo, todos ellos tienen un problema en común: La falta de naturalidad.

Buscamos naturalidad en los sistemas de voz artificial porque si bien es cierto que en ocasiones es suficiente entender un mensaje sin importar su forma, las personas en general se sienten más atraídas y por tanto son más propensas a comprender mejor un mensaje cuando viene de otra persona. Se piensa que la interacción hombre-máquina sería más sencilla de llevarse cabo verbalmente.

De ahí que la prueba de Turing, la cual pretende valorar la capacidad de una máquina a exhibir comportamiento semejante al humano, continúe siendo referente en nuestros días. En el campo de investigación de tecnologías del lenguaje es una meta permanente el conseguir un sistema de síntesis de voz que sea indistinguible del habla humana.

Los sintetizadores de voz que más se aproximan a la meta son de la clase que producen la síntesis mediante la concatenación de fonemas y difonemas. Se ha observado también que el reto para lograr naturalidad no está exclusivamente en la calidad del corpus de difonemas que se utilizan para generar frases, sino en cuál es la mejor opción de difonema de acuerdo con el contexto de la frase.

Existen diferentes tipos de sistemas para selección de frase, todos se basan en esquemas de árboles de probabilidad determinísticos o estocásticos. Los más exitosos de últimos años pertenecen al segundo grupo, destacando los Modelos Ocultos de Markov *Hidden Markov Models* HMM propuestos por Tokuda (Keiichi Tokuda et al., 2013) este trabajo es referencia fundamental en el proyecto descrito en el presente documento.

El corpus de difonemas además de ser de buena calidad en términos de sonido, debe también ser compacto para reducir costos en términos de almacenamiento y procesamiento de información.

Por esta razón se han buscado mecanismos para disminuir el tamaño de los archivos de sonido. Cuando el contenido tímbrico es relevante, se puede echar mano de los sistemas de compresión de archivos de sonido como es el caso de los .mp3.

En el caso de síntesis de voz, se prescinde de la compresión porque no estamos tan interesados en conservar la energía acústica de la señal de voz, pero sí en tener un mapeo preciso que nos permita reproducir las frecuencias formantes de los diferentes fonemas. Por esta razón se utilizan métodos de parametrización de voz que conserven tales frecuencias y eliminen el resto de la señal. Los más populares parten de dos esquemas: Coeficientes Cepstrales de Frecuencia Mel *Mel Frequency Cepstral Coefficients* MFCC[1] y los Coeficientes de Predicción lineal *Linear Predictive Coding* LPC (Holmes & Holmes, 2001).

1.1 Planteamiento de Problema

Mucho se ha probado y documentado en lo referente a síntesis de voz utilizando parametrización de frecuencia Mel-Cepstral MFCC. La codificación predictiva lineal LPC por su parte ha quedado relegado al ámbito de reconocimiento de voz debido a que en síntesis produce una voz en exceso artificial. Sin embargo, existen variantes de LPC, destacando el Par Lineal Espectral *Line Spectral Pair* (McLoughlin, 2008) el cual se ha aplicado a reconocimiento con cierto éxito y se tiene documentado un primer intento en síntesis (Chennoukh, Gerrits, & Miet, 2001; Franco, Herrera, & Del Río, 2016; Nakatani, Yamamoto, & Matsumoto, 2006; Soong & Juang, 1984). Dicho intento fue trabajo de Naktani y colegas, debido a que luego de hacer algunas pruebas, notaron que las formantes de los sonidos vocales son menos planas parametrizadas con LSP que con MFCC.

Además de reconocimiento y síntesis, LSP se ha usado en ciertas variantes de modificación de voz usando STRAIGHT (Speech Transformation and

[1] A lo largo del documento se utilizarán los acrónimos de acuerdo con las siglas en inglés de los diferentes sistemas de parametrización de voz, ya que resultan de uso universal.

representation using adaptive interpolation of weighted spectrum) como lo hicieran (Arakawa, Uchimura, Banno, Itakura, & Kawahara, 2010; Kang & Liu, 2006). Por su parte (Sagayama & Itakura, 2002) proponen el uso de un modelo dual a LPC conocido como Composite Sinusoidal Model CSM donde también se obtienen las LSP.

MFCC si bien es eficiente, tampoco es óptimo en términos de naturalidad e inteligibilidad. Además de la poca precisión en su espectro, el cual cómo se menciona en (Nakatani et al., 2006) es mucho más plano que el espectro de los sonidos vocales originales. Por esta razón se decidió probar LSP como una alternativa de parametrización que pudiera en un momento dado mejorar lo conseguido con MFCC.

Se han hecho ya trabajos previos de síntesis de voz en español utilizando MFCC (Herrera-Camacho & Ávila, 2013) denominado HTS-MFCC pero no hay síntesis de voz en español utilizando LSP por lo que se buscó implementar y documentar un sistema que la empleara (Franco, Herrera, et al., 2016) se denomina HTS-LSP.

Defendemos también el uso de LSP porque al estar basado en LPC viene directamente de un modelado físico del tracto vocal visto como filtro. Por otro lado, se puede revertir el proceso y hacer una reconstrucción de la señal de voz original, a diferencia de MFCC donde esto no es posible. Finalmente, los archivos generados con LSP son de menor tamaño, lo cual implica una economía de recursos computacionales en procesamiento y almacenamiento.

1.2 Objetivos del proyecto

- Evaluar el sistema existente HTS-MFCC.
- Agregar al sistema la parametrización LSP y conseguir un HTS-LSP.
- Evaluar y comparar ambos sistemas.

1.3 Metodología

Se parte de la hipótesis de que el actual sistema de síntesis HTS que ya se ha diseñado utilizando parametrización MFCC (Herrera-Camacho & Ávila, 2013) proporciona alto grado de inteligibilidad y casi naturalidad. Para objetivar estos supuestos se hicieron las pruebas MOS correspondientes valorando

naturalidad e inteligibilidad. Se publicaron los resultados (Franco, Del Rio, & Herrera, 2016). En el capítulo 4 se encuentran los detalles del experimento.

Posteriormente, se hizo una propuesta sobre un nuevo tipo de parametrización con base en LSP. La parametrización se utilizó en HTS. Una vez implementada la parametrización como entrada en el sistema HTS, se sintetizaron algunas frases y fueron sometidas a una segunda serie pruebas tipo MOS. Se evaluaron nuevamente dos aspectos de la voz sintetizada: naturalidad e inteligibilidad.

En esta serie de pruebas se hizo una comparación tanto de la voz parametrizada con MFCC como de aquella usando LSP. En todo momento la referencia fue la voz original del locutor de la voz con la que se entrenó el sistema.

De acuerdo a los resultados arrojados por las pruebas MOS (Franco, Herrera, et al., 2016), la voz parametrizada con LSP tuvo calificaciones más altas que la de MFCC. Además, las opiniones de los evaluadores fueron mucho más consistentes en las calificaciones otorgadas.

Se decidió someter a la voz sintetizada a una serie de pruebas de reconocimiento de hablante usando un sistema también desarrollado en el Laboratorio de Tecnologías del Habla por (Trangol & Herrera, 2015). La comparación de hablantes se hizo entre el locutor de la voz original y la voz sintetizada. El resultado de la comparación mostró que hay alta probabilidad de que ambas voces provengan del mismo hablante.

Con los resultados de las pruebas MOS y la de reconocimiento de hablante, podemos afirmar que la voz parametrizada usando LSP es una buena alternativa para generar voz sintetizada.

1.4 Estado del Arte

Desde principios del siglo XX se han realizado distintos esfuerzos para generar “máquinas parlantes”, capaces de realizar Síntesis de Voz. Sin embargo, a casi un siglo de que apareció el primer sintetizador de voz eléctrico que se tiene documentado -VODER de Homer Dudley- (“Homer Dudley’s Speech Synthesisers,” n.d.) . No se ha terminado de lograr el objetivo de tener un sistema de síntesis de voz que resulte indistinguible de la voz humana. Si bien

las voces sintéticas de la actualidad cumplen casi cabalmente el requisito de inteligibilidad, aún no es así con el de la expresión. Es la combinación de naturalidad e Inteligibilidad lo que da realismo a los sistemas de voz sintetizada.

1.4.1 Métodos Representativos de Síntesis de Voz

Existen tres sistemas de síntesis vocal: síntesis de formantes, síntesis articulatoria y síntesis concatenativa. A continuación, se explica con detalle en que consiste cada uno.

1.4.2 Síntesis de Formantes

Se define como *frecuencias formantes* a aquellas frecuencias características de un fonema. Sin importar el hablante, las frecuencias formantes permanecen constantes en cada emisión de frase, independientemente de la entonación o intensidad con la que haya sido producida. Gracias a esta característica sabemos que los fonemas pueden ser relacionado en todo momento por estas frecuencias.

Fisiológicamente hablando, las frecuencias formantes son resultado de las resonancias producidas a lo largo del tracto vocal. Son modificaciones a la onda sonora proveniente de la glotis que tuvo su origen en la vibración de las cuerdas vocales producida por una corriente de aire en los pulmones.

En la voz humana existen dos tipos de sonidos: vocales y sordos (o no-vocales). Los primeros son resultado de la vibración de las cuerdas vocales y los segundos resultan del flujo de aire que pasa directamente de los pulmones al tracto vocal.

Este proceso de generación artificial de formantes se puede lograr en un sistema de procesamiento de señales electrónicas. La señal proveniente de las cuerdas vocales se simula con una fuente sinusoidal. Los sonidos no-vocales, por su parte, se emulan a través de una fuente de ruido blanco. Las frecuencias formantes se consiguen pasando dicha fuente a través de un conjunto de filtros pasa banda. Un modelo que ha sido referente en este tipo de sistemas de fuente-filtros es el sintetizador de Klatt (Klatt, 1982) el cual fue de los primeros sistemas de síntesis en software, cuyo algoritmo y código fuente se publicaron a detalle.

1.4.3 Síntesis Articulatoria

La síntesis articulatoria está basada principalmente en el trabajo de Fant (Fant, 1970) que comenzó desde principios de los 60. Este tipo de síntesis pretende modelar las características físicas haciendo un estudio de la geometría del tracto vocal, principalmente de su largo y de su área transversal. Posteriormente mediante ecuaciones de movimiento de fluidos se hace un modelo matemático de los fenómenos acústicos que tienen lugar adentro del tracto.

El concepto físico de la presión que el aire ejerce sobre el tracto vocal, así como el chorro de aire que viaja dentro de él se simplifica observando el tracto vocal como una serie de tubos interconectados. Así como el tejido del tracto vocal cambia su grosor de acuerdo con el sonido que se emite, cada uno de estos tubos tiene un diámetro distinto correspondiente a un fonema determinado.

Este modelo tubular es referente en dos tipos de síntesis: la primera denominada Circuitos Acústicos y la segunda Linear Predictive Coding o LPC. Se hablará de LPC y cómo utiliza el modelo tubular más adelante en este documento, en lo referente a circuitos acústicos podemos mencionar que el modelo tracto vocal-tubular fue muy popular a mediados del siglo veinte ya que constituyó el principio para la elaboración de una familia de sintetizadores de voz eléctricos.

Muchos sintetizadores eléctricos fueron llevados a la práctica fundamentados en analogías acústicas-eléctricas. Destaca el trabajo de Stevens, Kasowski con Fant (Stevens, Kasowski, & Fant, 1953). La síntesis articulatoria perdió un poco de popularidad durante los 60 y 70, no fue sino hasta 1982 con el trabajo de Maeda que se reutilizó la analogía electro-acústica y sin duda al día de hoy el trabajo más relevante donde se emplea síntesis articulatoria es Vocal Tract Lab (Birkholz & Jackel, 2003; Birkholz, Jackèl, & Kroger, 2006), el cual continúa vigente en su interesante proyecto en el sitio www.vocaltractlab.com.

1.4.4 Síntesis Concatenativa

Para hacer síntesis es necesario es necesario enlazar los fonemas uno con otro luego de ser previamente seleccionados de una base de datos llamada

corpus, dichos sonidos pudieron ser previamente grabados o reducidos a su mínima expresión por paramterización. A este tipo de síntesis de voz se le conoce como *síntesis concatenativa*.

La síntesis concatenativa es la más eficiente en sistemas de síntesis hoy día. En la síntesis concatenativa se pueden modificar más detalladamente las unidades mínimas de lenguaje logrando una mayor naturalidad cuando éstos se producen.

Como consecuencia de lo anterior, la inteligibilidad y entonación de una voz artificial de síntesis concatenativa superan a aquellas logradas con síntesis articulatoria o con síntesis de formantes.

Los métodos para emular la prosodia (tono y duración) en la concatenación de las palabras son principalmente los basados en el principio de Suma-Traslape (Overlap-Add), en estos métodos destacan PSOLA, MBROLA y selección de unidades.

Se dice que (Thierry Dutoit, 2008) para producir lenguaje hablado de manera inteligible, se requiere de la habilidad de generar lenguaje continuo coarticulado. Lo cual nos conduce a pensar que los puntos de transición entre fonemas son mucho más importantes para la inteligibilidad de lo que son los segmentos estables. Incluso los fonemas vocales largos y sostenidos varían en amplitud y frecuencia, además de que contienen elementos inarmónicos.

Con base en este argumento, la síntesis de voz concatenativa busca inteligibilidad "pegando" trozos de habla en lugar de fonemas aislados. Esto conlleva a una mejor coarticulación.

1.4.5 Síntesis Concatenativa por difonemas

Un primer intento de lograr una concatenación más precisa es mediante el uso de **difonemas** como unidades mínimas para producir lenguaje hablado.

Normalmente, el difonema comienza y termina con una parte estable como se muestra en la figura:

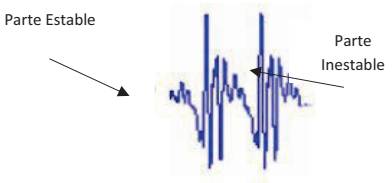


Figura 1.1 “Representación de una señal de voz”

El problema es que la cantidad de difonemas presentes en un idioma es enorme. Típicamente una base de datos de difonemas es de al menos 1500 unidades. En términos prácticos, tres minutos de habla muestreados a 16 KHz con resolución de 16 bit suman alrededor de 5 MB.

Para resolver este problema, se busca una lista de palabras donde aparezca al menos dos veces cada difonema. El texto se lee por un locutor profesional para evitar mucha variación en tono y articulación. Posteriormente, los elementos elegidos son marcados mediante herramientas de visualización o algoritmos de segmentación. Finalmente se recolectan en una base de datos.

A *groso modo*, la manera en cómo se lleva a cabo la síntesis es la siguiente:

1. El sintetizador recibe la entrada fonética y se realiza un procesamiento previo de lenguaje (se hablará más delante de dicho proceso).
2. Se establece duración, tono y tipo de fonema.
3. Se recolecta de la base de datos una serie de fonemas candidatos para llevar a cabo la síntesis.

Por lo general, los fonemas elegidos difícilmente reúnen de manera natural los requerimientos para darle a la frase producida la suficiente inteligibilidad por lo que hay que realizar dos tareas adicionales. La primera tarea consiste en hacer modificaciones en la prosodia. La segunda tarea tiene que ver con la “suavización” de las transiciones de los difonemas ya que son muy notorias debido a las ya mencionadas variaciones de amplitud y frecuencia.

Algunos ejemplos de síntesis por difonemas se encuentran en www.francocarlos.com[2] en los audios a continuación: *fest_diphone_ked.wav, fest_diphone_rab.wav, fest_diphone_esp.wav.*

[2] Todos los ejemplos de audio citados en éste trabajo se encuentran en el sitio web www.francocarlos.com

1.4.6 Linear Predictive Coding LPC

El sistema de Codificación Lineal Predictiva (Linear Predictive Coding) conocido como LPC, (Holmes & Holmes, 2001) es uno de los diferentes métodos para la producción de voz de manera artificial. Este sistema parte de una aproximación electrónica al sistema fisiológico en donde la vibración de las cuerdas vocales es una señal sinusoidal que produce los sonidos vocales y una fuente de ruido blanco para los no-vocales. El tracto vocal es modelado como un sistema de filtrado cuyas bandas corresponden directamente a las frecuencias formantes del sonido vocal a reproducir. La característica fundamental de LPC es que la señal de voz viene reducida a su expresión más simple. Se conservan solamente las frecuencias formantes del fonema a producir y se deja de lado la vibración que la produjo.

Esto se hace con objeto de ahorrar información, ya que LPC privilegia el contenido del discurso sobre naturalidad del hablante. Es por eso que la voz resultante en la síntesis por LPC tiene esa característica "robótica" en su timbre. El archivo *sintesis_lpc_after.wav* contiene un ejemplo de este tipo de síntesis, se puede escuchar el audio original en el archivo *sintesis_lpc_before.wav*

La implementación en software de un sistema LPC está basada en la expresión matemática (1.1). Donde $S(n)$ representa la señal de voz original, la suma de dicha señal retrasada k muestras pasadas desde 1 hasta p multiplicadas por sus amplitudes A_k es su aproximación artificial. Finalmente, $e(n)$ es la diferencia o error existente entre ambas.

$$S(n)=\sum_{k=1}^{p} A_k S(n-K) + e(n) \quad (1.1)$$

La función de transferencia está representada en la ecuación (1.2), en donde la expresión (1.1) puede entenderse como un filtro $A(z)$ cuya entrada es $E(z)$ y su salida $S(z)$.

$$\frac{S(z)}{E(z)} = \frac{1}{1-\sum_{k=1}^{p} A_k z^{-k}} = \frac{1}{A(z)} \quad (1.2)$$

La reconstrucción de la señal es obteniendo los valores de los diferentes coeficientes del filtro A_k se hace mediante un sistema de ecuaciones simultáneas que se resuelve por matrices. El método de resolución se conoce como Levinson-Durbin y está ampliamente documentado en la literatura especializada. Generalmente se eligen 13 coeficientes para conservar la inteligibilidad de la frase como fue el caso en el audio arriba mostrado. Este ejemplo fue implementado en Matlab por Carlos Acosta en el Laboratorio de Tecnologías del Lenguaje de la Facultad de Ingeniería de la UNAM.

El sistema LPC ha sido ya superado por otros sistemas de síntesis de voz que son capaces de reconstruir la señal de voz con mucha mayor naturalidad e inteligibilidad. Se menciona en este texto dado que es la base teórica de la parametrización de voz del Par Espectral Lineal o *Line Spectral Pair* (LSP) el cual sostenemos que es una buena alternativa vigente que puede fácilmente ser una alternativa a la parametrización basada los coeficientes de Frecuencia Mel *Mel-Frequency Cepstral Coefficients* (MFCC) que son un estándar en el ramo.

El sistema de filtrado de una señal sinusoidal/ruido blanco sigue en uso, se le conoce como Vocoder *(Voice-coder)*. Es muy eficiente para la producción de sonidos vocales, independientemente del sistema previo de selección de fonemas que se haya empleado. Este sistema por ejemplo es el recurso de hacer síntesis en el sistema HTS *Hidden Markov Models as Text to Speech Synthesis*.

1.4.7 Métodos Overlap Add OLA

Se ha visto que modificar duración y tono en una frase no son operaciones triviales. De manera intuitiva, el lector podría pensar que, modificaciones a tono y duración se consiguen interpolando muestras y re-muestreando la señal. Los resultados de realizar tal proceso equivalen a aquellos observados cuando se modifica la velocidad de reproducción de una cinta de audio analógica, es decir: el tono sube o baja de manera exagerada. Se han buscado alternativas para resolver éste problema, uno de los más eficientes ha sido el procesamiento de la señal mediante un algoritmo conocido como TD-PSOLA (Stylianou, 2008) *Time Domain Pitch Synchronous Overlap Add* (Fragmentación y traslape de la señal sincronizada en tono en dominio del tiempo). Tal cual su nombre lo indica, el algoritmo tiene la siguiente estructura:

1. Se analizan los distintos periodos en la señal de voz y se colocan indicadores de tono (*pitch marks)*
2. Se hace un ventaneo (fragmentación de la señal) con una cierta duración.
3. Se identifica la frecuencia fundamental F_0 en cada uno de los segmentos contenidos en las ventanas.
4. Si se desea aumentar la duración, se repiten ciertos segmentos para aumentar el periodo. Si por el contrario la intención es volverla más corta, se eliminan algunos segmentos.
5. Si se desea cambiar el tono se reacomodan las ventanas con modificaciones de la duración entre una y otra, dependiendo si se quiere aumentar o disminuir la frecuencia.
6. Finalmente se suman las ventanas resultantes para realizar la síntesis

Los archivos nombrados a continuación muestran ejemplos de síntesis usando TD-PSOLA, *salida_psola.wav* y *salida_psola_entonacion.wav* muestran sonido sintetizado a partir de texto. La diferencia entre ambas es la entonación que fue modificada. El tercer archivo *tdpsola_pruebasonido.wav* muestra una señal de voz grabada sin modificaciones y la cuarta es ésta misma señal con modificaciones en tono y duración. A continuación, presentamos los detalles del algoritmo arriba mencionado:

Se tiene una señal de voz como se mostró en la figura 1.1. En esta señal es necesario hacer una detección de las partes periódicas de la misma, para ello hay varios, nosotros nos basamos en el procedimiento propuesto por Goncharoff (Goncharoff & Gries, 1998). En primer lugar, se buscan secuencias numéricas que se incrementen y decrementen con cierta proporción. Una vez hallados estos periodos se identifican mediante marcas de altura de tono o *pitch marks.* Posteriormente se separa la señal en tramas o *frames*, cada frame tiene una duración de dos periodos. La ventaja de tener estas ventanas como unidades aisladas es que podemos combinarlas teniendo sus puntos centrales en la frecuencia principal. Luego se traslapan unas con otras y se tiene una reconstrucción de la señal original. La figura 1.2 muestra un diagrama de esta.

1.4.8 Modificaciones de tono y duración.

Precisamente la ventaja de hacer esta separación de la señal en tramas es lo que nos permite hacer modificaciones en duración y tono. Para modificar la duración es necesario duplicar algunas de las tramas.

Por su parte si se busca un acortamiento de la duración de la señal, algunas de las tramas deben ser eliminadas. La figura 1.4 ilustra dicho concepto. Se recomienda la ventana de dos periodos para facilitar la reconstrucción de la onda en el momento del traslape, así como se ilustra en la figura 1.3.

La modificación del tono se logra mediante la recombinación de las tramas. En tal caso es necesario modificar la duración de las marcas de tono. Vale la pena mencionar un ejemplo para ilustrar el concepto:

- Se tiene un segmento de voz con un tono de 100 Hz (10 ms entre cada pitch mark)
- Se realiza el ventaneo de Hanning
- Sí se colocan las ventanas a una distancia de 9 ms y luego se hace la suma-traslape, se obtendrá ahora un tono de 111 Hz.

1.4.9 Detalle de la manipulación de las Marcas deTono

Como se puede apreciar en las secciones anteriores la base del método TD PSOLA, los elementos críticos son las marcas de tono. Se ha dicho que es necesario modificarlas para ejecutar los cambios de duración y tono.

Se mencionó que las marcas de tono se hallan mediante un algoritmo de detección. Dichas marcas se pueden representar como una secuencia de análisis $T_a = \{t^a_1, t^a_2, \dots, t^a_M\}$, el periodo local entre dos de éstas marcas se define como:

$$P^a_m = \frac{t^a_{m+1} - t^a_{m-1}}{2} \quad (1.3)$$

Que no es más que un valor medio entre la marca de tono inicial y la marca de tono final. De este punto, se hace un ventaneo de la señal para separar en frames, éste se define como:

$$x^a_m[n] = W_m[n]x[n] \quad (1.4)$$

De aquí es necesario crear una secuencia de síntesis de las marcas de tono que depende de la duración y cambio en el tono deseados $T_s = \{t^s_1, t^s_2, .., t^s_M\}$, la relación de ésta secuencia de síntesis con la de análisis está relacionada por una función $M[i]$ que especifica cuáles frames de análisis deberán corresponder en la síntesis. Esta función es una suerte de línea de tiempo virtual entre síntesis y análisis, tal como se ve en la figura 1.5.

En el caso de la modificación del tono, tal y como se dijo en la sección dos, hay modificaciones en la duración de los periodos entre las marcas de tono. La consecuencia de esto puede ser la obtención de una señal más corta en el tiempo. Por esta razón, a veces es necesario duplicar algunos frames en afán de preservar la duración original de la señal.

Detallaremos esta explicación con un ejemplo:

- Se tiene un tono de 100Hz ventaneado con 5 frames cada uno con una separación de 10 ms. Es decir, hay una duración total de (5-1) *10 ms= 40ms entre la primera y la última marca de tono.
- Si deseamos cambiar el tono a 150 Hz es necesario poner la distancia de las marcas a 6.6 ms, el probema es que ahora nuestra duración total es de (5-1) *6.6ms= 26 ms.
- Para preservar la duración original, tenemos que duplicar dos frames, de esta forma volvemos a nuestra duración de 40 ms (7-1) * 6.6 ms =40 ms.

1.4.10 Multi-Band Resynthesis Overlap Add (MBROLA)

Se habló en párrafos anteriores acerca de dos tareas principales a resolver en la síntesis concatenativa, la primera tiene que ver con la modificación de la prosodia y la segunda con hacer una transición sutil entre fonemas.

El que la transición no sea sutil tiene que ver con una unión incorrecta entre fonemas, la cual puede ser de tres tipos:

Mala unión de Fase (Phase Mismatch): Este tipo de problemas ocurren cuando las formas de onda no están centradas en las mismas posiciones relativas dentro del periodo de tiempo en que se encuentran.

Mala unión de Tono (Pitch Mismatch): Sucede cuando ambos segmentos tienen la misma envolvente espectral, pero fueron pronunciados con diferentes tonos.

Mala unión de Envolvente de Espectro (Spectral Envelope Mismatch): Esta falla resulta cuando las unidades fonéticas fueron extraídas de contextos diferentes entre sí. La discontinuidad ocurre sólo en un período.

Ante estos problemas de unión, Dutoit y Leich (T Dutoit & Leich, 1993) proponen una solución conocida como MBROLA. Este algoritmo deriva directamente del TD-PSOLA, de hecho, es muy semejante. La diferencia radica en que no se hace un análisis individual de las ventanas. Ni son necesarias las marcas de tono.

Como lo muestra el diagrama, el sistema toma como referencia un difonema procedente de un corpus. El primer paso es diferenciar si es vocal o sordo. Si se trata de un sonido vocal, entonces se separa y se hace un análisis de bandas de este. El análisis se lleva a cabo mediante un sintetizador armónico que se encarga de calcular nuevas amplitudes y fases con características regulares. Estos difonemas re-sintetizados son después concatenados utilizando el método Overlap Add OLA.

Dado que objetivo de MBROLA es hacer las formas de onda lo más semejantes entre sí. Es esencial el reajuste de fases del que se habló anteriormente y se explicará a continuación con mayor detalle. El ajuste de las ondas se hace en los bordes de la última parte del primer segmento y de la primera parte del segundo segmento. El último borde y el subsiguiente se denotan como S_N^L y S_0^R respectivamente y los ajustes a los mismos se definen como M_L y M_R los cuales se obtienen de las siguientes fórmulas:

$$S_{N-1}^L = S_{N-1}^L + (S_0^R - S_N^L)\frac{1}{2}\left(\frac{M_L - i - 0.5}{M_L}\right) \quad (1.5)$$

$$S_j^R = S_j^R + (S_N^L - S_0^R)\frac{1}{2}\left(\frac{M_R - i - 0.5}{M_R}\right) \quad (1.6)$$

Para $i=0...M_{L-1}$ y $j=0...M_{R-1}$

Para la solución de la mala unión de la envolvente de espectro se usa el algoritmo propuesto por Charpentier y Moulines (Moulines & Charpentier, 1990) el cual consiste en la interpolación de los periodos vocales de tono regular (voiced pitch periods).

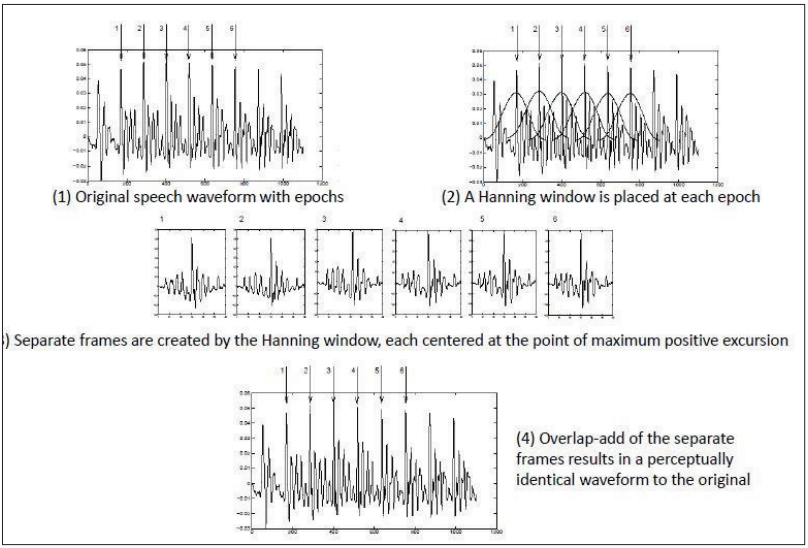
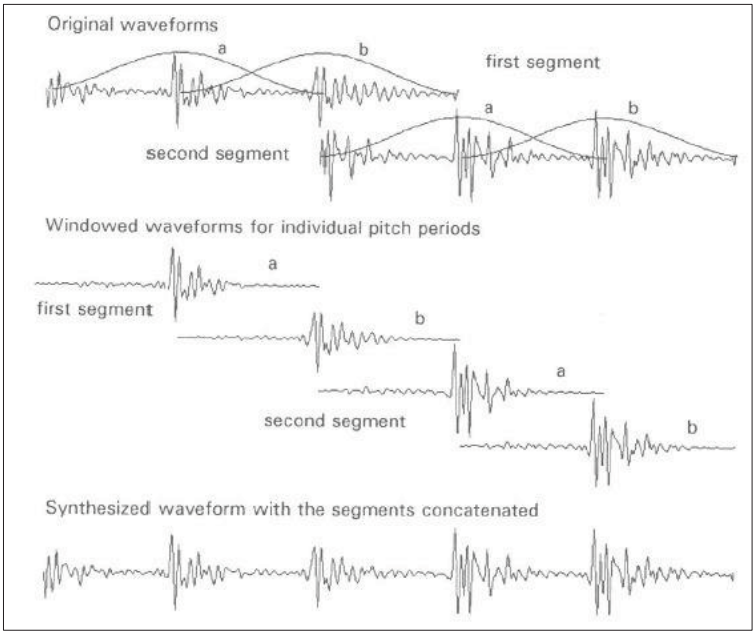


Figura 1.2 “(1) Detección de pitch marks. (2) Aplicación de ventanas Hanning. (3) Separación en frames. (4)



Reconstrucción de la señal original.”

Figura 1.3 “Traslape de segmentos”

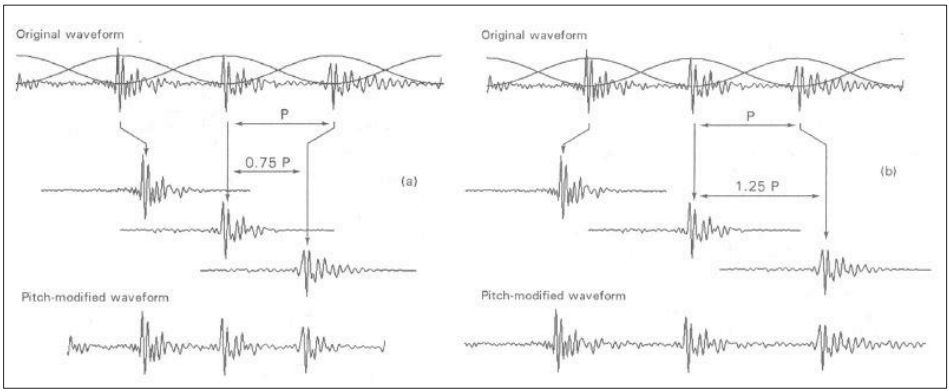


Figura 1.4 “Traslape para modificar tono”

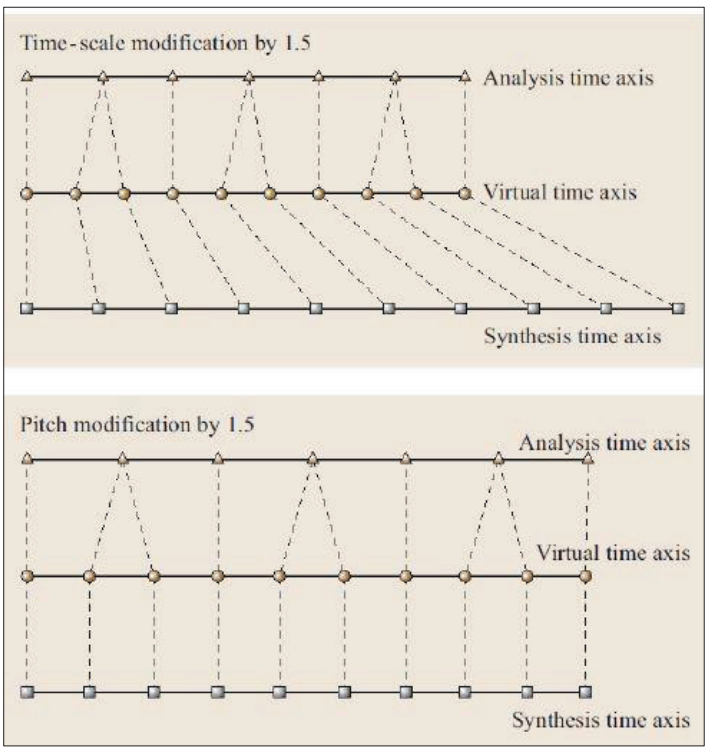


Figura 1.5 “línea de virtual tiempo de pitch marks entre análisis y síntesis”

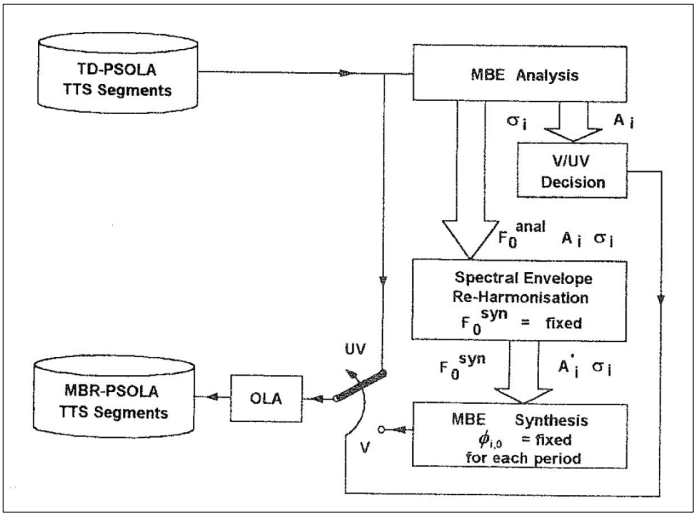


Figura 1.6 “Esquema MBROLA”

1.5 Sistemas referentes en el presente trabajo

A continuación, haremos un repaso de los sistemas que sirvieron como base para el sistema de síntesis de este trabajo. Se hace una descripción general ya que se profundizará en ellos en capítulos posteriores.

1.5.1 Mel Frequency Cepstral Coefficients MFCC

Los coeficientes obtenidos a partir de un proceso de filtrado conocido como Mel-Cepstral, son un conjunto de valores numéricos que resumen la información básica de las características que constituyen una señal de voz (Holmes & Holmes, 2001). El procedimiento para obtenerlos está basado en dos conceptos: El rango de frecuencias Mel y la separación de frecuencias por medio de Cepstrum.

El rango de frecuencias Mel está basado en la reducción de frecuencias de la señal de voz teniendo como referencia el rango auditivo humano, es decir, aquellas frecuencias que se pueden percibir más fácilmente. Por otro lado, *Cepstrum* es un concepto matemático que separa de la señal de voz en dos

bandas de frecuencias baja y alta. La baja corresponde a los formantes de los fonemas producidos debido a las cavidades del tracto vocal y la banda alta es relativa a la excitación en las cuerdas vocales. Esta última es una señal periódica muy particular a los distintos fonemas independientemente de las variaciones en el tracto vocal.

El algoritmo de MFCC se puede resumir de acuerdo con el diagrama siguiente:

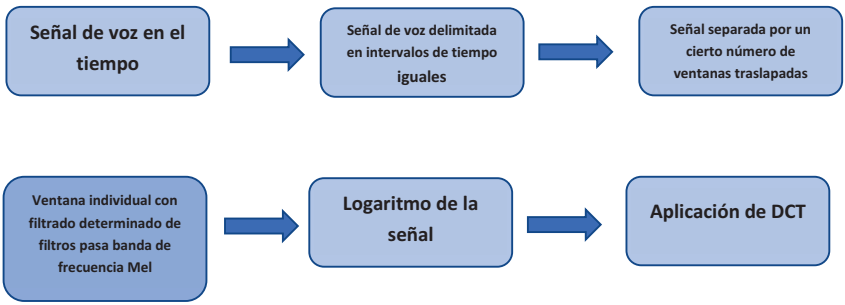


Figura 1.7 “Algoritmo MFCC”

Una señal de voz se divide en intervalos iguales de tiempo y posteriormente se hace un ventaneo traslapado de la misma. Posteriormente en cada una de las ventanas se aplica un conjunto de filtros pasa banda cuyo número varía de acuerdo con la precisión deseada. Al resultado de la señal filtrada en cada uno de los filtros es después una función logarítmica. Es a nueva señal de acuerdo al concepto de Cepstrum es necesario volver a aplicar una FFT la cual, debido a su simetría, se obtiene mediante una transformada coseno discreta.

1.5.2 Algoritmo detallado

A continuación, hacemos una descripción a detalle de la obtención de los coeficientes MFCC. La figura 1.7 muestra el sistema en esquema.

1. Se hace preénfasis a la señal de voz, es decir se amplifican las altas frecuencias para facilitar el cálculo de las formantes con amplio contenido en el espectro alto.
2. Se aplica una ventana Hamming para obtener la frecuencia promedio en diferentes tramas o *frames.* Generalmente se aplica una ventana de 20 ms a intervalos de 10 ms.

3. Se obtiene la DFT de cada frame.
4. Se aplica un banco de filtros a cada frame. De acuerdo con Davis y Mermelstein (Davis & Mermelstein, 1978) los filtros se distribuyen de manera no lineal de acuerdo a la escala Mel. Normalmente se utilizan 20 filtros. Los primeros 10 están linealmente distribuidos y los siguientes 10 crecen en forma logarítmica.
5. Se aplica la transformada Coseno Discreta, la cual es una variante de la FFT a la salida de cada filtro. Normalmente se obtienen de 10 a 12 coeficientes MFCC, pero el número es modificable por el usuario.

Los MFCC son una manera compacta de almacenar sonido. No son otra cosa más que números que revelan las diferentes amplitudes de la señal, pero no contienen en sí mismos energía acústica codificada.

Si se van a utilizar para hacer síntesis, hacen la función de un filtro a través del cual pasa una fuente sonora dual que emite una señal sinusoidal para sonidos vocales y una señal de ruido blanco para sonidos sordos.

1.5.3 Line Spectral Pair LSP

Line Spectral Pair (Itakura & Sugamura, 1979) es un método de parametrización, o cuantización de una señal de voz que parte del ya mencionado **Linear Predictive Coding**. Se genera a partir de la ecuación (1.7) del filtro A(z) que representa el tracto vocal.

$$A_p(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p} \quad (1.7)$$

Se plantea que en el polinomio de los coeficientes del filtro se agregan un par de elementos P(z) y Q(z) que representan la glotis en el momento de abrirse y de cerrarse respectivamente. De ahí que uno lleve signo positivo y otro negativo, se representa como (1.8) y (1.9)

$$P(z) = A_p(z) - z^{-(p+1)} A_p(z^{-1}) \quad (1.8)$$

$$Q(z) = A_p(z) + z^{-(p+1)} A_p(z^{-1}) \quad (1.9)$$

Donde P(z) y Q(z) se relacionan con (1.7) de la siguiente forma:

$$A(z) = \frac{P(z)+Q(z)}{2} \quad (1.10)$$

En la práctica, la glotis nunca está totalmente cerrada ni totalmente abierta (McLoughlin, 2008). Esto significa que los polinomios añadidos son al final de cuentas más elementos para cuantizar nuestra señal de voz, consiguiendo darle más naturalidad que cuando se limita a la representación con coeficientes LPC. Las ecuaciones (1.8) y (1.9) no son más que el filtro A(z) sumado a sí mismo pero desplazado en el tiempo.

En el capítulo siguiente se hablará con detalle de este método ya que fue el que se eligió para la parametrización de voz de nuestro sistema.

1.5.4 HTS

La síntesis HTS (*Hidden Markov Models as Text to Speech Sythesis*) es una propuesta de principios de siglo 21. Funciona a partir de un proceso de selección de unidades a partir de frases completas parametrizadas por sus MFCCs. (K Tokuda, Yoshimura, Masuko, Kobayashi, & Kitamura, 2000; K Tokuda, Zen, & Black, 2002; Keiichi Tokuda et al., 2013). La selección se lleva a cabo mediante un algoritmo estadístico que define la aparición de cada unidad con respecto a probabilidades. Para hacer el cálculo de probabilidades, los creadores de HTS se basaron en un software de la Universidad de Cambridge originalmente diseñado para reconocimiento de voz. El programa en cuestión es el HTK (Hidden Markov Model Tool Kit) propuesto por (Young, 2013).

La síntesis de voz en el sistema HTS se logra a través de un sistema de filtrado basado en el ya mencionado ***Vocoder.*** Como se puede ver en la figura 8, el Vocoder tiene como señal de entrada una fuente sonora la cual tiene dos tipos de sonidos: Sonidos vocales *voiced sounds* o sonidos sordos (no vocales) *unvoiced sounds*. Los primeros emulan a aquellos elementos de la voz humana que surgen a partir de la vibración de las cuerdas vocales. Generalmente se producen a partir de una señal sinusoidal. Los sonidos no vocales o sordos representan aquellos fonemas que surgen al pasar una

corriente de aire a través del tracto vocal, por ejemplo, en los fonemas /f/ o /s/. Este tipo de sonidos se modelan con una fuente de ruido blanco.

Ambas señales fuentes sonoras pasan a través de un filtro pasa banda. Los parámetros de sintonización del filtro se establecen mediante los coeficientes Mel-Cepstral (descritos en sección 1.5.1), los cuales llevan codificada la energía del espectro de frecuencias de los fonemas que se van a producir. Finalmente, en la salida del filtro tenemos la señal de voz deseada. Una analogía para representar el sistema es el de la fabricación de galletas, nuestra fuente de voz sería la masa y el sistema de filtrado es el molde que les da la forma.

Los elementos de entrada en el sistema de síntesis de voz de HTS, van almacenados en forma de datos en el **vector de observación** el cual normalmente contiene los datos pertenecientes a una trama. Los datos que Tokuda reporta en sus diferentes escritos que se utilizan en cada frame son: los Coeficientes Mel-Cepstral, los valores de excitación de F0 y sus equivalentes dinámicos delta y delta-delta. El diagrama de funcionamiento de síntesis por HTS es el siguiente:

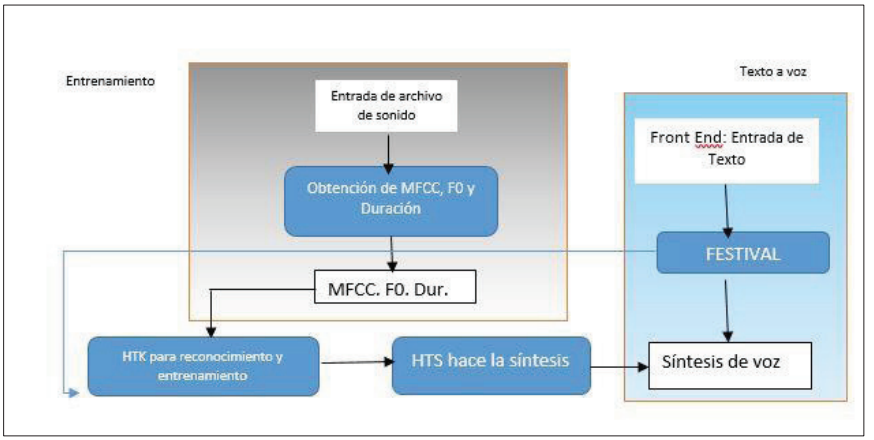


Figura 1.8 “Esquema General del sistema HTS”

De este sistema se habla detalladamente en el capítulo 3, ya que fue nuestra referencia principal como sistema de síntesis.

1.6 Conversión Texto a Fonemas

Se han mencionado ya los diferentes modelos de síntesis de voz. El reto que se enfrenta hoy en el desarrollo de síntesis de voz no es únicamente la forma de emular la voz humana, sino también encontrar un sistema de control eficiente para producirla.

Los tres métodos de síntesis aquí mencionados resultan complicados de manipular por una misma razón: Los múltiples parámetros que implican modificarse para producir una frase.

Los sistemas de cómputo actuales han facilitado este control multi-parámetro, gracias a la rapidez de los procesadores se han podido programar los diferentes parámetros y ejecutar en fracciones de segundo. Esto desafortunadamente sólo ha solucionado parte del problema ya que los investigadores en tecnologías del habla han descubierto que el lenguaje hablado es mucho más complicado de recrear de lo que parece, no sólo por la emulación de los fonemas sino por la articulación de las palabras.

El método tradicional para generar una frase sintetizada es teniendo la frase que se desea producir como texto a manera de entrada denominado *Text-to-speech*, desde luego los fonemas (sonido de las palabras) no necesariamente coincide siempre con los grafemas (letras), por ello es necesario un proceso previo de interpretación de texto. El proceso consiste en una serie de reglas por lo que se conoce como *synthesis by rules.*

A continuación, se presenta la explicación de esta etapa en la síntesis de texto tomado de las notas de Herrera (Herrera-Camacho & Ávila, 2013). Se hará mención de la terminología utilizada en Festival porque fue el sintetizador que se estudió (Taylor, Black, & Caley, 1998; K Tokuda et al., 2002) debido a ser uno de los mejores en su clase y que en él están basados los actuales sistemas de síntesis que se estudiaron.

En la figura 12 se muestra un diagrama de bloques de las varias etapas en un sistema texto a voz concatenado. La entrada del sistema es un texto sin restricciones en forma de una secuencia de caracteres, incluyendo números, abreviaciones y signos de puntuación. La función del normalizador de texto es procesar cualquier carácter no alfabético: los signos de puntuación que se identifiquen se dejarán en su lugar; las abreviaciones serán expandidas a su

forma completa; las cantidades se expandirán en sus formas completas también, por ejemplo “£2.75” se convertirá en “dos libras y setenta y cinco centavos”. Esta etapa se conoce en Festival como tokenización. Normalmente hay una única posibilidad de token por grafema, sin embargo, en el caso de los números o determinados signos de puntuación, las posibilidades aumentan considerablemente.

La salida del normalizador de texto es texto plano en forma de una secuencia de caracteres alfabéticos y signos de puntuación. Aquí se fonetizan todos los grafemas encontrados, por ejemplo, “casa” se convierte en “kasa”, “queso se vuelve “keso”, “hola” se modifica a “ola”, etc. En Festival se denomina como *lexicon* a los caracteres que denotan la sonoridad del fonema en cuestión. Por ejemplo: “photography” es en lexicon, (((f@)0)((tog)1)((r@f)0)((ii)0))).

El siguiente módulo llamado analizador de sintaxis/prosodia usa un algoritmo de análisis para segmentar el texto de tal forma que se le pueda asignar una entonación y ritmo significativos. Esto normalmente involucra un análisis gramatical, esto es, la identificación de sustantivos, verbos, preposiciones, conjunciones, etc. El módulo asigna marcadores al texto, los cuales indican, por ejemplo, las sílabas acentuadas, los puntos de acentuación tónica en un patrón de entonación y los tipos de patrones de entonación a ser usados en varias partes de la locución.

Es bien sabido en el campo de la lingüística que los fonemas modifican sus sonidos dependiendo del fonema que lo antecede y del que lo precede. Por esta razón los sistemas de texto a voz necesitan puntos de comparación para saber cuál es la mejor opción de fonema a sintetizar. De ahí la importancia de dotar al sistema de una base de datos o corpus que contenga diferentes opciones de fonemas. Dentro de la base de datos, cada fonema viene etiquetado con su probabilidad de ocurrencia.

La forma de calcular la probabilidad máxima de ocurrencia se hace mediante la resolución de árboles determinísticos. Normalmente los pasos a seguir son los siguientes:

- Pre-procesar el lexicon en texto funcional a un sistema de entrenamiento
- Definir un conjunto de equivalencias pares grafema-fonema
- Construir las posibilidades de cada par grafema-fonema

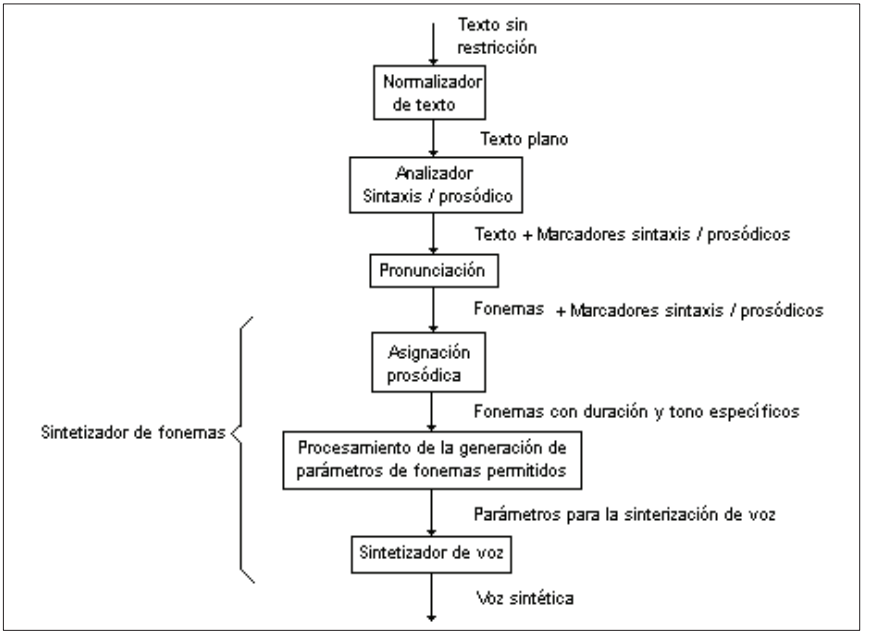


Figura 1.8 "Diagrama de bloques de un sintetizador concatenado"

- Construir modelos CART para predicción de fonemas desde grafemas
- Ir obteniendo los difonemas correspondientes y concatenándolos uno tras otro.

Se denomina CART (Clasiffication and Regression Tree) al sistema probabilístico de extracción de datos que se aplica en este proceso de selección. Un ejemplo del árbol de clasificación y regreso aplicado a Festival es el siguiente:

- Se tiene como texto de entrada la palabra Queso, la cual se fonetiza como /K//E//s//o/.
- Se revisa cada token (grafema) de forma individual y se hace una pregunta, es decir: Fonema /k/ ¿viene consonante o vocal? Respuesta: Vocal. ¿Esta vocal es débil o fuerte? Respuesta: Débil. ¿La siguiente letra es consonante o vocal? Respuesta: Consonante.
- El sistema determina un 80% de probabilidad que el siguiente fonema sea /E/

Las iteraciones necesarias se realizan hasta completar el texto presentado como entrada al mismo tiempo que el programa va concatenando los diferentes difonemas que forman parte del corpus. En su esquema más básico, el programa es limitado en cuanto a modificaciones en la prosodia del texto sintetizado.

1.6.1 Selección de unidades

Anteriormente expuesto en este documento, se tiene ya mencionado que en la síntesis concatenativa se parte de fragmentos de voz previamente grabados por un locutor. A partir de estos fragmentos de voz es como se van a reconstruir diferentes palabras.

Se denomina síntesis de voz por unidades (Thierry Dutoit, 2008) a aquel tipo de síntesis, donde las frases sintetizadas son logradas a través de la concatenación de palabras completas extraídas de una base de un corpus de frases pre-grabadas. A últimos años, los especialistas en síntesis de voz prefieren utilizar este sistema de selección unidades sobre otros, como el de fonemas o difonemas, ya que al trabajar con palabras o frases completas es posible mantener una mejor inteligibilidad y naturalidad en cada frase. Las distintas unidades de voz tienen un sistema de etiquetado que permite después ubicarlas como vectores de observación (K Tokuda et al., 2000, 2002) que son estados dentro del sistema de selección por modelos ocultos de Markov (HMM) -del que se hablará más adelante en el texto-. Otra manera de hacer la selección de unidades es por medio de un algoritmo estadístico de conjuntos de unidades con elementos comunes, de aquí se desprenden dos métodos propuestos por Alan Black: *Clustering* (Black & Taylor, 1997) y *CLUSTERGEN* (Black, 2006). Ambos métodos son la base de selección del conocido sistema de síntesis de voz FESTIVAL, desarrollado en conjunto por CMU y la Universidad de Edinburgo. Ejemplos de sonido de este sistema se puede escuchar en los audios *fest_clunits_esp.wav* y *fest_multisyn.wav*.

Con el paso del tiempo, la selección de unidades utilizando HMM ha demostrado ser mucho más eficiente que los métodos basados en clusters por lo que incluso FESTIVAL la ha adoptado. Por esta razón no se hablará con detalle en el texto de los sistemas Clustering y CLUSTERGEN. Si el lector desea profundizar en estos sistemas, puede encontrar información relevante en la página de www.festvox.org

Capítulo 2

El presente capítulo pretende dar al lector un panorama general del concepto Par lineal espectral *Line Spectral Pair* LSP. Se muestran algunos antecedentes en su aplicación, así como también la forma en que lo utilizamos en el presente trabajo.

2.1 Justificación en el uso de LSP

Mucho se ha probado y documentado en lo referente a síntesis de voz utilizando MFCC. LPC por su parte ha quedado relegado al ámbito de reconocimiento de voz debido a que en síntesis produce una voz en exceso artificial. Sin embargo, de LPC se han realizado parametrizaciones variaciones, destacando el Par Lineal Espectral *Line Spectral Pair* (McLoughlin, 2008) el cual se ha aplicado a reconocimiento con cierto éxito y se tiene documentado un primer intento en síntesis (Chennoukh et al., 2001; Franco, Herrera, et al., 2016; Nakatani et al., 2006; Soong & Juang, 1984). Destaca el trabajo de Naktani y colegas, debido a que luego de hacer algunas pruebas, notaron que las formantes de los sonidos vocales son menos planas parametrizadas con LSP que con MFCC.

Además de reconocimiento y síntesis, LSP se ha usado en ciertas variantes de modificación de voz usando STRAIGHT (*Speech Transformation and representation using adaptive interpolation of weighted spectrum*) como lo hicieran (Arakawa et al., 2010; Kang & Liu, 2006). Por su parte (Sagayama & Itakura, 2002) proponen el uso de un modelo dual a LPC conocido como *Composite Sinusoidal Model* CSM donde también se obtienen las LSP.

MFCC si bien es eficiente, tampoco es óptimo en términos de naturalidad e inteligibilidad. Además de la poca precisión en su espectro, el cual cómo se menciona en (Nakatani et al., 2006) es mucho más plano que el espectro de los sonidos vocales originales. Por esta razón se decidió probar LSP como una alternativa de parametrización que pudiera en un momento dado mejorar lo conseguido con MFCC.

Se han hecho ya trabajos previos de síntesis de voz en español utilizando MFCC (Herrera-Camacho & Ávila, 2013) denominado HTS-MFCC pero no hay síntesis de voz en español utilizando LSP por lo que se buscó implementar y

documentar un sistema que la empleara (Franco, Herrera, et al., 2016) Al que se denomina HTS-LSP.

Defendemos también el uso de LSP porque al estar basado en LPC viene directamente de un modelado físico del tracto vocal visto como filtro. Por otro lado, se puede revertir el proceso y hacer una reconstrucción de la señal de voz original, a diferencia de MFCC donde esto no es posible. Finalmente, los archivos generados con LSP son de menor tamaño, lo cual implica una economía de recursos computacionales en procesamiento y almacenamiento.

2.2 Trabajo Relacionado

La parametrización de voz con LSP ha sido tema de interés en sistemas de síntesis y reconocimiento de voz en las tres últimas décadas. Los trabajos más relevantes a nuestra investigación fueron los de Nakatani, Arakawa, Tokuda y Bäckstörm. Nakatani (Nakatani et al., 2006) y sus colegas evaluaron frases usando parametrización Mel LSP pero sus estudio estuvo exclusivamente enfocado en el análisis de fonemas aislados del idioma japonés y no frases completas. Arakawa y sus colegas (Arakawa et al., 2010) utilizaron LSP para mejorar algunas características del sistema STRAIGHT, aunque los principios de dicho sistema difieren de aquellos con los que se rige el sistema del presente trabajo. Bäckstörm (Backstrom, 2004) en su proyecto doctoral hace un detallado análisis matemático de LSP, su trabajo es muy amplio y no se centra únicamente en señales de voz. Tokuda y su equipo (Tokuda et al., 2013) dejaron la puerta abierta para experimentar ya fuera con parametrizaciones LSP o MFCC pero su enfoque es en HTS desde una perspectiva global y no reportan resultados en cuál de las parametrizaciones resulta más efectiva.

2.3 Teoría Básica de LSP

Line Spectral Pair (Itakura & Sugamura, 1979) es un método de parametrización, o cuantización de una señal de voz que parte del ya mencionado **Linear Predictive Coding**. Se genera a partir de la ecuación (2.1) del filtro A(z) que representa el tracto vocal.

$$A_p(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p} \text{ (2.1)}$$

Se plantea que en el polinomio de los coeficientes del filtro se agregan un par de elementos P(z) y Q(z) que representan la glotis en el momento de abrirse y de cerrarse respectivamente. De ahí que uno lleve signo positivo y otro negativo, se representa como (2.2) y (2.3). Si observamos las ecuaciones podemos inferir que P(z) y Q(z) es la resta y suma respectivamente del filtro A(z) consigo mismo pero desplazado en el tiempo.

$$P(z) = A_p(z) - z^{-(p+1)} A_p(z^{-1}) \text{ (2.2)}$$

$$Q(z) = A_p(z) + z^{-(p+1)} A_p(z^{-1}) \text{ (2.3)}$$

Donde P(z) y Q(z) se relacionan con (2.1) de la siguiente forma:

$$A(z) = \frac{P(z)+Q(z)}{2} \text{ (2.4)}$$

En la práctica, la glotis nunca está totalmente cerrada ni totalmente abierta (McLoughlin, 2008). Con ello se garantiza que no se dará el caso en que el filtro se anule.

Otra ventaja que tiene este sistema de parametrización es que las raíces del polinomio (2.1) corresponden específicamente a las frecuencias formantes de la señal de voz parametrizada. A partir de ahí podemos llevar a cabo reconocimiento y/o síntesis de voz. A este conjunto de frecuencias obtenidas se le conoce como *Line Spectral Frequencies o* LSF. Los polinomios P(z) y Q(z) se pueden expresar también en términos de sus frecuencias (Kabal & Ramachandran, 1986) de la siguiente forma:

$$P(z) = (1 - z^{-1}) \prod_{i=2,4,\ldots,M} (1 - 2z^{-1} cos\omega_i + z^{-2}) \text{ (2.5)}$$

$$Q(z) = (1 + z^{-1}) \prod_{i=1,3,\ldots,M} (1 - 2z^{-1} cos\omega_i + z^{-2}) \text{ (2.6)}$$

El rango de frecuencias va de 0 a π radianes y P(z) contiene a los coeficientes pares mientras Q(z) los impares.

La figura 2.1 Muestra gráficamente la comparación entre LPC y LSP. La gráfica azul está formada a partir de los valores en LPC y los puntos rojos y verdes corresponden a P(z) y Q(z) respectivamente. Ahí están indicadas las frecuencias correspondientes a las resonancias en el tracto vocal. Al observar la gráfica el lector podrá corroborar que P(z) y Q(z) tienen una correspondencia directa con las frecuencias de la señal LPC.

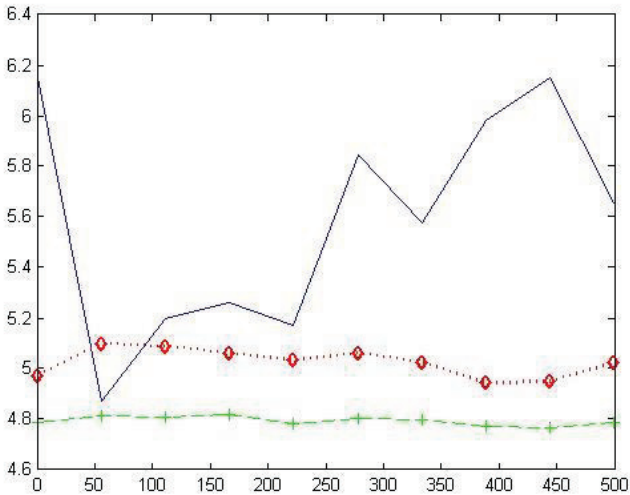


Figura 2.1 Representación gráfica de LSP contra LPC

Las raíces α_i y β_i de los polinomios P(z) y Q(z) tienen una serie de propiedades importantes (Bäckström & Magi, 2006) destacando las siguientes :

1. α_i y β_i están en el círculo unitario $|\alpha_i| = |\beta_i| = 1$ y pueden ser presentadas como $\propto_i = e^{i\pi\lambda_i}$ y $\beta_i = e^{i\pi\gamma_i}$
2. λ_i y γ_i son distintos y $\lambda_i \neq \lambda_j$, $\gamma_i \neq \gamma_j$, para $\gamma_i \neq \lambda_j$ y $i \neq j$.
3. λ_i y γ_i están entrelazadas y $\gamma_i < \lambda_i < \gamma_{i+1}$

Ya que las raíces están en el círculo unitario, tienen frecuencias correspondientes a sus respectivos ángulos, estas se conocen como Line Spectral Frequencies LSF.

2.4 LSP aplicado a una señal de voz

El proceso para extraer el LSP de una señal LPC ha sido implementado en diferentes tipos de hardware y software. Nuestra referencia fue el código desarrollado por (Rabiner, 2015) al cual se le hicieron algunos ajustes. La señal de voz convertida a LCP fue producto del software del Laboratorio de Tecnologías del Lenguaje desarrollado por Carlos Acosta.

El código usado para los LSP se encuentra en libremente en internet. La figura 2 muestra los elementos que fueron añadidos al mismo.

```
%function    [P,PF,Q,QF]=atolsp(A,fs)
clc;
clear all
close all

load('Dprim.mat'); x=Dprim'; fs=44100;
[L M]=size(x);
Pout=zeros(L,M);
Qout=zeros(L,M);
PFout=zeros(L,M);
QFout=zeros(L,M);
inicio=1;
iniciocolum=1;

while (inicio<L)&& (iniciocolum<M)
    [Pout(inicio:14,iniciocolum) PFout(inicio:13,iniciocolum) Qout(inicio:14,iniciocolum)...
        QFout(inicio:13,iniciocolum)]=atolsp(x(inicio:L,iniciocolum),fs);
inicio=1;
iniciocolum=iniciocolum+1;
end

pfabs=abs(Pout(1:13,1));
qfabs=abs(Qout(1:13,1));
figure; plot (abs(Dprim(1,1:13))); %hold 'on' ; stem (PFout(1,2:13),'r');
%hold 'on' ; stem (QFout(1,2:13),'g');
figure; plot(abs(Dprim(1,1:13))); hold 'on';
stem((pfabs),'r');
hold 'on'; stem((qfabs),'g+','--');
```

Figura 2.2 “Cambios en el código de LSP”

Las frecuencias LSP servirán como entrada al filtro pasabanda HTS Engine (HTS, 2015) que forma parte del sintetizador en HTS. La voz será recreada de acuerdo con los valores de las LSF. Tales frecuencias, reiteramos, corresponden a las formantes de la señal de voz.

Capítulo 3

En este capítulo se hará una descripción detallada del sistema HTS explicando componentes y su funcionamiento. Se darán nociones básicas del concepto matemático de los Modelos Ocultos de Markov HMM. Finalmente se hablará de los coeficientes Mel General Cepstral y cómo de ellos se deriva el Par Lineal Espectral.

3.1 Sistema HTS, antecedentes

El sistema HTS *Hidden Markov Models as Text to Speech Synthesis* es una propuesta de la primera década del 2000 (Keiichi Tokuda et al., 2013). HTS corresponde a la clase de sintetizadores de voz concatenativos, de los que se habló en el capítulo 1. En estos sintetizadores, la frase se genera al unir fonemas y difonemas.

Originalmente, en los sintetizadores concatenativos, los fonemas y difonemas eran escogidos a partir de grabaciones de señales de voz. Esto generaba dos inconvenientes: Una base de datos de varios Megabytes y poca naturalidad, ya que podían escucharse las transiciones entre uno y otro de los difonemas. Por esta razón, se elige simplificar la voz a su mínima expresión mediante un proceso de parametrización. La voz parametrizada es un mapeo en un espectro frecuencia-amplitud de las frecuencias formantes del fonema que se desea reproducir. Existen diferentes tipos de parametrización de voz como DWPT *Discrete Eavelet Packet Transform,* AF *Articulatory Features* y MFCC *Mel Frequency Cepstral Coefficients.* Detalles y reseña histórica sobre DWPT y AF se pueden encontrar en (Ganchev, 2011). Agregamos la parametrización LSP que utilizamos en el presente trabajo y de la que se habló en el capítulo 2.

La parametrización de voz, cualquiera que ésta sea, no contiene energía acústica en sí misma. Su utilización primaria fue para reconocimiento de hablante, en ese caso, no era necesaria. En síntesis de voz, como se dijo en el capítulo 1, la parametrización funciona para configurar las frecuencias formantes en un filtro a pasabanda, llamado Vocoder, del inglés *Voice Decoder* o decodificador de voz, través del cual pasará una señal sinusoidal para generar fonemas vocales y una fuente de ruido blanco para generar fonemas

sordos. Los primeros emulan a aquellos elementos de la voz humana que surgen a partir de la vibración de las cuerdas vocales. Generalmente se producen a partir de una señal sinusoidal. Los sonidos sordos representan aquellos fonemas que surgen al pasar una corriente de aire a través del tracto vocal, por ejemplo, en los fonemas /f/ o /s/.

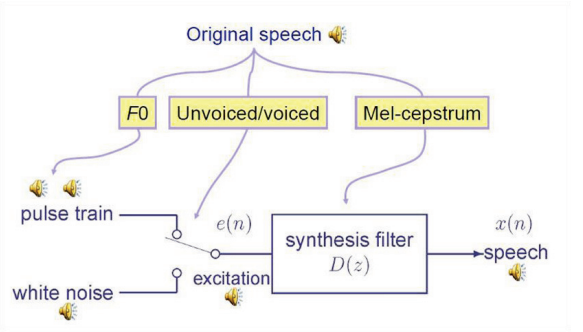


Figura 3.1 "Esquema del Vocoder en HTS"

3.2 Funcionamiento general de HTS

Cómo se puede ver en la figura 3.1, en el caso particular de HTS, la parametrización de voz que entra al Vocoder consta de tres vectores de datos: Coeficientes Generales Cepstral MGC, la frecuencia fundamental f0 y la duración (K Tokuda et al., 2002). Sobre MGC se hablará más adelante en el documento ya que es parte medular de la parametrización LSP.

La figura 3.2 muestra un diagrama a bloques del sistema HTS, arriba por la derecha se muestra la señal de voz como entrada al bloque denominado SPTK. En ese bloque se utiliza un programa desarrollado exclusivamente para procesamiento digital de voz llamado *Speech Processing Toolkit* (Sptk Manual, 2013). En este software se lleva a cabo la separación de la señal de voz en los tres vectores de datos que se mencionaron anteriormente: MGC, f0 y duración.

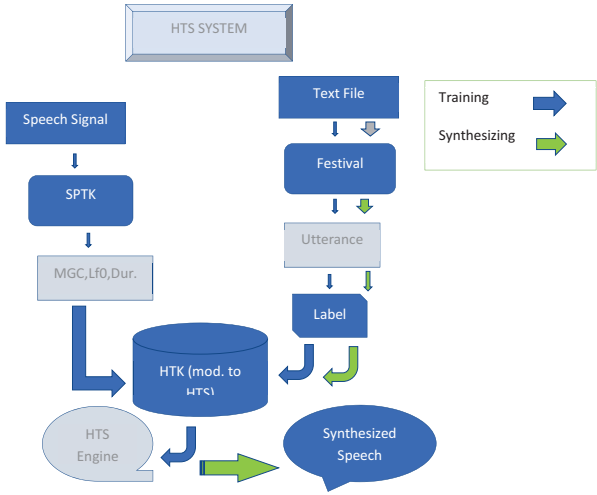


Figura 3.2 “Esquema General de HTS”

Los tres vectores ingresan como Modelos Ocultos de Markov a un segundo software conocido como HTK *Hidden Markov Models Toolkit* el cual fue creado en la Universidad de Cambridge (Young, 2013) por Steve Young. Se utiliza para llevar los cálculos probabilísticos de los modelos ocultos de Markov. Del principio sobre el cual operan tales modelos se detalla en otra sección del presente capítulo.

Bajo el bloque HTK vemos que la señal se dirige a otro bloque nombrado *HTS Engine*. Así se denomina el software que lleva a cabo la función del Vocoder mencionado anteriormente en el texto.

Si miramos la figura en la parte izquierda, veremos que simultáneamente a la señal de voz, ingresa al sistema un archivo de texto. Tal archivo contiene justamente la frase que se desea sintetizar. El archivo de texto es procesado en Festival (CMU, 2016) el software de síntesis de voz que mencionamos en el capítulo 1 y que se utiliza en la etapa de procesamiento de texto. En este caso descompone el texto en fonemas y lo reordena en un archivo denominado *utterance*. Es justamente este archivo el que sirve de entrada al vocoder para especificar que frecuencias corresponden a que fonema y así llevar a cabo la síntesis.

Para que el archivo *utterance* sea compatible con el arriba mencionado software de síntesis, es necesario hacerle ciertas modificaciones requeridas por el vocoder HTS-Engine. Una vez realizadas es renombrado *label*.

3.3 Los Modelos Ocultos de Markov HMM

La explicación de la sección 3.2 pretende dar al lector un panorama general en el camino que siguen los datos para llegar a la síntesis de voz. Sin embargo, antes de hacer una explicación detallada de lo que ocurre en cada una de las etapas, es importante abordar la teoría de lo que sucede para elegir correctamente la secuencia de fonemas de la frase deseada. Como se dijo anteriormente las frases descompuestas en coeficientes MGC, f0 y duración, entran como modelos ocultos de Markov, en adelante denominados HMM a un software específicamente diseñado para su cálculo que es HTK.

Bien merece la pena que el lector conozca un poco la teoría de los HMM y cómo operan en la práctica. Un modelo oculto de Markov es un conjunto de S_i (donde i= 1, …, N) estados. A cada estado corresponde un conjunto de probabilidades de transición a_{ij} donde $\sum_{j=1}^{N} a_{ij} = 1$. Cada estado tiene al mismo tiempo un conjunto de probabilidades b_j donde $\sum_{j=1}^{N} b_j = 1$. La observación de cada estado tiene lugar en un tiempo t_i. La figura ilustra un modelo con tres estados, es decir N=3.

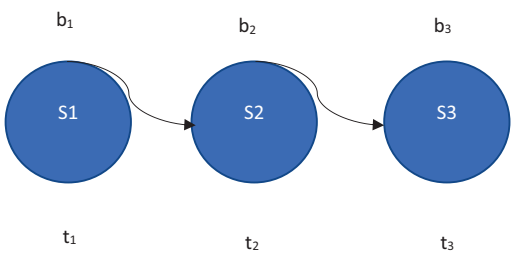


Figura 3.3 “Ejemplo de HMM”

Para empezar a calcular las probabilidades a_{ij} y b_j Es necesario definir un vector de condiciones iniciales $\pi_i = 1, \dots, N$ en el instante t_1.

La forma general para calcular las probabilidades de un estado S_i dado un estado anterior S_{i-1} en un tiempo *t=i* es:

$$P(S_{i-1}|S_i) = \pi_i a_{i+1} b_i \ (3.1)$$

En un tiempo *t=i+1,*

$$P\left(S_{(i-1)+1}\middle|S_i\right) = \pi_{i+1} a_{i+2} b_{i+1} \ (3.2)$$

Los cálculos se repiten progresivamente hasta que el tiempo alcanza un valor *t=N.* Los resultados se almacenan en un vector $X(P_i, t)$. Para hallar la secuencia de estados más probable, se eligen los valores en el vector $X(P_i, t)$ de acuerdo a:

$$V_T = argmax[X(P_i, t_i)] \ (3.3)$$

El siguiente ejemplo buscará ilustrar el proceso. Imagine el lector que un sistema recibe la instrucción de sintetizar la frase "*El perro murió".* Es necesario calcular las probabilidades de la mejor combinación en una secuencia de estados. Cada palabra representa un estado Si (con i=1,2,3), cada estado corresponde a un elemento gramatical en la oración S_1 ***artículo***, S_2 ***sustantivo,*** S_3 ***verbo.*** Las probabilidades de observación definidas previamente son: OP_{art}=0.8, OP_{sust}=0.18, OP_{verbo}=0.02.

En la oración *El perro murió.* Es altamente probable que un artículo aparezca al inicio de la frase, seguido del sustantivo y terminando con el verbo. Por lo tanto, las probabilidades de transmisión del estado S_1 son: $P_{art\text{-}sust}$=0.8, $P_{art\text{-}verbo}$=0.2. Las probabilidades de transición del estado S_2 al estado S_3 son: $P_{sust\text{-}verbo}$=0.9, $P_{sust\text{-}art}$=0.1.

Definimos una condición inicial π=1. Los cálculos en tiempo t=1, aplicando la ecuación (1) son:

P_1(condición inicial|$P_{art\text{-}sust}$) =1*0.8*0.8=0.64

P_1(condición inicial|$P_{art\text{-}verbo}$) =1*0.8*0.2=0.16

Los cálculos de probabilidades en tiempo t=2:

P_2(condición inicial|$P_{sust\text{-}verbo}$) =0.64*0.19*0.9=0.1

P_2(condición inicial|$P_{sust\text{-}art}$) =0.16*0.18*0.1=0.0028

En tiempo t=3 termina el proceso. Los máximos valores para cada t se almacenan en el vector $X(P_i, t)$:

$$X(P_1, t_1) = (0.64, 0.16)$$
$$X(P_2, t_2) = (0.10, 0.0028)$$

La secuencia correcta de estados está dada por:

$$V_T = argmax[X(P_i, t_i)] \ (3.4)$$
$$V_{T1} = argmax[X(P_1, t_1)] = 0.64$$
$$V_{T2} = argmax[X(P_2, t_2)] = 0.10$$

Dicha secuencia corresponde correctamente al orden en que deben aparecer los estados artículo, sustantivo y verbo en la frase del ejemplo.

La ventaja de utilizar HMM en contraste con otros sistemas es que podemos acceder a la base de datos de fonemas de manera no lineal, a diferencia de otros métodos de selección lineal como es CART (Black & Taylor, 1997) en el programa Festival (Taylor et al., 1998). Originalmente, HMM se aplicó a reconocimiento de hablante o reconocimiento de texto. Con este objeto se creó la herramienta HTK que se utiliza en HTS.

3.4 Entrenamiento del Sistema

Antes de poder sintetizar una frase, el sistema debe ser entrenado con las especificaciones del idioma deseado. En esta etapa de entrenamiento se definen también otras características como son parametrización, número de coeficientes de esta, frecuencia de muestreo, entre muchas otras.

El sistema adaptado a español mexicano se entrenó utilizando 300 frases en español fonéticamente balanceadas. Las frases ingresan tanto como archivos de sonido (.wav) así como con su respectiva transcripción en archivo de texto.

La probabilidad más alta de ocurrencia de una secuencia de fonemas se calcula en HTK utilizando los HMM para obtener la mejor combinación.

La conversión de texto a fonemas se lleva a cabo dentro de Festival (Taylor et al., 1998). Dado que Festival fue originalmente diseñado para síntesis en idioma inglés, el sistema está adaptado a la gramática inglesa. Las características gramaticales del idioma van codificadas en un software llamado *lexicon.* Para adaptarlo a español es necesario generar un lexicon con la gramática española, donde se indican particularidades de español no existentes en el inglés. Ejemplos de esto pueden ser la acentuación de vocales, el uso de la letra “ñ”, diferencias de pronunciación entre fonemas /c/ o /z/, etc.

Desde anteriores adaptaciones al español mexicano, se hizo uso de un lexicon creado originalmente para español de Andalucía. No hay problema de aplicar esta elección ya que gramaticalmente el español ibérico es idéntico al mexicano. La única consideración a es elegir un fonema /s/ cuando en la escritura aparezcan letras “c” o “z”.

El análisis del texto para convertir a fonema en Festival se lleva a cabo en el siguiente orden: Enunciado a frase, frase a palabra, palabra a sílaba y sílaba a fonema (Black, 2006). Una vez realizada la conversión, Festival entrega un archivo denominado *Utterance* (.utt) el cual contiene las frecuencias formantes necesarias para sintetizar cada fonema.

Para el caso de HTS, la síntesis tiene lugar en el arriba mencionado HTS-Engine (K Tokuda et al., 2002) por lo que será necesario convertir los archivos .utt al formato de éste último titulado *label* (.lab). Al igual que los .utt, los archivos .lab indican al vocoder las frecuencias formantes requeridas para determinada frase.

Veamos ahora que ocurre con los archivos de sonido que también son entrada al sistema. Los 300 archivos .wav fueron utilizados en la primera versión de HTS en español mexicano (Herrera-Camacho & Ávila, 2013). Fueron grabaciones realizadas con la voz de un locutor profesional en una cámara anecóica. Dichos archivos deben ser convertidos al formato RAW (.raw), los cuales son esencialmente archivos .wav sin encabezado.

Justamente son los archivos tipo. raw los que se descomponen en tres elementos: Coeficientes Generales Mel MGC, frecuencias fundamentales Logf0 y duración de estos.

Las ubicaciones de los datos correspondientes a la señal de voz: coeficientes mel-cepstral generales, f0 y duración (MGC, F0 y BAP) y los archivos *label* que incluyen la información texto a fonemas, están indicados en un archivo llamado Master Label File MLF, el cual es una requisición del software HTK para llevar a cabo los cálculos de probabilidades (Young, 2013). En éste MLF, los fonemas vienen también acomodados de acuerdo con su función de contexto, es decir, si inician o terminan palabra. También se considera su posición con respecto a fonemas anteriores o posteriores a él. La figura 3.4 muestra las entradas del sistema previo al entrenamiento de los HMMs.

Con base en los datos del MLF se genera una matriz gaussiana *prototipo* a partir de la cual se acomodarán los datos. Se nombra tal conjunto Hmm0. Mediante la instrucción HCompV, se calcula la media de dicha gaussiana y de acuerdo a este valor, se reagrupan los datos en una nueva gausssiana o modelo llamada Hmm1. Se consideran ahora como principal y se agrupan los datos individuales de los MLF en un solo archivo llamado Master Macro File MMF.

Los archivos MMF son de varios tipos de acuerdo con las características de los datos que los constituyen. En éste caso son: average, init, monophone, fullcontext, clustered, untied, re_clustered, tiedlist y stc. Todos ellos llevan él las probabilidades de ocurrencia y orden de aparición de los fonemas.

De acuerdo con el MMF se generan nuevos modelos agrupados según sus medias con la instrucción HERest. Respectivamente se nombran Hmm2, Hmm3.

Las pausas contenidas entre fonemas también se toman en cuenta, ellas se agrupan mediante HHed en otros modelos denominados Hmm4 y Hmm5. Nuevamente se ajustan sus varianzas con HERest.

Una vez armados los modelos, el sistema, utilizando cálculo de probabilidades Viterbi, toma los valores *más representativos* de cada modelo y organiza y con ellos un modelo lineal.

```
# MATLAB and STRAIGHT
USESTRAIGHT = 0
MATLAB      = /usr/bin/matlab -nodisplay -nosplash -nojvm
STRAIGHT    =

# Festival commands
USEUTT    = 1
TEXT2UTT  = /home/carlos/Festival_new/festival/examples/text2utt
DUMPFEATS = /home/carlos/Festival_new/festival/examples/dumpfeats

# speech analysis conditions
SAMPFREQ   = 48000   # Sampling frequency (48kHz)
FRAMELEN   = 1200   # Frame length in point (1200 = 48000 * 0.025)
FRAMESHIFT = 240 # Frame shift in point (240 = 48000 * 0.005)
WINDOWTYPE = 1 # Window type -> 0: Blackman 1: Hamming 2: Hanning
NORMALIZE  = 1  # Normalization -> 0: none  1: by power  2: by magnitude
FFTLEN     = 2048     # FFT length in point
FREQWARP   = 0.72   # frequency warping factor
GAMMA      = 1      # pole/zero weight for mel-generalized cepstral (MGC) analysis
MGCORDER   = 34   # order of MGC analysis
BAPORDER   = 24   # order of BAP analysis
LNGAIN     = 1     # use logarithmic gain rather than linear gain
LOWERF0    = 110    # lower limit for f0 extraction (Hz)
UPPERF0    = 280    # upper limit for f0 extraction (Hz)

# windows for calculating delta features
MGCWIN  = win/mgc.win
LF0WIN  = win/lf0.win
BAPWIN  = win/bap.win
NMGCWIN = 3
NLF0WIN = 3
NBAPWIN = 3

all: analysis labels
```

Figura 3.4 “Entradas en HTS”

Ese finalmente será el modelo del que se tomarán los datos para llevar a cabo la síntesis. A través de dos nuevas instrucciones propuestas exclusivamente para HTS: HMMSAlign y HSGen. La figura 4 muestra el modelo de selección para el fonema “a”.

Los diferentes hmm generados son después utilizados ya con las probabilidades de ocurrencia de cada fonema para poder calcular su lugar y espacio en una frase de acuerdo con los cálculos expuestos en la sección anterior. Con ello se agrupan en modelos de matrices gaussianas individuales en donde quedan agrupados todos los fonemas de características similares, por ejemplo, todos los fonemas /a/ en el mismo árbol, los fonemas /e/ en el mismo árbol y así sucesivamente. Con esto se consigue linealizar el proceso de selección

3.5 Coeficientes Mel General Cepstral

El concepto de **Mel General Cepstral** MGC (Keiichi Tokuda, Kobayashi, Masuko, & Imai, 1994) engloba dos parametrizaciones distintas para una señal de voz. El análisis Mel-Cepstral y el **de Linear Predictive Coding** LPC.

El análisis Mel-Cepstral es muy recurrente tanto en reconocimiento como en síntesis de voz, en él está basado el sintetizador HTS-MFCC que sigue en uso en el Laboratorio de Tecnologías del Lenguaje UNAM. La parte correspondiente a LPC es el punto de partida para la parametrización de voz propuesta la cual se basa en LSP.

El principio que rige a Mel General Cepstral consiste en definir el espectro H(z) de una señal de voz de la siguiente forma:

$$H(z) = S_{\gamma}^{-1}(\sum_{P=1}^{N} A_P z^{-P}) \quad (3.5)$$

Donde S_γ es una generalización de la función logaritmo:

$$S_{\gamma} = \begin{cases} \dfrac{\omega^{\gamma} - 1}{\gamma}, & 0 < |\gamma| \leq 1 \\ \log\omega, & \gamma = 0 \end{cases} \quad (3.6)$$

Ese principio aplicado a *H(z)* con la ecuación (3.5) nos da la siguiente información:

$$H(z) = \begin{cases} (1 + \gamma \sum_{P=1}^{N} A_P z^{-P})^{\frac{1}{\gamma}}, 0 < |\gamma| \leq 1 \\ \exp \sum_{P=1}^{N} A_P z^{-P}, \gamma = 0 \end{cases} \quad (3.7)$$

Cuando γ=0, la parametrización corresponde a la definición de Cepstrum, la cual forma parte del algoritmo para obtener la parametrización MFCC. Por otro lado, si γ=1 se obtiene una parametrización LPC de la cual se obtiene el LSP.

Para la conversión LPC a LSP se parte de que el filtro $H(z) = 1 + \sum_{P=1}^{N} A_P z^{-P}$ es igual a la suma de los polinomios P(z) y Q(z). (Zheng, Song, Li, Yu, & Wu,

1998) Estas ecuaciones se definieron en el capítulo anterior pero se repetirán a continuación para comodidad del lector. Cada uno de los polinomios se define de la siguiente forma:

$$P(z) = A_p(z) - z^{-(p+1)}A_p(z^{-1})$$
$$= 1 + \sum_{p=1}^{P}(a_{p+}a_{P+1-p})z^{-p} + z^{-(p+1)} \quad (3.8)$$

$$Q(z) = A_p(z) + z^{-(p+1)}A_p(z^{-1})$$
$$= 1 + \sum_{p=1}^{P}(a_{p-}a_{P+1-p})z^{-p} - z^{-(p+1)} \quad (3.9)$$

Todo polinomio tiene *P/2* pares de raíces complejas conjugadas, por lo que las ecuaciones se pueden representar de la siguiente forma:

$$P(z) = (1 + z^{-1})\prod_{i=1}^{\frac{P}{2}}(1 - z^{-1}e^{-j\omega})\,(1 - z^{-1}e^{-j\omega})$$
$$= (1 + z^{-1})\prod_{i=1}^{\frac{P}{2}}(1 - 2cos\omega_i z^{-1} + z^{-2}) \quad (3.10)$$

$$Q(z) = (1 - z^{-1})\prod_{i=1}^{\frac{P}{2}}(1 - z^{-1}e^{-j\theta})\,(1 - z^{-1}e^{-j\theta})$$
$$= (1 - z^{-1})\prod_{i=1}^{\frac{P}{2}}(1 - 2cos\theta_i z^{-1} + z^{-2}) \quad (3.11)$$

Los valores de *ω* y *Θ* representan en *P(z)* y *Q(z)* respectivamente las frecuencias formantes del fonema a representar. Todas ellas van entrelazadas en el intervalo *(0, π)* y se les conoce como **Line Spectral Frequencies** LSF.

Capítulo 4

En esta sección se presenta la documentación correspondiente a las pruebas realizadas al sistema. Se pretende hacer un resumen de los métodos estadísticos de evaluación que se aplicaron recientemente sobre las parametrizaciones HTS-LSP y HTS-MFCC, Se hicieron pruebas MOS, MUSHRA, CCR, ABX para valorar naturalidad y una prueba SUS para valorar la inteligibilidad.

4.1 Introducción

Se trabajó hace algún tiempo en el Laboratorio de Tecnologías del Lenguaje sobre la propuesta del sintetizador HTS (Keiichi Tokuda et al., 2013) para desarrollar un sintetizador en español del centro de México (Herrera-Camacho & Ávila, 2013). Ese trabajo dentro de otras cosas tomó la parametrización de voz basada en los coeficientes cepstrales de frecuencia en escala Mel llamados en inglés *Mel Frequency Cepstral Coefficients* MFCC. Luego de llevar a cabo algunas pruebas estadísticas de valoración con usuarios (Franco, Del Rio, et al., 2016), se consideró la utilizar una nueva parametrización de voz alternativa a los MFCC que había tenido poco uso en reconocimiento y síntesis de voz pero que sin embargo sigue vigente. Dicho esquema se basa en el Par Linear Espectral o *Line Spectral Pair* LSP para representar la voz (Nakatani et al., 2006). La parametrización fue implementada y de igual modo valorada estadísticamente (Franco, Herrera, & Escalante, 2017).

La primera valoración se valió exclusivamente de las pruebas MOS (ITU-T, 2016) para calificarse, y era necesario además de conocer la opinión del usuario en términos de naturalidad e inteligibilidad, en qué posición se encontraba la voz parametrizada con LSP con respecto a la voz parametrizada utilizando MFCC. Ya que ambas parametrizaciones fueron programadas en el sistema HTS se denominó a cada una HTS-LSP y HTS-MFCC respectivamente. Los resultados mostraron una ligera superioridad de la voz HTS-LSP en la preferencia de los encuestados (Franco et al., 2017).

Ya que la voz parametrizada con MFCC es un estándar en síntesis y reconocimiento de voz, los autores juzgaron necesario aplicar más pruebas

que sustentaran o en un momento refutaran los resultados de la valoración MOS. Se eligieron tres pruebas más para naturalidad: MUSHRA, ABX y CCR, la inteligibilidad se valoró usando SUS. Los detalles y resultados de cada prueba se muestran a continuación.

4.2 Evaluación MOS

La prueba MOS es sin duda de las más utilizadas para medir calidad de audio de telecomunicaciones (ITU-T, 2016). Por esta razón fue el punto de partida al momento de valorar la parametrización HTS-LSP. Se tomó una población de 31 encuestados. Cada uno de ellos escuchó 5 frases, en tres versiones: Voz original, voz sintetizada por HTS-MFCC y voz sintetizada por HTS-LSP. Se les pidió evaluar naturalidad e inteligibilidad en una escala de 0 a 5 en ambos casos. Los resultados promedio fueron los siguientes:

Valor Estadístico	Naturalidad HTS-LSP	Inteligibilidad HTS-LSP	Naturalidad HTS-MFCC	Inteligibilidad HTS-MFCC
Promedio (CI 95%)	3.47	3.6	3.07	3.44
Desviación Estándar	0.56	0.57	0.65	0.76
Máximo	4.8	5	4	5
Mínimo	2.4	2.8	1.8	1.4

Tabla 1. Resultados de MOS

Podemos ver a través de los resultados de las pruebas MOS que la parametrización HTS- LSP gozó de una mayor aceptación en la población entrevistada. Los promedios tienen un intervalo de confianza *confidence Interval* CI de 95%. En general están arriba de la media en la escala de calificaciones de la norma que sería de 2.5. Para tener una seguridad mayor en nuestros resultados se procedió a aplicar otra serie de pruebas, dando especial atención a la parametrización HTS-LSP que fue utilizada recientemente por los autores (Franco et al., 2017).

4.3 Evaluación MUSHRA

La prueba MUSHRA (Itu-BS.1534, 2015) es una norma recomendada por la *International Telecommunications Union* ITU diseñada específicamente para la evaluación de diversos códecs de audio. Está organizada de forma tal que el entrevistado analiza el mismo contenido de audio codificado de diferentes maneras, incluida la grabación original en archivo *lossless* (wave o aiff) y también filtrada con frecuencia de corte 3500 Hz para que sirva de "ancla" a quien escucha. Dicho de otra forma, para que el entrevistado tenga oportunidad de escuchar el audio original con ligeras modificaciones y verificar que no se autoengaña con la referencia.

Se entrevistó a una población de 11 escuchas. Todos ellos son especialistas en ingeniería de sonido o estudiantes de tecnología musical, ya que la norma pide escuchas con experiencia en el campo. Cada persona escuchó 5 frases en cuatro versiones diferentes: La grabación original, la grabación original filtrada con pasa bajas a una frecuencia de corte de 3.5 kHz, una versión sintetizada usando parametrización HTS-MFCC y finalmente una versión sintetizada usando HTS-LSP. El sujeto se sentó frente a una computadora y escuchó las frases usando audífonos con reducción señal ruido de 93 dB. Las frases se reproducen en orden aleatorio cada vez que se repite la prueba. De acuerdo con la norma, cada frase se tiene que validar en una escala del 0 al 100 donde al menos una frase debe llevar 100 de calificación. La tabla 2 muestra los resultados de la evaluación, los cuales aparecen graficados en la figura1.

Type	Reference	Anchored	LSP	MFCC
	100	53	90	63
	100	77	81	74
	100	73	77	74
	100	55	76	75
	100	74	90	46
	100	54	78	70
	100	70	76	40
	100	86	53	71
	100	50	40	30
	100	30	30	50
	100	67	74	83
Average	100	62.6363636	69.5454545	61.4545455

Tabla 2. Resultados de MUSHRA

La referencia fue reconocida y valorada con la máxima calificación en todo momento por parte de los sujetos de prueba. El ancla sorpresivamente fue valorada con menor calificación con respecto a la parametrización HTS-LSP y 1.5 puntos arriba de la parametrización HTS-MFCC. Entre estas dos últimas hay una diferencia de puntaje de 7 puntos resultando más alta la parametrización HTS-LSP. Todos los promedios tienen un intervalo de confiabilidad CI de ±95%.

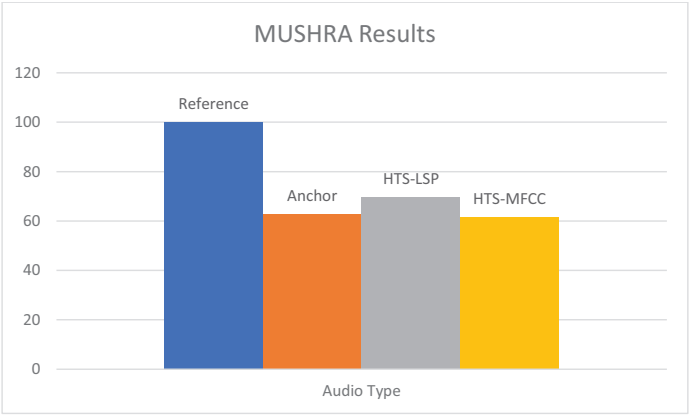


Figura 1. “Gráfica de promedios de prueba MUSHRA”

4.4 Comparación MOS y MUSHRA

Considerando que la escala de calificaciones MOS es de 0 a 5. Vemos que HTS-LSP tuvo un promedio de 3.47 es decir está en un 69.4% de la calificación máxima. Lo cual está muy cerca de su calificación MUSHRA (que va de 0 a 100) de 69.54.

La población encuestada fue totalmente distinta en ambas pruebas por lo que podemos concluir satisfactoriamente que hay consistencia en las opiniones de la gente que escuchó la parametrización.

4.5 Otros métodos para valorar la Naturalidad

Con objeto de respaldar los datos obtenidos de las pruebas MOS y MUSHRA, se echó mano de otros métodos un poco más tradicionales para valorar la naturalidad.

4.5.1 Valoración de Naturalidad usando ABX

El método de valoración ABX (Munson & Gardner, 1950), consiste en presentar al sujeto que escucha dos ejemplos de sonidos A y B para que señale que tanto se aproxima a la referencia X, la cual es una tercera muestra de sonido. Con esto se busca que tan semejantes son ambos sistemas y si hay consistencia por parte del sujeto al emitir su opinión.

La aplicación en este caso de ABX consiste en mostrar al escucha un tipo de voz sintetizada A, un tipo de voz sintetizada B y enseñarle también una grabación de voz natural como X, con esto veríamos que tanto se acerca alguno de las dos voces artificiales a la voz original.

Para probar nuestra parametrización, se utilizaron como A una frase sintetizada usando HTS-LSP. Como B la misma frase sintetizada usando HTS-MFCC y la referencia X fue la frase grabada por el locutor que prestó su voz para nuestro proyecto.

En la prueba el usuario debió responder con los conceptos “mucho” o “poco” a dos preguntas: “¿Qué tanto se parece A a X?” y “¿Qué tanto se parece B a X?”.

Se hizo la prueba a 30 personas, en su mayoría estudiantes universitarios con promedio de edad de 23 años. A la primera pregunta, donde se valora HTS-LSP, 17 personas contestaron “mucho” y 13 respondieron “poco”. A la segunda pregunta correspondiente a HTS-MFCC los resultados fueron 10 de “mucho” y 20 de “poco”.

Como se esperaba y de acuerdo con los resultados de las pruebas anteriores, la HTS-MFCC tuvo comparativamente una menor aceptación que la contraparte basada en LSP

La calificación de ABX es cualitativa, si le otorgamos el valor de 1 a mucho y 0 a poco la calificación máxima posible sería de 30 dada la población. En términos proporcionales, 17 es un 56.6% de 30. Lo cual también es consistente con los resultados obtenidos en MOS y MUSHRA los cuales van cerca del 60%

4.5.2 Prueba CCR

Cuando el objetivo es medir diferencias de calidad entre dos sistemas, una prueba de comparación de categoría *Comparison Category Rating* (CCR)

puede ser utilizada. La prueba CCR (ITU-T, 1996) consiste en reproducir a un escucha dos voces sintetizadas distintas y para valorarlas utiliza una escala discreta de 7 puntos de -3 (muy malo) hasta 3 (muy bueno). Los resultados se promedian para obtener un promedio de opinión de calificaciones de comparación (CMOS) para cada voz sintetizada.

La figura 2 señala los resultados obtenidos en la evaluación CCR. En esta evaluación, HTS-LSP tuvo una aceptación mucho mayor que HTS-MFCC. Ambos promedios fueron 1.04 y 0.47 respectivamente. En términos porcentuales, sería necesario usar una escala del 0 al 7 donde la calificación de 1 sería equivalente a 5, la calificación -1 sería 4, la calificación 0 equivale a 3 y así sucesivamente. En esos términos, HTS-LSP tiene un porcentaje de 71.42 % que es consistente con lo obtenido en MUSHRA y MOS donde la calificación es 69% cercana al máximo.

4.6 Conclusiones

Las pruebas ABX y CCR se hicieron a las dos parametrizaciones para tener una valoración relativa de ambas. Si bien pudieron reportarse los resultados de HTS-LSP únicamente, era necesario mostrar en qué posición está LSP con respecto al estándar de parametrización de voz basado en MFCC.

Podemos ver con ambas pruebas, aunadas a los resultados arrojadas por MOS y MUSHRA que la voz parametrizada utilizando HTS-LSP resulta al escucha promedio mucho más natural que aquella parametrizada con HTS-MFCC. Las pruebas nos permiten ver que la parametrización LSP tuvo mejor aceptación. Cabe mencionar sin embargo que dicha aceptación no fue calificada muy por encima de la otra. Ambas parametrizaciones están a un nivel cercano en aceptación del usuario. La elección de la parametrización dependerá exclusivamente del uso que se le quiera dar al sistema de síntesis.

Los resultados de las cuatro pruebas muestran que la naturalidad está a un 70% de aproximarse al ideal. Esas fallas pudieran tienen diferentes causas las cuales las cuales ameritan una reflexión sobre el sistema en su totalidad para ver exactamente que debemos modificar para lograr ese 30% restante.

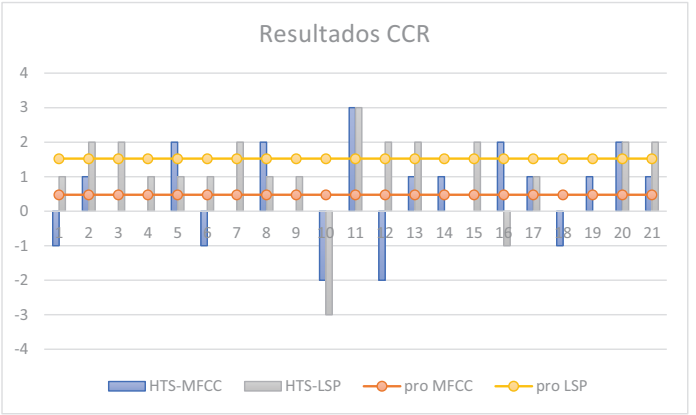


Figura 2. Resultados de evaluación CCR

Debemos también tener en cuenta que la naturalidad en la voz es un concepto complejo y calificarla depende de varios factores como la expectativa o la experiencia del sujeto que hace la prueba o el contexto de su aplicación. Es decir, la valoración final se dará escuchando una aplicación concreta de la voz sintetizada, por ejemplo, en un GPS o un personaje animado.

4.6.1 Valoración de Inteligibilidad

Como se menciona anteriormente, se realizó una prueba SUS *Semantically Unpredictable Sentences* (Benoît, Grice, & Hazan, 1996) para valorar la inteligibilidad. 30 personas tomaron dictado de 5 oraciones sintetizadas utilizando HTS-LSP. Los sujetos fueron estudiantes universitarios cuyo promedio de edad es 23 años. Las oraciones fueron semánticamente irregulares, es decir sin un significado lógico. Esto se hace con objeto de evitar que el sujeto de prueba corrija de manera inconsciente los errores que pudieran suscitarse si se hace un dictado de oraciones con un significado claro. Recordemos que el ser humano tiende a atribuir significado a las palabras de acuerdo con el contexto semántico del mensaje y no individualmente a cada palabra.

Los enunciados fueron:

1. El perro amarillo voló detrás de la almohada.
2. Me gusta bailar de cabeza sobre el mar.
3. Cielos de mermelada sobre lagos de fierro.
4. El club de viento se saturó de pinturas abstractas.
5. La hermosa detective se cansó de tanta azúcar.

El dictado tuvo lugar en un salón de clase de 10 por 10 metros cuadrados. Se utilizó una bocina Bosé modelo *Soundlink* conectada por Bluetooth a una computadora portátil. El suscrito escucho las oraciones sentado en la parte posterior del salón para asegurarse de que había claridad de escuchar el audio aún a diez metros del altavoz.

Los dictados fueron revisados y se le dio una calificación de dos puntos a cada oración escrita correctamente. El promedio de calificación en los exámenes fue de 6 puntos. En promedio dos de cinco oraciones no resultaron claras al escucha. La tabla 3 muestra las fallas que hubo en las frases.

Frase Número	Texto	Número de errores
1	El perro amarillo voló detrás de la almohada.	16
2	Me gusta bailar de cabeza sobre el mar	2
3	Cielos de mermelada sobre lagos de fierro.	23
4	El club del viento se llenó de pinturas abstractas.	10
5	La hermosa detective se cansó de tanta azúcar.	3

Tabla 3. Errores en las frases dictadas.

La frase número 3 fue la más difícil de identificar por el grupo, seguida de la frase número dos. Estás dos frases son las que tienen menos regularidad en su contenido semántico. La mayoría de estos errores dentro de la frase 3 se encuentra en la mala identificación de la palabra *fierro,* varias personas escribieron *hierro*. En la frase uno, la palabra donde falló la mayoría fue *almohada* muchos entendieron *alborada.* En ambos casos ninguna de las frases pierde sentido si reemplazamos las palabras hierro por fierro y almohada por alborada respectivamente. De aquí podemos nuevamente notar la capacidad del cerebro humano de inconscientemente dotar de un sentido

lógico a la oración. Si observamos las oraciones restantes vemos que, si bien no tienen contenido de uso común, su significado no resulta tan disparatado y por tanto es más sencillo de entender.

4.6.2 Conclusiones respecto a la Inteligibilidad

Atribuimos las fallas de las oraciones 1 y 3 a que su contenido semántico resulta inverosímil en exceso y por esta razón el sujeto se resistió a escribirlo tal cual se oye. Si se sintetizan en HTS-LSP, frases con las palabras que menos se entendieron (fierro y almohada) o las palabras en sí, no presentan problema para identificarse.

Es imposible hacer un sistema de síntesis de voz que sea completamente inteligible en términos semánticos ya que la Inteligibilidad no depende únicamente de una capacidad auditiva sino también cognitiva. Prueba de ello es que el tipo de problemas aquí expuesto bien podrían ocurrir si una persona dictara las mismas cinco frases a viva voz.

Capítulo 5

El presente capítulo busca retomar la interpretación de los resultados de la prueba MUSHRA y hacer una valoración de la parametrización con LSP. Se da un panorama de cuál sería el próximo paso en el presente trabajo.

5.1 Conclusiones

En primer lugar, podemos decir que la parametrización de voz utilizando LSP cumple de manera satisfactoria la premisa de ser una alternativa a la voz parametrizada con MFCC. Se recomienda incluso como una mejor opción dado el nivel de aceptación que tuvo en las pruebas MOS aplicadas.

Por otro lado, la parametrización LSP es reversible al proceso de filtrado LCP que representa directamente la señal de voz, el cual puede ser reutilizado en otros procesos mientras que con los MFCC no podemos regresar a la señal original.

La tercera ventaja que posee el LSP es el tamaño del archivo, el cual es más pequeño que el de MFCC y nos permite economizar en procesamiento y espacio en memoria.

La desventaja de LSP continúa siendo la falta de naturalidad en la voz producida. En ese sentido consideramos que hay aún trabajo por hacer que parte hacia dos ramas. La primera de ellas y con la cual se ha estado experimentando en últimos meses es la de modificar la voz del locutor antes de hacer el entrenamiento del HTS.

Dicha modificación se lleva a cabo haciendo una comparación punto a punto del espectro de la señal de voz con el espectro de una señal sintetizada. Ambas señales partiendo exactamente de la misma frase.

El otro camino posible para conseguir una mejora en términos de naturalidad parte de utilizar otro método en la selección de los difonemas. En este caso, en lugar de utilizar un árbol estocástico, se echa mano de las Redes Neuronales Profundas *Deep Neural Networks* DNN las cuales están en boga actualmente en el campo de las tecnologías del leguaje, tanto en análisis como, en síntesis.

El problema de tomar este camino es que las DNN aún están en transición de teoría a práctica y los modelos funcionales de sistemas de síntesis de voz son aún escasos. El grupo de Tecnologías del Lenguaje de la UNAM está incursionando también en el tema, sin embargo, los resultados serán parte de proyectos de investigación posteriores a la tesis doctoral.

Bibliografía

Arakawa, A., Uchimura, Y., Banno, H., Itakura, F., & Kawahara, H. (2010). High quality voice manipulation method based on the vocal tract area function obtained from sub-band LSP of STRAIGHT spectrum. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (2), 4834–4837. https://doi.org/10.1109/ICASSP.2010.5495142

Backstrom, T. (2004). *Linear predictive modelling of speech - constraints and line spectrum pair decomposition*. *Matrix*. Retrieved from https://aaltodoc.aalto.fi/bitstream/handle/123456789/2392/isbn9512269473.pdf?sequence=1

Bäckström, T., & Magi, C. (2006). Properties of line spectrum pair polynomials-A review. *Signal Processing*, *86*(11), 3286–3298. https://doi.org/10.1016/j.sigpro.2006.01.010

Benoît, C., Grice, M., & Hazan, V. (1996). The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences. *Speech Communication*, *18*(4), 381–392. https://doi.org/10.1016/0167-6393(96)00026-X

Birkholz, P., & Jackel, D. (2003). A three-dimensional model of the vocal tract for speech synthesis. *Of the 15th International Congress of …*. Retrieved from http://rickvanderzwet.nl/trac/personal/export/360/liacs/API2010/workshop1/birkholz-2003-icphs.pdf

Birkholz, P., Jackèl, D., & Kroger, B. (2006). Construction and control of a three-dimensional vocal tract model. *Acoustics, Speech and Signal*. Retrieved from http://ieeexplore.ieee.org/abstract/document/1660160/

Black, A. (2006). CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling. *INTERSPEECH*. Retrieved from http://www.scs.cmu.edu/afs/cs.cmu.edu/Web/People/awb/papers/is2006/IS061394.PDF

Black, A., & Taylor, P. (1997). Automatically clustering similar units for unit selection in speech synthesis. Retrieved from https://www.era.lib.ed.ac.uk/handle/1842/1236

Chennoukh, S., Gerrits, A., & Miet, G. (2001). Speech enhancement via frequency bandwidth extension using line spectral frequencies. *Acoustics, Speech, and*. Retrieved from http://ieeexplore.ieee.org/abstract/document/940919/

CMU. (2016). Festival. Retrieved September 20, 2017, from http://www.cstr.ed.ac.uk/projects/festival/

Davis, S., & Mermelstein, P. (1978). Evaluation of acoustic parameters for monosyllabic word identification. *The Journal of the Acoustical Society of*. Retrieved from http://asa.scitation.org/doi/abs/10.1121/1.2004059

Dutoit, T. (2008). Corpus-Based Speech Synthesis. In *Springer Handbook of Speech Processing* (pp. 437–456). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-49127-9_21

Dutoit, T., & Leich, H. (1993). MBR-PSOLA: Text-to-speech synthesis based on an MBE re-synthesis of the segments database. *Speech Communication*. Retrieved from http://www.sciencedirect.com/science/article/pii/016763939390042J

Fant, G. (n.d.). *Acoustic theory of speech production : with calculations based on X-ray studies of Russian articulations*. Retrieved from https://books.google.com.mx/books/about/Acoustic_Theory_of_Speech_Production.html?id=qa-AUPdWg6sC&redir_esc=y

Franco, C., Del Rio, F., & Herrera, A. (2016). ATINER Conference Paper Series Speech Synthesis of Central Mexico Spanish using Hidden Markov Models, 1–12.

Franco, C., Herrera, A., & Del Río, F. (2016). Speech Synthesis in Mexican Spanish using LSP as voice parameterization.

Franco, C., Herrera, A., & Escalante, B. (2017). Speech Synthesis in Mexican Spanish using LSP as voice parameterization. *Iiisci.org*. Retrieved from http://www.iiisci.org/Journal/CV$/sci/pdfs/SA404GP17.pdf

Ganchev, T. (2011). *Contemporary methods for speech parameterization*. Springer.

Goncharoff, V., & Gries, P. (1998). An algorithm for accurately marking pitch pulses in speech signals. *Proc. of the SIP'98*. Retrieved from https://scholar.google.com/scholar?hl=es&q=9.%09V+Goncharoff%2C+P+Gries+"An+algorithm+for+accurately+marking+pitch+pulses+in+speech+signals"+-+Proc.+of+the+SIP%2798%2C+1998&btnG=&lr=

Herrera-Camacho, A., & Ávila, F. D. R. (2013). Development of a Mexican Spanish Synthetic Voice Using Synthesizer Modules of Festival Speech and HTSStraight. *International Journal of Computer and Electrical Engineering*, 36–39. https://doi.org/10.7763/IJCEE.2013.V5.657

Holmes, J. N., & Holmes, W. (Wendy J. . (2001). *Speech synthesis and recognition*. Taylor & Francis.

Homer Dudley's Speech Synthesisers. (n.d.). Retrieved from http://users.polytech.unice.fr/~strombon/SSI/z.Supplements/vocoder/http___www.obsolete.pdf

HTS. (2015). hts_engine API. Retrieved September 20, 2017, from http://hts-engine.sourceforge.net/

Itakura, F., & Sugamura, N. (1979). LSP speech synthesizer its principle and implementation. *Trans. of the Committee on Speech Research*.

Itu-BS.1534. (2015). Method for the subjective assessment of intermediate quality level of audio systems Policy on Intellectual Property Right (IPR) Series of ITU-R Recommendations, 1534–3. Retrieved from http://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1534-3-201510-I!!PDF-E.pdf

ITU-T. (1996). T-REC-P.800-1996, *800*.

ITU-T. (2016). Recommendation ITU-T P.800.1 : Mean opinion score (MOS) terminology.

Kabal, P., & Ramachandran, R. (1986). The computation of line spectral frequencies using Chebyshev polynomials. *IEEE Transactions on Acoustics,*. Retrieved from http://ieeexplore.ieee.org/abstract/document/1164983/

Kang, H., & Liu, W. (2006). Selective-LPC based Representation of STRAIGHT Spectrum and Its Applications in Spectral Smoothing*. Retrieved from https://pdfs.semanticscholar.org/421c/34834037d4afbee63278cda4b6d334f05ed8.pdf?_ga=2.64644329.778383573.1494899526-1020446631.1494899442

Klatt, D. H. (n.d.). Software for a cascade/parallel formant synthesizer. Retrieved from http://www.fon.hum.uva.nl/david/ba_shs/2009/klatt-1980.pdf

McLoughlin, I. (2008). Line spectral pairs. *Signal Processing*. Retrieved from http://www.sciencedirect.com/science/article/pii/S0165168407003167

Moulines, E., & Charpentier, F. (1990). Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*. Retrieved from http://www.sciencedirect.com/science/article/pii/016763939090021Z

Munson, W. A., & Gardner, M. B. (1950). Standardizing Auditory Tests. *The Journal of the Acoustical Society of America*, *22*(5), 675–675. https://doi.org/10.1121/1.1917190

Nakatani, N., Yamamoto, K., & Matsumoto, H. (2006). Mel-LSP Parameterization for HMM-based Speech Synthesis. *Eurasip Proceedings SPECOM 2006*. Retrieved from http://www.eurasip.org/Proceedings/Ext/SPECOM2006/papers/045.pdf

Rabiner, L. R. (2015). Lawrence Rabiner - MATLAB Central. Retrieved September 20, 2017, from https://www.mathworks.com/matlabcentral/profile/authors/12136-lawrence-rabiner

Sagayama, S., & Itakura, F. (2002). Symmetry between linear predictive coding and composite sinusoidal modeling. *Electronics and Communications in Japan, Part III: Fundamental Electronic Science (English Translation of Denshi Tsushin Gakkai Ronbunshi)*, *85*(6), 42–54. https://doi.org/10.1002/ecjc.1100

Soong, F., & Juang, B. (1984). Line spectrum pair (LSP) and speech data compression. *Acoustics, Speech, and Signal Processing,*. Retrieved from http://ieeexplore.ieee.org/abstract/document/1172448/

Sptk. (2013). Reference Manual for Speech Signal Processing Toolkit.

Stevens, K., Kasowski, S., & Fant, C. (1953). An electrical analog of the vocal tract. *The Journal of the Acoustical*. Retrieved from http://asa.scitation.org/doi/abs/10.1121/1.1907169

Stylianou, Y. (2008). Voice Transformation. In *Springer Handbook of Speech Processing* (pp. 489–504). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-49127-9_24

Taylor, P., Black, A., & Caley, R. (1998). The architecture of the Festival speech synthesis system. Retrieved from https://www.era.lib.ed.ac.uk/handle/1842/1032

Tokuda, K., Kobayashi, T., Masuko, T., & Imai, S. (1994). Mel Generalized Cepstral Analysis — A Unified Approach to Speeh Spectral Estimation. Retrieved from http://www.sp.nitech.ac.jp/~tokuda/selected_pub/pdf/conference/tokuda_icslp1994.pdf

Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., & Oura, K. (2013). Speech Synthesis Based on Hidden Markov Models. *Proceedings of the IEEE*, *101*(5), 1234–1252. https://doi.org/10.1109/JPROC.2013.2251852

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T. ., & Kitamura, T. . (2000). Speech Parameter Generation Algorigthms for {HMM}-Based Speech Synthesis. *Proceedings\ ICASSP 2000*, (l), 1315–1318. Retrieved from http://ieeexplore.ieee.org/abstract/document/861820/

Tokuda, K., Zen, H., & Black, A. (2002). An HMM-based speech synthesis system applied to English. *IEEE Speech Synthesis Workshop*. Retrieved from http://www.scs.cmu.edu/afs/cs.cmu.edu/Web/People/awb/papers/IEEE2002/hmmenglish.pdf

Trangol, J., & Herrera, A. (2015). Traditional Method and Multi-Taper to Feature Extraction Using Mel Frequency Cepstral Coefficients. *International Journal of Information and*. Retrieved from http://search.proquest.com/openview/10ff98b02b7123d58901b88599a26de6/1?pq-origsite=gscholar&cbl=2027420

Young, S. (2013). The HTK Book. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

Zheng, F., Song, Z., Li, L., Yu, W., & Wu, W. (1998). The distance measure for line spectrum pairs applied to speech recognition. *Proceedings of the 5th International Conference on Spoken Language Processing 1998 (ICSLP '98)*, 1123–1126. Retrieved from http://www.isca-speech.org/archive/icslp_1998/i98_0171.html