TRABAJO FIN DE GRADO

INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

Conversión de voz basada en Modelos Ocultos de Markov

Autor
Eduardo Sesma Caselles

Director
Luis Vicente Borruel

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Escuela de Ingeniería y Arquitectura

—

Zaragoza, 2015

AGRADECIMIENTOS:

Mi idea no era poner el apartado de agradecimientos de ámbito personal en una memoria de un proyecto científico-tecnológico, pero este proyecto es más que eso. Este proyecto pone punto y a parte a un gran reto, una parte importante de mi vida donde ha habido mucha gente presente... y por este motivo creo que se merecen estar aquí también presentes y agradecérselo:

...

A mi tío César. Por alimentarme durante toda mi vida con mil nuevos inventos, por estar siempre ahí y despertar en mi el interés por las tecnologías. Sin duda, él fue el principio de todo esto.

.

A mis abuelos, los cuales ya no pueden estar, pero siempre los tengo presentes. Por ser parte de mí, al igual que sus valores de amor y constancia.

.

A mi hermano, del que siempre estaré orgulloso de como es y lo que me aporta.

.

A los Esmochaos, mis compañeros de clase. Esa nueva gran familia de telecos con los que sin ellos no hubiera sido posible. Nos sigamos cruzando en el camino o no, os deseo el mejor y más prometedor de los futuros. "Aún vais..."

.

A mis profesores, los buenos y los malos. Desde los que siempre estaban disponibles por muy pesado que fuera, hasta los que nunca estaban. De todos he aprendido algo importante y no sólo de matemáticas o física. Gracias de por vida.

.

A Luis, el director de este proyecto. Por brindarme su tiempo tanto para lo puramente académico como para lo demás, por su manera de hacer las cosas, por su paciencia conmigo y por apostar por mi... pero sobretodo, por hacer del trabajo duro algo agradable. Ha sido un placer compartir todo esto contigo.

.

A mi compañera, mi mitad. A la que le debo gran parte de cada cosa que consigo.

.

Y los más importantes... a mis padres. Por apoyarme sin dudar en todo lo que me he propuesto, por asumir cada gasto que ha conllevado llegar hasta aquí, por los momentos de pararme los pies cuando era necesario o los momentos de levantarme cuando lo necesitaba y porque nunca me ha faltado de nada gracias a su esfuerzo. Os lo debo todo.

Resumen

El concepto de conversión de voz (heredado del inglés "voice-morphing"), se refiere a la técnica empleada para la transformación de la voz de un locutor fuente para que suene como si fuera hablada por un locutor objetivo.

Existen varias aplicaciones que pueden beneficiarse de este tipo de tecnología. Por ejemplo, en un sistema de conversión TTS (Text-To-Speech, conversión de texto a voz) que la tenga, podría producir diferentes voces. O más importante aún, en casos donde la identidad del locutor tiene un rol importante, como el doblaje de películas, un único locutor podría doblar todas las voces con la identidad propia de cada actor.

El objetivo de este proyecto es implementar el prototipo de una herramienta que analice las características de una locución de un locutor fuente y sintetice esa misma locución con la identidad de un locutor objetivo. El resultado, por lo tanto, debe preservar las características tonales y dinámicas del locutor fuente, mientras que en su percepción se debe reconocer la identidad del locutor objetivo como si este la hubiera pronunciado.

Para conseguir este objetivo, el prototipo dispone de un nuevo tipo de estructura basada en el uso de Modelos Ocultos de Markov. Esta metodología nos permite realizar un modelado estadístico de los locutores con el cual poder generar una nueva voz que contenga las características deseadas de cada uno.

El desarrollo del proyecto se organizó en cinco bloques. El bloque de documentación, tanto para adquirir conocimientos como para pasar lo adquirido a esta memoria, el de preparación de todo el entorno de trabajo, una fase de análisis, y finalmente las partes de implementación y obtención de resultados. Por último, se constató que el modelo es implementable y los resultados fueron satisfactorios, pero mejorables.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Presentación

La conversión de voz ("voice morphing" más comúnmente conocida en inglés) es usada para alterar la voz de una persona a través de software. Pueden existir diferentes propósitos para llevar esta alteración a cabo, desde hacer un simple efecto de audio a la voz, esconder la identidad de esta, hasta suplantar la identidad de otro locutor.

Todo comenzó con George Papcun a finales de los 90 cuando, junto con su grupo de trabajo en Los Alamos National Laboratory de Nuevo México, sintetizó una réplica convincente de la voz de un general del ejército. Partiendo de una grabación de diez minutos, se generó una locución que aseguraba que "quería reunir a las tropas para tomar el gobierno de los Estados Unidos". Obviamente, esto causó mucho revuelo y dio una gran publicidad a lo que conocemos como voice morphing, llegando a intervenir la CIA y dando lugar a varias teorías de la conspiración.

Actualmente existen varias aplicaciones disponibles de conversión de voz en diversos campos. Podemos encontrar desde AutoTune, el cual puede reafinar la voz de los cantantes, hasta MorphVox Pro, el cual te permite hablar con diferentes voces humanoides, robots e incluso animales.

En este proyecto se busca implementar una herramienta de conversión voz a voz, donde partiendo de una señal de voz de un locutor fuente se transforme para que suene como la voz de un locutor objetivo. Esta herramienta podría ser utilizada, por ejemplo, para doblaje de películas donde un solo locutor podría doblar todos los personajes con la identidad propia de cada uno.

1.2. Objetivos

En este proyecto se busca implementar el prototipo de una herramienta que extraiga una serie de parámetros característicos de la locución de un locutor fuente y genere esa misma locución con la identidad de un locutor objetivo.

En lo referente a tratamiento del habla, la conversión de voz de un locutor fuente con la finalidad de sonar como un locutor objetivo es una de las tareas más complicadas de realizar. En ella nos encontramos con tres problemas independientes que se deben solucionar antes de construir el sistema de conversión. En primer lugar, necesitamos un modelo matemático que represente la señal de voz con el cual poder generar una nueva voz sintética pudiendo manipular la prosodia[1] de esta sin problemas. En segundo lugar, saber extraer los marcadores característicos por los cuales una persona puede ser identificada por un oyente. Y por último, determinar el tipo de función de conversión.

Con el paso de los años las tecnologías de síntesis de voz han ido avanzando, desde la creación de modelos paramétricos del tracto vocal a nivel de fonema, pasando por la concatenación de fragmentos de audio pregrabado, hasta la la síntesis paramétrica por Modelos Ocultos de Markov (HMMs) estadístico-contextuales. En este proyecto se va a utilizar esta última tecnología, apoyándonos en la versatilidad que nos dan sus modelos, ya que modificando unos coeficientes y sin la necesidad de procesar el audio a nivel de señal, podemos manipular la prosodia sin problemas.

Los Modelos Ocultos de Markov, aparte de ser el sistema más novedoso, nos proporciona un mayor nivel de inteligibilidad que los de concatenación de fragmentos y, como se verá más adelante, un ahorro considerable de memoria, lo cual es un punto muy a tener en cuenta si en un futuro queremos hacer una aplicación para dispositivos móviles.

1.3. Organización de la memoria

Tras este capítulo a modo de introducción y resumen del proyecto, la memoria continúa organizada en los siguientes puntos:

- En el capítulo 2 se plantean los problemas a los que nos tenemos que

[1] Conjunto de las características sonoras que hacen referencia a la pronunciación, teniendo en cuenta el acento, el tono y la cantidad.

enfrentar desde el punto de vista teórico y los diferentes sistemas que se han usado hasta ahora para la conversión de voz dando un repaso al actual estado del arte. Incluye varias secciones donde se explican los conceptos que debemos saber manejar en este proyecto, como son:

  - Los conceptos básicos de modelado de una señal de voz y como se puede realizar su síntesis.
  - Como funcionan los Modelos Ocultos de Markov, su uso para el procesado de voz, entrenamiento y síntesis, y el software que utilizaremos.
  - Los conceptos referentes a la conversión de voz y los diferentes métodos que se pueden usar para hacerlo, repasando el estado del arte y finalizando con unas mejoras propuestas.

- En el capítulo 3 se propone la solución y estructura del sistema que se va a adoptar para realizar la conversión de voz.
- En el capítulo 4 se explican los detalles de la metodología llevada a cabo para el desarrollo del prototipo de la herramienta de conversión de voz haciendo referencia tanto a la estructura de datos y archivos, como a las funciones y herramientas utilizadas.
- En el capítulo 5 se exponen los resultados obtenidos con el uso del prototipo y se comparan con los de otras herramientas.
- En el capítulo 6 se presentan las conclusiones del proyecto y se exponen unas posibles líneas futuras de trabajo.
- Tras el capítulo 6, se adjunta el apéndice A donde se explica un ejemplo de Modelo Oculto de Markov para poder entender mejor este concepto.
- En el apéndice B se adjuntan los detalles de las funciones y scripts más relevantes que se han creado.
- En el apéndice C se expone el calendario que se ha seguido para la elaboración de este proyecto.
- Para finalizar, se adjunta la bibliografía a la que se ha hecho referencia durante esta memoria.

Capítulo 2

Planteamiento del problema

2.1. Modelado de una señal de voz

2.1.1. Modelo humano de producción de la voz

Si queremos extraer y generar parámetros de la voz humana, debemos saber como podemos modelar un sistema que represente exactamente el proceso de producción de la voz. El sistema de producción humana es muy complejo e intervienen gran cantidad de factores, desde la fuerza con la que los pulmones impulsan el aire pasando por las cuerdas vocales, hasta la forma que toma nuestro tracto vocal para canalizar esa excitación o de que manera sale por nuestros orificios nasales y labios.

Por ello, debemos simplificar un sistema que modele lo mejor posible al humano. Empezaremos por cuando nuestro cerebro genera la información necesaria para articular una frase (sería similar a generar un texto fuente para su posterior síntesis). Siguiendo con la figura 2.1, esta frase llevará implícita una serie información para su generación, como es la entonación o pitch, si los fonemas son sonoros o sordos y las características espectrales. Esto se convertirá en una serie de impulsos nerviosos que comunicarán a las cuerdas vocales a que frecuencia deben vibrar en el caso de ser la producción de un fonema sonoro, o por el contrario, permanecer relajadas ante el flujo de aire emitido por los pulmones en el caso de ser sordo, durante un determinado tiempo y con una determinada energía. Las características espectrales dispondrán los músculos del tracto vocal de una determinada forma para que actúe de filtro acústico sobre la onda de presión propulsada desde los pulmones y generar así la señal de voz que saldrá a través de la boca, resonando a su vez por las fosas nasales. Estas características espectrales se modelarán como si fuera un

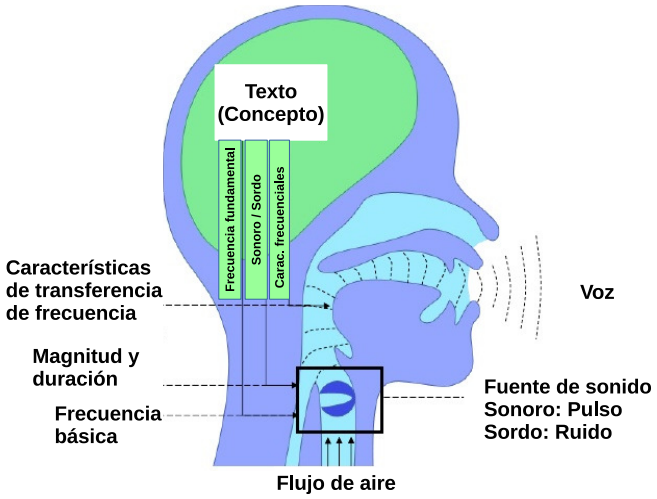

Figura 2.1: Proceso de producción de la voz

filtro con unos determinados coeficientes espectrales.

2.1.2. Excitación

Las cuerdas vocales cuando vibran con una cierta periodicidad (si el fonema es sonoro) es gracias al flujo de aire generado por los pulmones y son "afinadas" gracias a la posición muscular de los músculos de la garganta. Esto es lo que denominamos excitación. Si el fonema es sordo, las cuerdas vocales permanecen relajadas por lo que el aire pasa a través de ellas de forma aleatoria sin ninguna periodicidad. Esto se puede ver en la figura 2.2 donde una función continua va representando la curva de los valores sonoros que toma el pitch, mientras que una función discreta nos dice cuando el pitch es sonoro o sordo, creando una sucesión de ceros en tiempo entre las curvas sonoras.

La frecuencia fundamental en la que vibran las cuerdas vocales dependiendo de su longitud y tensión, es llamada pitch. El pitch determina la entonación con la que se está generando un fonema. Cada persona tiene diferentes cuerdas vocales en cuanto a longitud, nivel de tensión y funcionalidad por lo que esto le da un valor característico y personal para cada una. La frecuencia fundamental para los hombres suele oscilar entre 50 y 250 Hz, mientras que para las mujeres y niños lo normal es ser más alta, entre 120 Hz y 500 Hz.

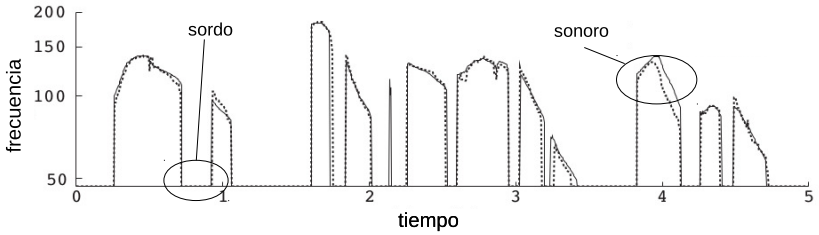

Figura 2.2: Evolución del pitch frente al tiempo.

2.1.3. Parámetros espectrales

El tracto vocal se puede puede modelar como un filtro acústico que modifica el espectro de la onda acústica que lo atraviesa. Normalmente se puede modelar como un filtro AR (autoregresivo) con todo polos y esto genera una distribución espectral dada por una sucesión de frecuencias de resonancia que llamamos formantes. Los formantes serán entonces quien caractericen el tracto vocal y serán diferentes para cada persona.

Para una mejor representación y manejo de los formantes se trabajará con el concepto del cepstrum (o mel-cepstrum si trabajamos en la escala mel, la cual se asemeja a la respuesta del oído humano). El cepstrum es una transformación no lineal que transforma la convolución en suma de secuencias. El proceso de la transformación viene descrito en (2.1), donde a la señal (en este caso, la respuesta del filtro) se le realiza la transformada de Fourier, posteriormente, el logaritmo del valor absoluto, y para finalizar, la transformada de Fourier inversa.

$$x[n] \rightarrow FFT \rightarrow X[k] \rightarrow log|*| \rightarrow log|X[k]| \rightarrow FFT^{-1} \rightarrow c[m] \qquad (2.1)$$

En la figura 2.3 podemos ver como el cepstrum contiene la mayor parte de la información en los primeros coeficientes, los cuales hacen referencia a la envolvente espectral. Tiene también, una serie de picos que nos dan el valor de la frecuencia fundamental de entonación. Si tomamos los primeros valores del cepstrum (hasta antes del segundo pico periódico) y realizamos el proceso inverso para recuperar la forma de la señal, podremos separar la parte de la excitación de la envolvente del tracto vocal.

El cepstrum tiene sus limitaciones, como por ejemplo, en voces con la frecuencia fundamental alta la envolvente espectral aparece como si se hubiera obtenido con una baja resolución frecuencial y la excitación no se puede separar de la envolvente [1]. Sin embargo, por otro lado, podemos obtener mejores resultados que otros métodos como los de predicción lineal (LPC)

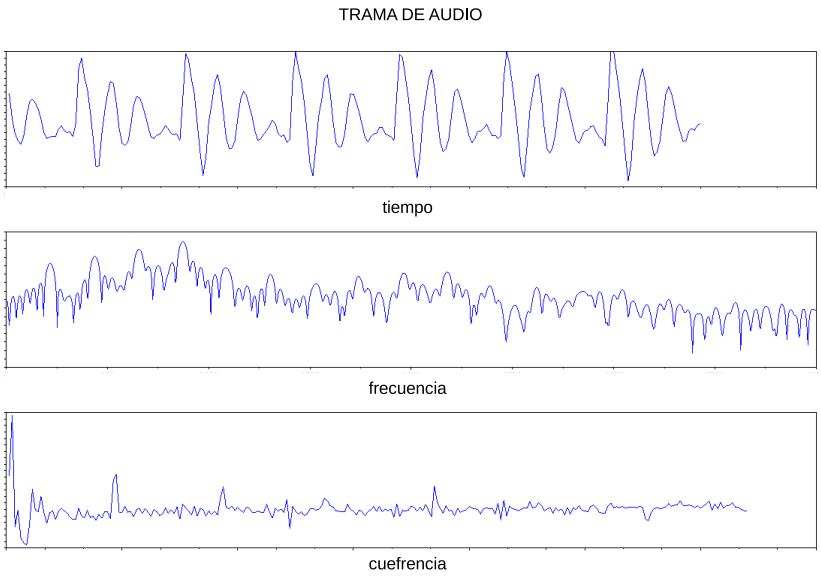

Figura 2.3: Ejemplo de un cepstrum. Arriba una señal en tiempo, en el centro en frecuencia y abajo su cepstrum.

en cuanto a separación de la envolvente y el pitch, y a su vez, muestra una representación espectral con un número bastante menor de coeficientes ya que sólo necesitamos los primeros.

2.1.4. Sistemas de síntesis de voz

Partiendo de los conceptos que hemos visto anteriormente de excitación y parametrización espectral se puede crear un sistema que nos permita sintetizar voz. El sistema se basará en la generación de una señal (excitación) que se tomará como entrada a un filtro (parámetros espectrales) obteniendo como resultado una voz sintética. A la hora de generar o tratar un audio, trabajamos con tramas pertenecientes a un conjunto de una señal de audio que ha sido troceado en ventanas. Por lo tanto, trama a trama, la variable discreta del pitch determinará si debemos generar un tren de impulsos o ruido blanco. Esto dependerá de si la trama a generar es sonora o sorda, tal como muestra la figura 2.4. En el caso de que sea sonora, la variable continua del pitch nos dirá con que periodicidad debemos crear las deltas del tren de impulsos, siendo la distancia entre las deltas la inversa de la frecuencia fundamental. Tanto el tracto vocal, como la radiación producida en los labios y la resonancia de las cavidades nasales, están tomadas en cuenta en los coeficientes espectrales del filtro.

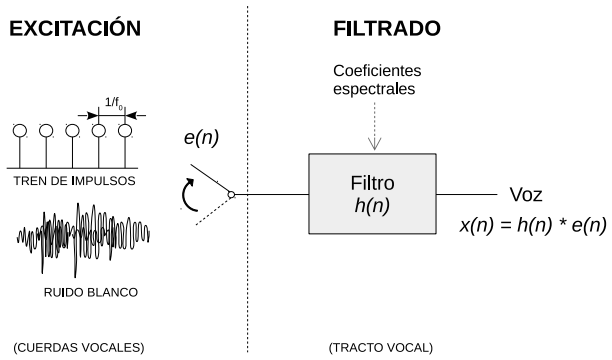

Figura 2.4: Modelado de una señal de voz

En resumen, se puede generar una voz sintética partiendo de una excitación que simule el flujo de la onda de presión acústica y un filtro que modele la respuesta frecuencial que sufre esta onda hasta ser radiada. Si conseguimos que las componentes de este sistema se asemejen al de la producción humana de unos determinados locutores, la voz sintética guardará relación con la del locutor en concreto, y dependiendo de la calidad de la síntesis, se podrá reconocer como si del mismo locutor se tratara.

2.2. Uso de Modelos Ocultos de Markov

Los Modelos Ocultos de Markov (HMMs, Hidden Markov Models) es una de las formas más usadas para modelar la producción de voz. Gracias a ellos, podemos dotar de una nueva estructura a nuestro sistema para modelar el habla de los locutores. Este sistema de modelado nos da la posibilidad de parametrizar las características que componen el habla y crear un estadístico capaz de definir la identidad propia de un locutor. Dicho estadístico se obtiene en base a las probabilidades de ocurrencia de un conjunto de unidades sonoras dentro una extensa base de datos.

Esto abre un gran abanico de posibilidades ya que trabajando con procesos estocásticos en vez de deterministas se está mucho menos limitados, pudiendo crear una locución con la identidad de un determinado locutor simplemente basándonos en su modelo estadístico. Además, estos modelos nos brindan la oportunidad de trabajar directamente con parámetros en vez de con audio.

2.2.1. Estructura

Los HMMs parten de las cadenas de Markov, donde podemos encontrar una serie de estados con una probabilidad de transición entre ellos (a_{ij}) y con posibles observables resultantes de cada uno (o_k) con una determinada probabilidad de salida ($b_i(o_k)$). En el apéndice A se puede ver un ejemplo de cadena de Markov para su mejor entendimiento.

En pocas palabras, lo que se quiere conseguir es buscar la sucesión de observables que maximice la probabilidad de ocurrencia, siendo estos observables el conjunto de características extraídas de cada trama de audio resultantes de cada estado. Por ello, se tiene en cuenta la probabilidad de transición entre estados junto con la probabilidad de salida de los observables dentro de cada estado.

Para obtener el estadístico de estas probabilidades que dan forma al modelo, se debe analizar una base de datos de un locutor que contenga audio con sus respectivas transcripciones. Estos audios se trocearán en pequeñas tramas y se agruparan por fonemas. Cada fonema tendrá una sucesión de estados, cada estado estará formado por una sucesión de tramas y a cada trama irá asociado un observable. Una vez realizado el análisis de toda la base de datos de dicho locutor se podrán establecer unas probabilidades de transición entre estados y unas de salida de cada observable.

Viéndolo gráficamente en la figura 2.5, el audio de un fonema es troceado en un número de tramas y a su vez, estas tramas se corresponden con los observables. El fonema tendrá asociada una secuencia de estados "q" y cada estado la posible salida de unos determinados observables. Finalmente, tras el análisis de varios fragmentos se obtendrán tanto las probabilidades de transición entre estados como las probabilidades de salida de cada observable dentro de cada fonema.

2.2.2. Transiciones

El Modelo Oculto de Markov viene definido por $\lambda = (\underline{\underline{A}}, \underline{\underline{B}}, \pi)$, donde $\underline{\underline{A}}$ es la matriz que contiene las probabilidades de transición entre estados, $\underline{\underline{B}}$ la matriz con los estadísticos de cada observable y π el conjunto de las probabilidades de estar inicialmente en cada estado del modelo [2].

Siguiendo con la figura 2.5, podemos ver que en este ejemplo están representados un HMM de 3 estados principales (más el de entrada y salida). π nos da la probabilidad del estado inicial, mientras que las probabilidades a_{ij} indican la probabilidad de pasar en la siguiente trama del estado i al j,

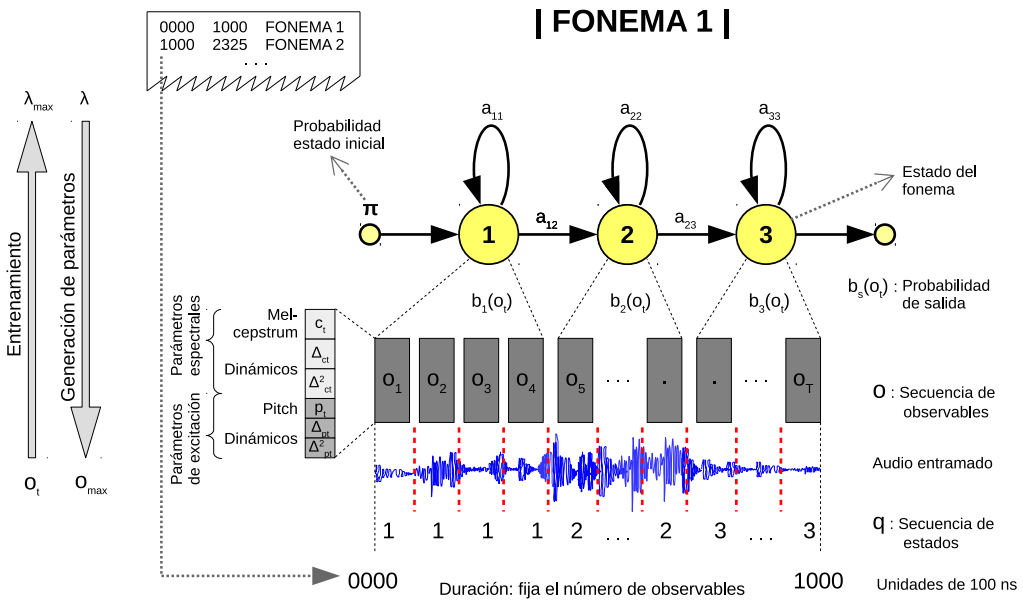

Figura 2.5: Ejemplo de HMM de 3 estados

o lo que es lo mismo, la probabilidad de que el observable de la siguiente trama pertenezca al estado siguiente. En el caso de que i sea igual que j, se hace referencia a la probabilidad de permanecer en el mismo estado, por lo que el siguiente observable pertenecería al estado actual. Cada vez que permanecemos o entramos en un estado nuevo se deberá extraer un observable en función de una determinada probabilidad de salida con distribución Gaussiana.

Así pues, dependiendo de las duraciones de cada fonema, se establecerá una secuencia de estados que determinará en que estado se encuentra cada trama y la secuencia de observables que se referirá al observable correspondiente para cada trama.

2.2.3. Características de los observables

Los tres puntos importantes a la hora de modelar una voz son: la información espectral, el pitch y las duraciones.

Los modelos pueden ser continuos o discretos. En nuestro caso, trabajando con voz, tomaremos modelos continuos para los observables puesto que este tipo de señales puede tomar cualquier valor dentro de un dominio infinito

y la probabilidad de los observables serán también continuas. Deberemos añadir a esto, para el caso del pitch, una variable discreta que diferencie entre excitación sonora o sorda como ya hemos visto en en la figura 2.4.

La información espectral queda modelada en los coeficientes cepstrum a través de la ecuación (2.1). Como ya vimos anteriormente, estos coeficientes representan el espectro de cada trama con la ventaja de que dan mayor resolución a bajas frecuencias, al igual que el oído humano.

En cuanto a la duración, esta hace referencia a lo que dura cada fonema, y con ello condiciona el número de tramas que lo forman, dando más tarde lugar a lo que será la secuencia de estados (ver figura 2.5). Por lo tanto, el modelo de duración estará condicionando también los modelos de pitch y coeficientes espectrales puesto que les dicta cuanto tiempo deben estar emitiendo en cada estado, generando tantas tramas como sean necesarias para alcanzar el tiempo establecido por el modelo de duración.

Los modelos que obtendremos serán de tipo "izquierda-derecha", siendo la única transición posible entre estados adyacentes. La figura 2.5 refleja perfectamente como está relacionada la estructura de un HMM desde la señal de audio hasta la correlación con cada estado para cada fonema.

Una vez troceada en tramas de la misma duración la señal de voz (normalmente usando un sistema de solapamiento), cada una de estas tramas contendrá la información de cada observable. Esta información esta contenida en un vector cuyas tres primeras componentes son referentes a los parámetros espectrales y las tres segundas a los parámetros de excitación. Los parámetros espectrales son los mel-cepstrums y sus dinámicos, mientras que los parámetros de excitación son el pitch y sus dinámicos. Los dinámicos se pueden entender como la primera y segunda derivada del coeficiente estático principal correspondiente desde un punto de vista de función temporal. Estos valores se usan para conocer si el próximo valor será mayor o menor que el actual (dependerá de si el valor del primer dinámico es mayor o menor que cero) y tendrá una tendencia dependiente del valor del segundo dinámico. Esto tiene su uso para condicionar que los coeficientes estáticos empleados en la síntesis mantengan una evolución lo más natural posible entre tramas adyacentes [3].

2.3. Aproximaciones al problema

Si bien hemos hablado en la introducción de este proyecto sobre el origen del Voice Morphing, siendo este a base de pequeños fragmentos de audio

extraídos de una grabación para su posterior reordenamiento con la finalidad de obtener una secuencia de fonemas de acuerdo a un texto, ahora se va a explicar un nuevo concepto, concibiendo la conversión de voz como una técnica que busca modificar la voz de un locutor fuente para que suene como si de un locutor objetivo se tratara.

Desde este nuevo punto de vista, en el actual estado del arte, se ha tratado de interpolar las dos voces para mantener la identidad fonológica entre dos uterancias[1] de diferentes locutores. Es decir, buscar una función que realice la transformación de cada unidad mínima que componen la voz origen para que tengan las mismas características que identifican a un locutor objetivo. Al escuchar dicha transformación debe sonar como si del locutor objetivo se tratara, o al menos en una cierta proporción entre el fuente y este. En este aspecto, esta aplicación se ha usado, por ejemplo, para crear voces peculiares en películas de animación.

Se debe prestar especial atención a los diferentes factores que van implícitos a la hora de hacer la conversión de voz. Ellos son el pitch o excitación y las características espectrales (explicadas en profundidad en la sección 2.1), la duración de cada fonema y su energía. Por lo que la voz resultante debe tener la misma duración y curva de pitch que la locución emitida por el locutor fuente (sujeta a una interpolación de acuerdo al pitch del locutor objetivo), mientras que la repuesta frecuencial debe ser la misma que el locutor objetivo. También se debe intentar que la energía (la cual está dada por el primer coeficiente del cepstrum) sea la misma que la del locutor fuente para intentar guardar la misma dinámica.

La estructura utilizada para la conversión de voz, sujeta a modificaciones con el paso del tiempo, siempre ha sido similar. Como se puede ver en la figura 2.6, partiendo de la locución de una misma frase, tanto del locutor objetivo como del locutor fuente, se les extrae el pitch y se interpola ayudado con algoritmos como DTW (Dynamic Time Warping) [4], el cual permite medir la similitud entre dos series que pueden variar con el tiempo y así poder interpolar el pitch alineando las duraciones de ambos, creando una curva "intermedia" entre las curvas fuente y objetivo de pitch. En la figura 2.7 se puede ver un ejemplo. También se suele aplicar factores de conversión, con el porcentaje de voz fuente y objetivo deseado. Posteriormente, se realiza un análisis de cada audio, normalmente LPC, para extraer la señal de excitación y los coeficientes del filtro. Debido a la propiedad del cepstrum con la

[1] Uterancia : A la hora de analizar lenguaje hablado, la uterancia es la mínima unidad del habla. Es una pieza continua, empezando y acabando con una pausa o silencio. No existen para el lenguaje escrito, sino sólo su representación, la cual se puede dar de diferentes maneras.

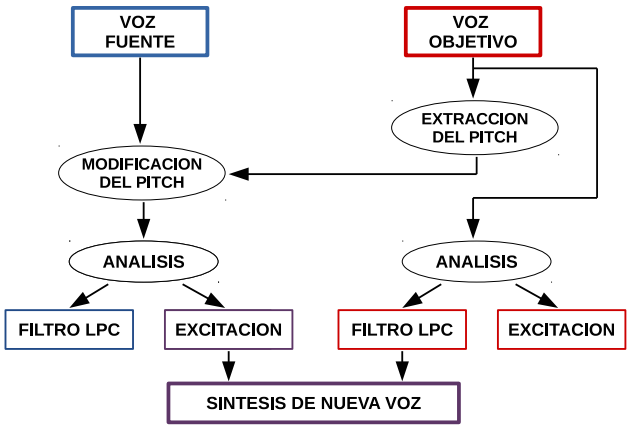

Figura 2.6: Estructura común utilizada para conversión de voz

cual se puede convertir la convolución de dos señales temporales en suma de cepstrums, es posible sumar los cepstrum del filtro LPC del locutor objetivo con los de la excitación del fuente para obtener la voz sintética transformada [5]. Hay que recalcar, que necesitamos las locuciones de ambos locutores usando este sistema (cosa que no ocurrirá con el sistema que posteriormente propondremos), además de que se está trabajando a nivel de procesado de audio en vez de modificar una serie de parámetros, lo cual queremos evitar.

Desde la década de los noventa varias técnicas de conversión de voz han sido propuestas. Una bastante satisfactoria fue usar un método estadístico de mapeo desde un locutor fuente a un objetivo en el dominio del cepstrum, pero tenía el inconveniente de que las formantes eran discontinuas debido a la no linealidad entre estas y la respuesta frecuencial a largo del tiempo. Por ejemplo, cuando una potencia espectral tenía un formante f_{1-a} y el otro f_{1-b}, la interpolación entre ellas daba espureos en dichas frecuencias modificando las características fonológicas. Para evitar este deterioro y la discontinuidad en las formantes, se propusieron varias soluciones. Una de ellas fue un método basado en modelado AR-HMM [6], donde el tracto vocal se modela como un filtro AR y para las cuerdas vocales se crea un modelo basado en HMMs. El mapeo realiza la función de conversión como un modelo basado en la mezcla ponderada de Gaussianas (GMM). Primeramente se realiza una fase de entrenamiento donde se analizan muestras con el mismo contenido fonético para ambos locutores, después se realiza un alineado con un algoritmo DTW para compensar las diferencias en duración entre las uterancias de los locutores, y finalmente, se estima una función estocástica de conversión independiente para el tracto vocal y para el modelo de cuerdas vocales.

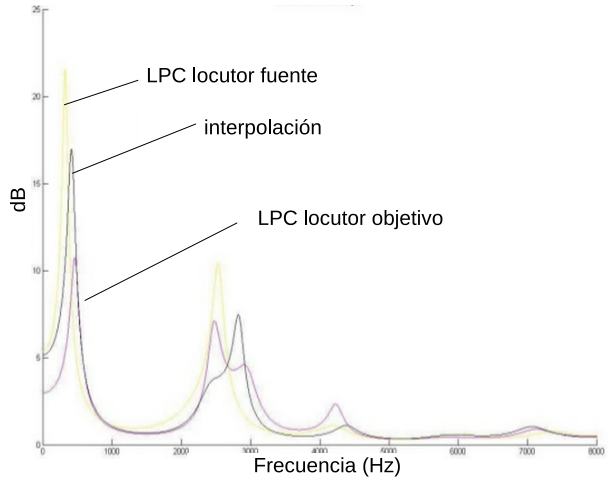

Figura 2.7: Ejemplo interpolación de formantes

Posteriormente, en la fase de conversión o morphing, no se requiere de la locución del locutor objetivo como en los sistemas anteriores, sino que se analiza la locución del locutor fuente para después usar la función de conversión con una interpolación lineal. Esta interpolación se realiza del modo $(1 - \sum_{k=1}^{n} \eta_k)x + \sum_{k=1}^{n} \eta_k F_k(x)$ siendo x la fuente original, $F_k(x)$ la función de conversión del locutor k y η_k el ratio de conversión. A la hora de sintetizar, se genera primero la excitación a partir de los cepstrums adaptados de las cuerdas vocales y más tarde es filtrado con los coeficientes AR del tracto vocal para una síntensis LPC.

Por otro lado, los resultados más satisfactorios que se han encontrado en cuanto a la conversión de voz, han sido realizados a través de la interpolación lineal de la función de transferencia del tracto vocal en escala logarítmica, tanto directa como con el uso de los polos derivados de la aproximación polinómica de la función [7].

La Universidad de Cambridge es una de las pioneras en el estudio de este tipo de conversión, donde a través de la búsqueda de un tipo de transformación lineal se pretende suavizar los efectos de las no linealidades que derivan en incoherencias de fase, coloreado espectral en fonemas no sonoros o en armónicos heredados del locutor fuente al locutor objetivo [5][8].

Se ha observado que para conseguir las funciones de transformación, se puede llegar a los mismos resultados con estimadores de máxima verosimilitud sin depender de la disponibilidad de entrenamiento en paralelo, que con estimadores de mínimo error cuadrático apoyados con un entrenamiento, lo cual le da bastante flexibilidad [9] y es un buen tema de estudio.

Actualmente, aunque se han conseguido muchas mejoras, no se han logrado resultados con la calidad suficiente como para tener una conversión de voz de alta fidelidad como la que se puede requerir en un estudio profesional.

En este proyecto se buscará un nuevo enfoque que no dependa de una función de transformación. Gracias a conocer los parámetros característicos de cada locutor a través de los modelos estadísticos, se reconocerán los fonemas del audio del locutor fuente para su posterior sintetización basada en la caracterización de los locutores.

Capítulo 3

Descripción de la solución

3.1. Descripción General

Después de estudiar el estado del arte y los actuales sistemas, se quiere desarrollar un sistema desde un nuevo punto de vista a la hora de realizar la conversión de voz. Todos los anteriores sistemas se basan en buscar una función de transformación entre el locutor fuente y el objetivo, con el problema que conlleva en la mayoría de los casos la no linealidad de la transformación, la necesidad de tener la locución del locutor objetivo o una extensa base de datos.

Por ello, en este proyecto se busca crear un sistema que utilice todo el potencial y ventajas que nos brindan los Modelos Ocultos de Markov. Es decir, un sistema que no necesite de la locución del locutor objetivo ya que tendrá una total caracterización paramétrica de este y que no dependa de la calidad de una función de transformación.

Este sistema, tras el entrenamiento de los HMMs de cada locutor, no necesitará las bases de datos de ellos, si no que será suficiente con los modelos de cada uno, los cuales no requieren más que de unas pocas decenas de MBytes.

Como se puede ver en la figura 3.1, no necesitaremos una función de transformación que actúe directamente sobre la señal del locutor fuente, sino que se extraerán los parámetros que caracterizan a un audio concreto del locutor fuente y se pasará de voz a texto (Speech-To-Text).

Gracias a los HMMs se conocerán los parámetros que caracterizan estadísticamente al locutor objetivo y se mezclarán con los extraídos del locutor fuente tomando lo que nos interese de cada uno (duraciones y pitch del

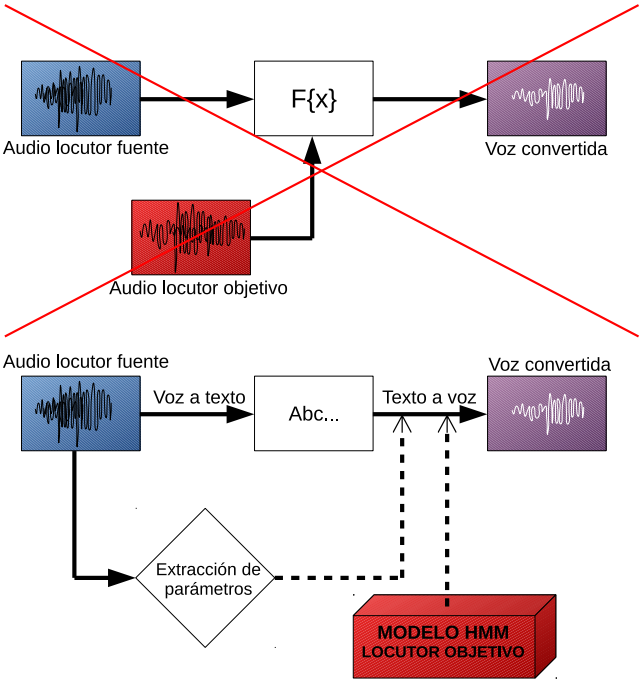

Figura 3.1: Actuales sistemas de conversión de voz (arriba) y el sistema que se plantea desarrollar (abajo)

locutor fuente y parámetros espectrales del locutor objetivo) para generar con ellos la síntesis del texto (Text-To-Speech). Lo que quiere decir, que esta síntesis estará condicionada por estos parámetros característicos resultantes para que suene con la identidad del locutor objetivo pero con la entonación, pausas y dinámica extraída del audio del locutor fuente.

Dichos parámetros hacen referencia directa a los estadísticos resultantes del entrenamiento de los HMMs de cada locutor, por lo que se propone extraerlos del locutor fuente, modificar aquellos los cuales sean necesarios adaptar de acuerdo al locutor objetivo (como por ejemplo el caso del pitch) y generar una voz a partir de dichos parámetros a través de los estadísticos del locutor objetivo. La estructura del sistema se mostrará mas adelante en la figura 3.4 con una descripción mas detallada.

3.2. Obtención de los modelos

Se parte de una base de datos de cada locutor para realizar el entrenamiento, las cuales deben contener audio con las locuciones de cada uno y sus

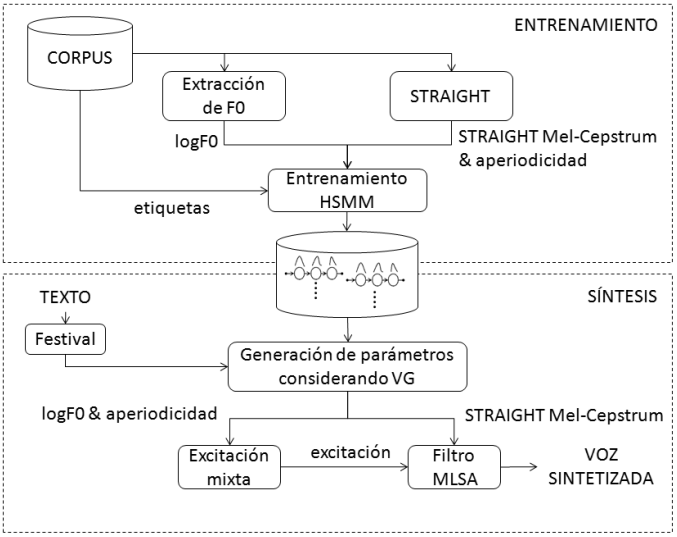

Figura 3.2: Descripción de un posible sistema para modelado por HMM [14]

respectivas transcripciones. Durante el entrenamiento se crearán los modelos estadísticos de acuerdo con lo que se va a explicar en la siguiente sección 3.2.1,y por lo cual, una vez finalizado este, ya no será necesario disponer de las bases de datos y la herramienta no necesitará de mucha capacidad de almacenaje para realizar la conversión de voz. Resaltar que una voz puede ser entrenada en un ordenador u otro tipo de dispositivo y posteriormente los archivos generados con el modelo estadístico pueden ser transferidos a otros dispositivos sin la necesidad de que estos gasten tiempo y memoria en realizar esta labor.

3.2.1. Entrenamiento

Para realizar el entrenamiento la base de datos (o corpus) deberá estar formada por audio del locutor objetivo, al cual se le quiere clonar con la voz sintetizada[1], y a su vez, la locución debe estar correctamente transcrita en archivos de texto para poder hacer la relación. Además, el audio debe estar etiquetado determinando los segmentos temporales que corresponden a cada fonema. Actualmente existen diferentes softwares que nos ayudan a realizar esta tarea.

[1]También es necesario una base de datos del locutor fuente para poder entrenar el sistema y así realizar el reconocimiento, pero este proyecto se basa más en la síntesis y no tanto en el análisis.

Si observamos la figura 3.2, del corpus se extrae el pitch (en este caso se trabaja con el logaritmo de la frecuencia del pitch) y los coeficientes mel-cepstrum. En base a esto y a las transcripciones, se entrena el modelo a través de algoritmos recursivos que buscan acotar cada vez mejor los estados dentro de cada fonema. A su vez, con arboles de decisión, se van relacionando parámetros que dependen del contexto con otros similares. Esto se debe a que con una base de datos limitada no se pueden estimar con precisión y de manera robusta todos los parámetros que dependan del contexto [10].

$$\underline{\text{Entrenamiento:}} \quad \lambda_{max} = \arg \max_{\lambda} p(O|\lambda) \tag{3.1}$$

$$siendo \quad p(O|\lambda) = \sum_{q} \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1},q_t} b_{q_t}(O_t) \tag{3.2}$$

En la figura 2.5 vemos bien reflejado como el entrenamiento se realizaría yendo de abajo a arriba, como indica la flecha, partiendo del audio entramado con su posterior parametrización hasta llegar a los estadísticos.

En resumen, lo que se busca en esta fase de entrenamiento es obtener los modelos λ que maximizan localmente la probabilidad de los observables dados (3.1). Para el desarrollo de la herramienta que queremos construir no es necesario profundizar más en los conceptos teóricos, ya que disponemos de herramientas previas (HTS) donde ya esta desarrollado estas partes. HTS utiliza la fase de entrenamiento de HTK [11].

3.3. Entradas del sistema

3.3.1. Reconocimiento del locutor fuente.

Una vez obtenidos los estadísticos de los Modelos Ocultos de Markov, sólo requeriremos de una entrada de audio con su transcripción del locutor fuente, (por supuesto, no es necesario que el audio provenga de la base de datos, ni que las frases de este sean las mismas que se entrenaron). Aunque también existe la posibilidad de realizar el reconocimiento sólo con el audio, sin necesidad de tener la transcripción, en este proyecto se le ayudará con la transcripción para facilitar el proceso de reconocimiento y obtener mejores resultados. En una línea futura, se podría desarrollar un reconocedor mejor (esto podría ser motivo de otro nuevo proyecto) del que se dispone en este proyecto y se podría hacer sólo con el audio.

El objetivo de la realización del entrenamiento del locutor fuente se debe a que necesitamos conocer su modelo para después poder realizar el correcto

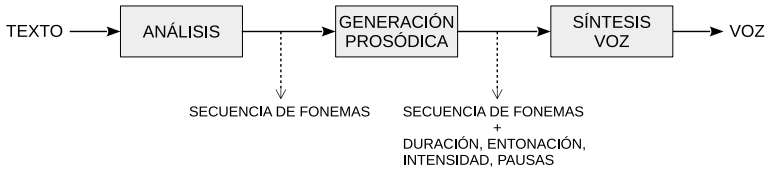

Figura 3.3: Sistema de conversión de texto a voz

reconocimiento de la locución. El audio será enventanado en tramas para su análisis, y en conjunto con la transcripción, gracias al modelo obtenido del entrenamiento podremos obtener los parámetros espectrales, las duraciones de cada estado (y por consiguiente la de los fonemas) y el pitch del locutor fuente.

3.3.2. Generación del locutor objetivo.

Cabe remarcar una vez más que no se necesitará dar como entrada del sistema un audio del locutor objetivo, ya que que esta puede ser generada (solamente generando los parámetros que lo caracterizan es suficiente, por lo que no haría falta llegar a generar el archivo de audio como tal). Por ello, con la transcripción del locutor fuente podemos generar los parámetros del locutor objetivo basándose en su modelo estadístico. Tanto el pitch, como las duraciones o los parámetros espectrales serán creados en base a las probabilidades del modelo. En otras palabras, estaremos generando unos parámetros "medios" con los que obtendríamos la voz del locutor objetivo diciendo la frase transcrita de la forma más probable que él lo diría según el modelo obtenido en el entrenamiento.

Basándonos en la síntesis de texto a voz (Text-To-Speech) vamos a generar los parámetros del locutor objetivo, como se puede ver en la figura 3.3. Partiendo de un texto, tras su análisis se puede sacar la secuencia de fonemas. Posteriormente, con esta secuencia de fonemas y gracias al estadístico de nuestro modelo ya entrenado, será posible añadir más información a cada fonema como es la duración, la entonación, intensidad o pausas. Este conjunto de características es lo que llamamos prosodia.

Esta vez, siguiendo el ejemplo de los Modelos Ocultos de Markov de la figura 2.5 la generación de parámetros (los cuales componen los observables) sería recorrer el sentido de la flecha de arriba a abajo, partiendo del modelo estadístico, hasta llegar a los observables con los que generaríamos las tramas de audio. Así pues, partiendo de la secuencia de fonemas, y basándose en las probabilidades de transición del modelo, se obtendrá una secuencia de

estados que dará como resultado la duración de los fonemas (esta duración será desechada, ya que la que interesa es la del audio del locutor fuente).

Según las probabilidades de salida de cada estado, se obtendrán unos observables que contendrán los parámetros como la entonación (pitch), el tracto vocal y la intensidad que vendría dada por los cepstrums y la energía de estos (la cual va contenida en el primer coeficiente mel-cepstrum) y las pausas que irían determinadas en la secuencia de fonemas entre la sucesión de fonemas vocales y los declarados como silencios.

$$\underline{\text{Generación de parámetros:}} \quad o_{max} = p(o|q, \lambda_{max}) \tag{3.3}$$

$$\text{con distribución de} \quad b_i(o_t) = \mathcal{N}(o_t; \mu_i, \Sigma_i) \tag{3.4}$$

A la hora de generar los parámetros (o lo que es lo mismo, obtener los observables de salida de cada estado), a la inversa de la fase de entrenamiento, se busca hallar el observable $\underline{o} = (\underline{o_1}, \underline{o_2}, \underline{o_3}, ..., \underline{o_T})$ que maximiza la probabilidad para una secuencia de estados fija $\underline{q} = (q_1, q_2, q_3, ..., q_T)$ y un modelo λ, tal cual describe la ecuación (3.3).

A la hora de establecer las probabilidades, en los HMMs esto se puede hacer con una distribución Gaussiana o con una suma de varias Gaussianas. Es cierto que con una densidad de probabilidad basada en una mezcla de Gaussianas (suma de Gaussianas ponderadas por unos pesos) se puede obtener una mejor resolución en los cepstrums. Se ha comprobado que, por ejemplo, hay una mejora apreciable con la mezcla de ocho Gaussianas [12], pero con el modelo de HTS que tenemos sólo se usan dos ejemplos, uno con una única Gaussiana y otro con dos. Como no se ha apreciado mucha diferencia, a partir de ahora siempre se hablará de una densidad de probabilidad de una Gaussiana.

También se deberán tener en cuenta los parámetros dinámicos asociados a los coeficientes estáticos, tanto del pitch como de los cepstrums. Gracias a estos parámetros conseguiremos una evolución en el espectro de modo progresivo y sin saltos bruscos a lo largo del tiempo. Esto generará en su escucha una voz más natural, con una menor aparición de sonidos "artificiales" o de "voz robotica".

3.4. Conversión

A continuación, lo que se propone es intercambiar los parámetros de ambos locutores, tomando los que nos interesen de cada uno. Para generar una señal de voz que suene como si el locutor objetivo la hubiera grabado pero

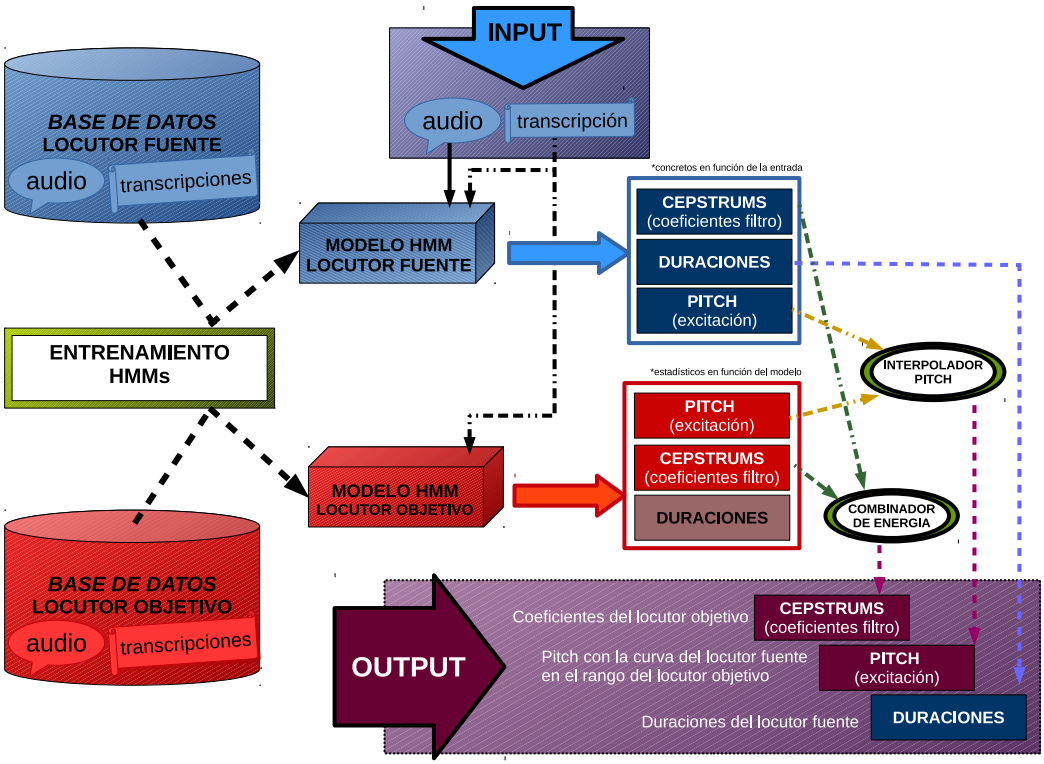

Figura 3.4: Descripción del proceso

con la misma entonación, duraciones, pausas y dinámica que la locución del audio dada por el locutor fuente, necesitamos la curva de pitch, duraciones y curva de energía del locutor fuente y los parámetros espectrales del locutor objetivo, tal como indica la figura 3.4.

Para trabajar con las características espectrales usaremos el cepstrum. Las duraciones del locutor fuente serán tomadas directamente sin ninguna modificación para la síntesis de voz final, en cambio, tanto la energía de los cepstrum como el pitch deberán ser modificados.

Para mantener la misma energía en cada trama, se sustituirá el primer coeficiente cepstrum del locutor objetivo por el primero del locutor fuente en cada una de ellas (combinador de energía, figura 3.4). En este primer coeficiente recae la mayor parte del peso de la energía que tendrá cada trama mientras que el resto de coeficientes colorean la señal de excitación. Con ello, se conseguirá tener en cada trama un tracto vocal equivalente al del locutor objetivo pero con la energía del locutor fuente.

En cuanto al pitch, la modificación consistirá en adaptar la curva del pitch del locutor fuente respecto a los valores de la curva del locutor objetivo. Esto

debe ser así puesto que si se quiere crear una voz convincente del locutor objetivo no podemos tomar directamente el pitch del locutor fuente, ya que debe estar adaptado como si del locutor objetivo se tratara. Un ejemplo muy claro sería cuando pretendemos hacer la conversión de voz entre un hombre y una mujer o viceversa, si el hombre tuviera una excitación con frecuencias de mujer, las cuales suelen ser más altas, sonaría irreal y lo mismo en el caso contrario.

Dentro de las diferentes maneras que se pueden utilizar, se implementarán dos tipos de interpolación en este proyecto y se juzgará cual tiene mejor resultado. Una de ellas será ajustando la curva del locutor fuente dentro de los margenes del rango de frecuencias donde se mueve el locutor objetivo y otra trasladando la media del locutor fuente a la media del objetivo (interpolador de pitch, figura 3.4).

3.5. Síntesis

Como se ha visto en la sección 2.1, para realizar la síntesis de voz, vamos a seguir el esquema de la figura 2.4. Una variable discreta estipulará si hay que generar una excitación sorda o sonora, asignándole una periodicidad concreta al tren de deltas de acuerdo al pitch que le corresponde a esa trama en el caso de que sea sonora o ruido blanco si es sorda. Para cada trama, al igual que con el pith, también se le aplicará un filtro con los coeficientes espectrales del observable resultante correspondiente. Finalmente, el filtrado de la excitación nos dará como resultado la generación de una trama de voz.

En conclusión, tras el entrenamiento, este sistema requeriría únicamente de una entrada de audio del locutor fuente y su transcripción. Con ello obtendríamos las duraciones a nivel de estado del locutor fuente y los cepstrums del locutor objetivo. Estos serían utilizados para configurar los coeficientes del filtro a emplear siguiendo el modelo de creación de voz por excitación-filtrado. Para finalizar, se tomaría la excitación resultante de la interpolación del pitch de ambos locutores como entrada para el filtro.

Capítulo 4

Metodología e implementación

4.1. Herramientas

Para usar los toolkits como HTK, Festival, SPTK y HTS, se ha preferido llevar el proyecto bajo la plataforma Linux ya que la compatibilidad y manejo de estos sistemas se hace de una manera más sencilla y fiable. Además de esto, se crearon una serie de funciones tanto en C o Matlab, como en Unix-Shell.

4.1.1. HTS

Debemos tener en cuenta que, aunque en nuestro caso queremos hacer una conversión de voz a voz, siempre que esto lleva implícito una fase de síntesis, se está hablando de tomar como entrada un texto y generar una voz sintética como salida, lo que comúnmente es conocido como TTS (Text-To-Speech). Aquí entra en juego HTS.

HTS es un sistema de síntesis de voz basado en Modelos Ocultos de Markov. Este sistema es de código abierto y ha sido desarrollado por el Instituto Tecnológico de Nagoya (Japón) y con la colaboración de la Universidad Carnegie Mellon (Estados Unidos). HTS fue creado en 2002 a partir de la herramienta HTK (Universidad de Cambridge, Reino Unido), siendo un parche de esta. Su propósito fue el de facilitar el estudio y la investigación de la síntesis de voz a partir de HMMs. Es un proyecto en constante mejora en el cual aún se sigue trabajando.

Por ejemplo, se ha escogido este sistema de síntesis por HMMs frente al método de concatenación de forma de onda, porque aunque con este método se pueden obtener una voz con apariencia más natural, suele tener transi-

```
      0  1100000 X^X-#+o1=l@x_x/A:0_0_0/B:x-x-x@x ...
1100000  2200000 X^#-o1+l=a@1_1/A:0_0_0/B:1-0-1@1- ...
2200000  3300000 #^o1-l+a=l@1_2/A:1_0_1/B:0-0-2@2- ...
3300000  4400000 o1^l-a+l=u@2_1/A:1_0_1/B:0-0-2@2- ...
```

Figura 4.1: Etiquetado de la palabra "hola" precedida de silencio en HTS. Formato HTKLabel. En este ejemplo la duración se muestra a nivel de fonema pero también se puede dar a nivel de estado.

ciones defectuosas entre los fragmentos que pudieran dar lugar a una mala inteligibilidad. Además, este sistema suele ser irregular en cuanto al nivel de calidad y requiere una base de datos de gran tamaño en la memoria de nuestro dispositivo puesto que tiene que almacenar todos los fragmentos de audio. Otra ventaja de los HMMs es que podemos modificar parámetros como el pitch o la velocidad con sólo modificar unos coeficientes, mientras que con la concatenación de fragmentos este proceso es mucho más cerrado, ya que tendríamos que procesar los fragmentos. Es por ello que una herramienta como HTS es perfecta para nuestras necesidades, además de ser el sistema más utilizado en cuanto a Modelos Ocultos de Markov.

Sin profundizar demasiado en la arquitectura de HTS, este usa una serie de funciones (HInit, HERest, HHEd) con las que va creando los modelos a partir de la base de datos durante el entrenamiento de forma iterativa y recursiva hasta alcanzar una convergencia. Usa una estructura de ficheros donde va almacenando la información de los observables que extrae de las tramas. Las tramas son tomadas a través de un enventanado deslizante (pudiendo ser Blackman, Hamming o Hanning) y con la posibilidad de asignar una longitud y desplazamiento deseado. De cada trama se obtiene un observable donde se almacenan en diferentes ficheros, según su tipología, la frecuencia fundamental y sus dinámicos, y los coeficientes cepstrum (25 en nuestro caso) con sus respectivos dinámicos.

HTS busca la forma de onda de una voz concreta a base de recorrer una serie de estados dentro de cada fonema. En este proyecto usaremos cadenas de 7 estados (5 estados principales más el de entrada y de salida).

Al final de la fase de entrenamiento se van generando una serie de etiquetas[1] donde se ven reflejados los fonemas con los posibles contextos. Como se puede ver en la figura 4.1 cada línea da la información de la duración de cada fonema, cuales son los dos fonemas anteriores y los dos siguientes, y el contexto en el que se encuentra (a través de un formato propio de la herramienta).

[1]Llamaremos etiqueta a los archivos (.lab) que contienen la transcripción de los fonemas con la duración de sus estados y el contexto de cada uno (ver figura 4.1).

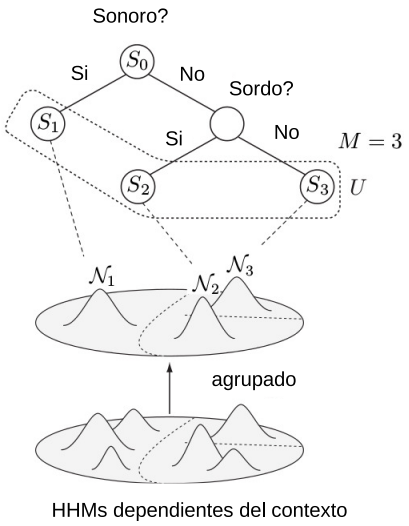

Figura 4.2: Ejemplo de árbol de decisión

La duración va dada según el estándar utilizado en estas aplicaciones, en unidades de 100 ns, e impone el número de tramas que tendrá cada fonema como se puede ver en la figura 2.5.

Posteriormente, va guardando una lista de todos los fonemas con los posibles contextos que pueden tener y más tarde va relacionando los fonemas que se pueden considerar iguales. Los fonemas pueden venir en varios contextos, y esto quiere decir que se van acotando en función de si ese fonema guarda relación con él mismo en otros contexto o no, pudiendo asociar el mismo fonema con varios contextos en uno sólo. Al final, con toda esta información crea unos estadísticos de los estados y las probabilidades de salida de cada observable a través de unos árboles de decisión (figura 4.2), siguiendo el número de índices que recorre en cada árbol para cada estado [10].

HTS también realiza tareas a través de otras herramientas como SPTK o Festival.

Como punto de partida se estudió la demo dada con HTS para entrenar la base de datos CMU-Artic-SLT de la Carnegie Mellon University. A través del estudio detallado de este procedimiento se pudo entender como trabajaba HTS y se extrajeron varias funciones para su posterior utilización en nuestra herramienta[2].

[2]Recordar que todas las herramientas y toolkits que se han usado son de código abierto, por lo que se puede hacer uso de ellas para crear nuevas herramientas sin incurrir en ningún problema legal.

Una de las funciones más importantes que se extrajeron fue HMGenS, como se verá en el punto 4.3.

4.1.2. SPTK

Como su nombre indica (Speech Procesing ToolKit) es una herramienta para procesado de señales de voz, la cual interviene sobretodo en HTS en las partes donde se necesitan procesar las tramas de audio.

En este proyecto se han usado diferentes funciones de SPTK para el desarrollo del mismo. Varias de las funciones creadas para este proyecto incorporan llamadas a funciones de SPTK, como el caso en las funciones GetCepstrum y GetLogFO a la hora de extraer la frecuencia fundamental de las tramas del audio del locutor fuente y los cepstrums.

La fase de síntesis, una vez obtenidos los cepstrums y la frecuencia fundamental de cada trama resultantes de la conversión de voz, esta basada en las funciones que SPTK lleva incorporadas para generar la excitación y hacer el filtrado. Toma el modelo de excitación-filtro para generar un archivo de audio, tomando trama a trama como excitación dicha frecuencia fundamental resultante y el filtro basado en los coeficientes cepstrum también resultantes de dicha conversión.

4.1.3. Festival

En un sistema de síntesis de voz (Text-To-Speech) independiente, desarrollado por la universidad de Edimburgo, con el cual HTS realiza funciones de etiquetado entre otros.

Se puede ver presente en funciones como txt2utt donde pasa el contenido en un archivo de texto a un formato uterancia. En la función utt2lab basada en scripts de HTS, este a su vez utiliza herramientas de festival para obtener la transformación de un archivo de uterancias a formato HTKLabel que puede entender HTS (ver 4.1)

4.2. Duraciones

Para la conversión de voz vamos a tomar como entrada un audio del locutor fuente y su transcripción. Este audio será de tipo crudo o binario (.raw) y la transcripción vendrá dada en un documento de texto (.txt). En correlación

con la figura 3.4 y siguiendo la 4.3, del audio se extraerá la frecuencia fundamental de cada trama y se realizará el reconocimiento de la locución. Este reconocimiento se podrá realizar gracias al previo entrenamiento y dará como resultado un archivo (.rec) que contendrá la secuencia de fonemas reconocidos en la locución y la duración de cada uno de sus estados. El inconveniente de este reconocimiento es que nos entrega los fonemas con una nomenclatura diferente a la que usaremos con HTS.

En este proyecto se ha trabajado con varios formatos de transcripciones o etiquetas los cuales ya venían supeditados a cada toolkit. Por ello, para conseguir la etiqueta (.lab) donde se realizará la imposición de las duraciones (ver formato de etiqueta (.lab) en la figura 4.1), necesitamos algunos pasos intermedios para generarla (ver figura 4.3). El texto de la transcripción se necesita convertir a un fichero tipo utt el cual contiene las uterancias de la transcripción.

Para realizar esta labor nos hemos apoyado en la herramienta Festival la cual gracias a una base de datos en español de la Universidad Politécnica de Madrid, puede convertir textos (.txt) a uterancias (.utt). Desde el fichero de uterancias, de nuevo con Festival, podemos convertirlo en etiqueta. Esta etiqueta está en formato HTKLabel, el cual puede ser entendido por HTS y contiene las duraciones a nivel de fonema y la secuencia de estos con su contexto.

Por lo tanto, por un lado tenemos un archivo (.lab) con la secuencia de fonemas etiquetados con su contexto y duraciones basadas en el estadístico del locutor objetivo, y por otro lado un archivo (.rec) con las duraciones a nivel de estado específicas del audio del locutor fuente. Aquí entra en juego la función MixDur que se ha creado para mezclar estos dos archivos.

El propósito de la función es crear un archivo (.lab) que haga la relación entre las duraciones a nivel de estado extraídos del audio del locutor fuente con las etiquetas que contienen cada fonema con su contexto, desechando las duraciones generadas estadísticamente de este último (ver apéndice B). Finalmente, el archivo (.lab) final es generado a base de alinear los fonemas de cada archivo (con diferentes nomenclaturas en cada uno) y copiar las duraciones de cada estado provenientes del fichero (.rec) del locutor fuente, seguidas de las etiquetas (.lab) previas con la información de cada fonema y su contexto.

El resultado sería así una etiqueta (.lab) con la información necesaria de cada fonema y su contexto con las duraciones deseadas de cada estado y en el formato que HTS puede entender para su síntesis.

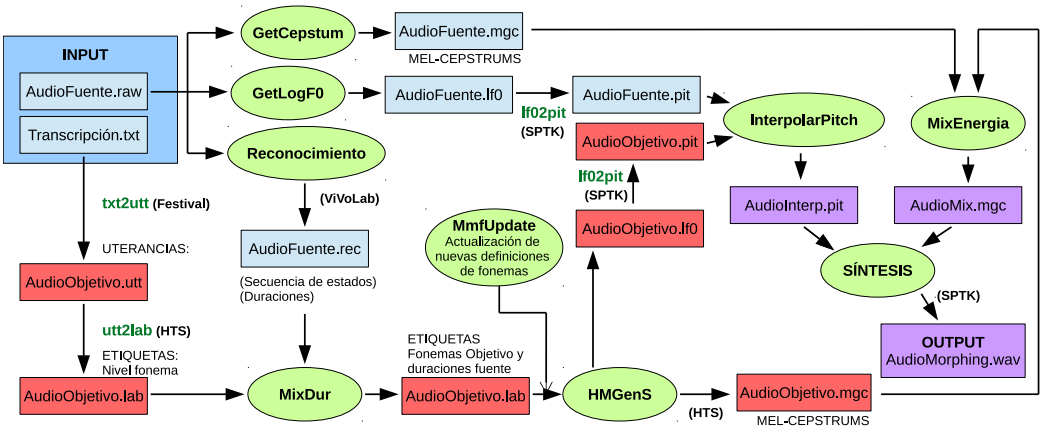

Figura 4.3: Esquema secuencial entre funciones y archivos

4.3. Generación de parámetros: HMGenS

La función HMGenS de HTS ha sido de gran relevancia ya que a partir de una etiqueta se puede obtener dos archivos con la información necesaria para poder sintetizar directamente la voz referente al contenido de dicha etiqueta de acuerdo con los estadísticos del locutor previamente entrenado [13].

Como se puede ver en la figura 4.3, dando como entrada la etiqueta (.lab) con la imposición de las duraciones del locutor fuente a la función HMGenS, se obtendrán los archivos con los coeficientes mel-cepstrum (.mgc) y el logaritmo de la frecuencia fundamental del pitch (.lf0) de cada trama, basándose en el modelo estadístico del locutor objetivo.

El archivo (.lf0) con la información del pitch del locutor objetivo se utilizará para adaptar la curva de pitch del locutor fuente, mientras que el archivo (.mgc) con los parámetros espectrales se mezclará con los del fuente para conservar la energía de este último.

4.4. Pitch

Tal y como decíamos al principio del punto 4.2, se extrae la frecuencia fundamental de las tramas del audio del locutor fuente a través de un script llamado GetLogF0 creado a base de funciones de SPTK. Posteriormente, se almacena esta información en un fichero (.lf0) con el formato del logaritmo de la frecuencia fundamental, que es con el que SPTK los extrae. Más tarde, en cuanto a la sintetización, SPTK necesitará de un nuevo formato, así que

finalmente habrá que reconvertir estos ficheros en archivos con del tipo (.pit) pitch, los cuales almacenan el número de muestras del periodo del pitch para cada trama.

En cuanto a los parámetros del locutor objetivo, volviendo a la figura 4.3 y según lo visto en la sección 4.3, gracias a la funcion HMGenS, partiendo de la etiqueta (.lab) generada en el punto anterior, podemos obtener los ficheros con los logaritmos de la frecuencia fundamental (.lf0) y los coeficientes mel-cepstrum. Ambos ficheros son generados con las características dadas por el modelo estadístico del locutor objetivo, excepto la secuencia de estados que ha sido forzada por las duraciones del locutor fuente. Por lo tanto, será necesario al igual que antes, convertir el archivo con la información tonal (.lf0) basada en el estadístico del locutor objetivo al formato requerido para la síntesis (.pit).

Una vez que ya tenemos los ficheros con la información del pitch de ambos locutores, debemos interpolarlos para obtener una curva valida para el locutor objetivo sin que suene fuera de su tonalidad. Para ello, se han creado diferentes versiones de la función InterpolarPitch (ver apéndice B), la cual busca recrear la curva del locutor fuente dentro del rango de frecuencias del locutor objetivo.

4.4.1. Interpolación del pitch por media

La forma más sencilla de adaptar la curva del pitch del locutor fuente dentro de las frecuencias que utiliza el locutor objetivo es trasladando la media del locutor fuente a la media del locutor objetivo.

$$\underline{\text{Traslación aditiva:}} \quad pitch_{interpolado} = pitch_{fuente} - \mu_{fuente} + \mu_{objetivo} \quad (4.1)$$

Esta traslación puede ser aditiva (ecuación (4.1)) si a el valor del pitch del locutor se le resta su media y se le suma la del objetivo. Fácilmente se puede despejar que al restar la media a todos los valores del pitch del locutor fuente, lo que se esta haciendo es normalizar los valores de este con media igual a cero para posteriormente sumarle a todos los valores la media del pitch del locutor objetivo por lo que la media resultante sería la de este último. En este caso el rango y la variabilidad de la curva del pitch sería exactamente igual a la del locutor fuente pero centrada en la media del locutor objetivo. En contraposición, este método puede darnos problemas si la curva del locutor fuente dispone de una desviación típica muy grande, pueden existir valores que estén fuera de las posibilidades tonales del locutor objetivo.

En este proyecto, el sistema de archivos trabaja con los valores de las

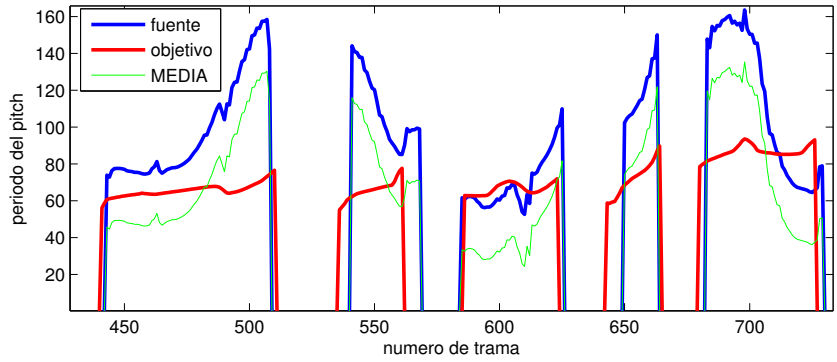

Figura 4.4: Interpolado del pitch por traslación aditiva teniendo en cuenta la media del periodo (representado en el domino del periodo, [número de muestras])

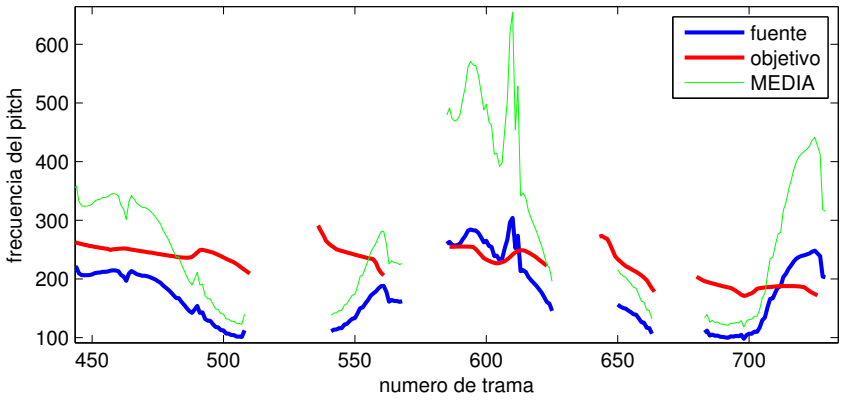

Figura 4.5: Interpolado del pitch por traslación aditiva teniendo en cuenta la media del periodo (representado en el domino de la frecuencia, [Hz])

muestras del periodo del pitch, por lo tanto, a primera vista, lo más sencillo sería trabajar con el periodo. En cambio, el oído humano es más sensible a los cambios en frecuencias. Teniendo esto en cuenta, finalmente se aplicará el algoritmo de interpolación trabajando tanto con la media de la frecuencia como con la media del periodo para comprobar auditivamente estos resultados y decidir que método es mejor.

Este razonamiento tiene una base teórica debido a que los resultados de la interpolación aditiva trabajando con el periodo no son lineales cuando estos son pasados a frecuencia. Esto quiere decir, que mientras que la forma de la curva del pitch se respeta en el dominio del periodo, al pasar este a frecuencia, siendo una transformación inversamente proporcional, no se respeta la linealidad de la traslación. Obteniendo asi como resultado una curva del pitch en frecuencia con la media adaptada a la del locutor objetivo, pero con un escalado diferente de la original. Esta diferencia se puede observar en las figuras 4.4 y 4.5, respetándose únicamente en la primera la forma de la curva del locutor fuente.

Al otro tipo de traslación de media lo llamaremos "proporcional" y vendrá dado por la ecuación (4.2). Lo que se consigue con este método es que al trasladar la media del fuente a una media objetivo menor, la proporción del rango es menor. O viceversa, si se traslada de la media del fuente a una media objetivo mayor, la proporción será mayor.

$$\underline{\text{Traslación proporcional:}} \quad pitch_{interpolado} = pitch_{fuente}\frac{\mu_{objetivo}}{\mu_{fuente}} \tag{4.2}$$

Este concepto cobra sentido basándose en que cuando el habla se compone de frecuencias más altas, el rango dinámico tiende a ser mayor y viceversa cuando las frecuencias son menores. Además, a diferencia del anterior, con este método la traslación es lineal se trabaje con el periodo del pitch o con la frecuencia, ya que se hace de manera proporcional en vez de aditiva. Por lo tanto, los resultados son similares para ambos dominios.

4.4.2. Interpolación del pitch por rango

Para realizar este tipo de interpolación se debe analizar el fichero (.pit) del locutor objetivo con el fin de extraer el rango de frecuencias en el que este locutor se mueve de modo estadístico y el máximo y mínimo pitch que hay en la curva del locutor fuente para conocer el rango de este. El siguiente paso es expandir o contraer la curva del locutor fuente dentro del rango del locutor objetivo, de modo que el mínimo de la curva del locutor fuente corresponda

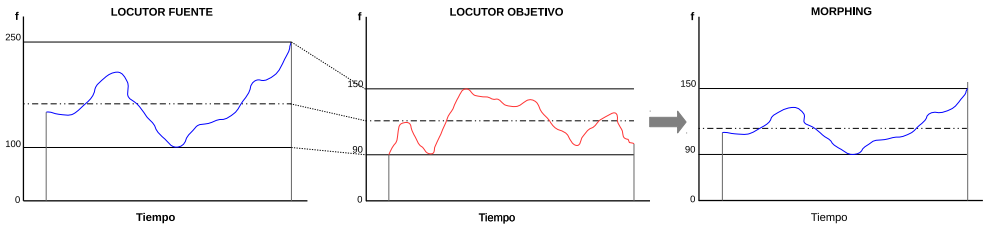

Figura 4.6: Interpolado del pitch por rango

con el mínimo del locutor objetivo y lo mismo con el máximo. De este modo, se consigue la misma curva extraída del locutor fuente, con la mismas subidas y bajadas de entonación, pero moviéndose dentro del rango de frecuencias del locutor objetivo, tal como muestra la figura 4.6.

Matemáticamente se debe extraer los rangos (margenes dinámicos) de cada locutor, lo que es igual a la resta de su valor mínimo al valor máximo del pitch. Para saber la relación α que guarda el rango de un locutor con otro, siguiendo la ecuación (4.4) se debe dividir el rango del locutor objetivo entre el del fuente. Posteriormente, se le debe restar el valor mínimo del pitch del locutor fuente al valor de este para situar el mínimo valor de su rango en cero. Después, se multiplica por α para restablecer el margen dinámico con la misma apertura que el rango del locutor objetivo. Para finalizar, el valor del pitch interpolado se obtendrá sumándole el valor mínimo del pitch objetivo para trasladar el margen dinámico obtenido previamente a los valores del pitch objetivo tal cual muestra la ecuación (4.3).

$$pitch_{interpolado} = pitch_{min_{objetivo}} + \alpha(pitch_{fuente} - pitch_{min_{fuente}}) \tag{4.3}$$

$$siendo \quad \alpha = \frac{\Delta_{objetivo}}{\Delta_{fuente}} \quad y \quad \Delta = pitch_{max} - pitch_{min} \tag{4.4}$$

Esta transformación es la más conservadora, ya que aseguramos que todos los valores están dentro de la tonalidad del locutor objetivo. Pero en contra, estaremos perdiendo la variabilidad dinámica original del locutor fuente ya que la curva de este se vera comprimida o expandida según el rango del locutor objetivo.

Sabiendo de antemano que para calcular el rango dinámico del locutor objetivo se partirá del pitch generado por el modelo estadístico de este para la frase concreta a convertir, este margen no tendrá un amplitud demasiado grande ya que este pitch se generará con una entonación neutral.

Para compensar esto a la hora de realizar la conversión de voz teniendo en cuenta que con emociones la amplitud de este rango varía considerablemente,

se probará también una interpolación por "rango proporcional" donde el pitch mínimo del locutor objetivo se resituará un tanto por ciento del rango por debajo o por arriba y lo mismo con el pitch máximo. A base de jugar con el porcentaje del rango a tener en cuenta se buscará que la interpolación del pitch entre dos locutores concretos no pierda la variabilidad dinámica original del locutor fuente y a su vez que estos valores estén adaptados dentro de la tonalidad del locutor objetivo.

4.5. Parámetros espectrales

Gracias nuevamente a la función HMGenS y avanzando con lo comentado en la sección 4.3, dándole como entrada el archivo (.lab) con la secuencia de fonemas con su contexto proveniente de la transcripción y las duraciones a nivel de estado extraídas del audio del locutor fuente, hemos podido generar el fichero con las frecuencias fundamentales y los coeficientes mel-cepstrum de cada trama basándonos en el modelo estadístico del locutor objetivo. Los coeficientes para cada trama han sido creados de acuerdo a las duraciones impuestas por el audio del locutor fuente. Este fichero (.mgc) contiene toda la información necesaria para la síntesis y el formato es el correcto. En él se incluyen los 25 coeficientes en la escala mel-cepstrum asociados a cada trama.

4.5.1. Energía

Para la posibilidad de mantener trama a trama la curva de energía del locutor fuente, necesitamos obtener la energía de este. Como bien se ve reflejado en la figura 4.3, se ha creado la función GetCepstrum que genera un archivo (.mgc) con los coeficientes mel-cepstrum extraídos del audio del locutor fuente.

La energía de cada trama vendrá determinada por el primer coeficiente cepstrum, por lo que se deberá sustituir en los cepstrum del locutor objetivo el primer coeficiente por los del locutor fuente. Para ello, se ha creado la función MixEnergia. Esta función lee los primeros coeficientes de cada trama del archivo (.mgc) obtenido en la extracción de parámetros a través de la funciones GetCepstrum y con ellos sobrescribe los primeros de cada trama del archivo (.mgc) del locutor objetivo obtenido con la función HMGenS.

Como resultado se generará un archivo (.mgc) con la mezcla de los coeficientes mel-cepstum que respetaran la curva de energía del locutor fuente

teniendo a su vez, la identidad característica del tracto vocal del locutor objetivo.

4.6. Síntesis

La síntesis se realiza a través del script "Sintetizar" creado en base a la herramienta SPTK, en el cual se pueden elegir los parámetros con los que queremos realizarla, como la frecuencia de muestreo, tipo de ventana, tamaño y desplazamiento, etc. Estos parámetros deben guardar relación con los utilizados a la hora de entramar el audio en el entrenamiento, ya que durante todo el proceso tanto observables como tramas iban secuenciados según el tamaño dado por estos valores.

Tal como se puede ver en la figura 4.3, la función necesita como entrada los ficheros (.pit) con el pitch y (.mgc) con los coeficientes mel-cepstrum que hemos generado y explicado en los puntos anteriores. Finalmente, nos generará un archivo (.wav) de audio con el resultado de conversión de voz.

Con esto queda constatado una vez más en la metodología llevaba a cabo, que durante todo este proceso no ha hecho falta recurrir al audio del locutor objetivo diciendo la misma frase como en otros sistemas. En su lugar, partiendo de la transcripción del locutor fuente hemos generado los parámetros necesarios para modelar la voz del locutor objetivo gracias a su estadístico y, posteriormente, realizar la conversión de voz entre ambas.

4.7. Bases de datos

4.7.1. Albayzin

Puesto que se quería realizar este proyecto con voces en español, se recurrió a bases de datos en español. Así pues, se tomaron las de Albayzin, que contiene bases de varios locutores con un corpus de casi 600.000 palabras en total con una distribución fonética equilibrada, y su correspondiente transcripción [14]. De todos los locutores que incluye, se entrenaron tres (dos voces femeninas y una masculina) con la finalidad de realizar la función de locutores objetivo. Estas bases de datos cuentan con 200 audios para cada locutor y han sido grabados con una entonación neutral (plana), o lo que es lo mismo sin que la prosodia de estos muestre una determinada emoción.

Con una base de datos de cada locutor del orden de 200 audios, se pueden

lograr voces inteligibles y con la identidad del locutor, pero la calidad deja entrever que se trata de una voz sintética con un sonido un tanto "robótico". Para intentar compensar esto en cierta medida se intentará entrenar el sistema con los audios de todos los locutores que componen el corpus de Albayzin (del orden de unos 300) como si de un solo locutor se tratara. Este entrenamiento consta del orden de más de 6.000 audios y se espera conseguir una voz intermedia de los 300 locutores con una identidad propia característica y una mejor calidad que los locutores por sí solos entrenados con 200 audios.

4.7.2. SEV-Joaquín

Para realizar la función del locutor fuente se va a emplear la base de datos del corpus SEV (Spanish Expresive Voices) con voces expresivas en español como su propio nombre indica, la cual fue desarrollada por el grupo de tecnologías del habla de la Universidad Politécnica de Madrid. Este corpus ha sido desarrollado con el propósito de investigar en el marco del habla con emociones. La base de datos original dispone de un locutor femenino y otro masculino, pero en este proyecto sólo se utilizará la del masculino. [15]

Dispone varias emociones, entre ellas trabajaremos con las de alegre, enfadado, sorprendido y triste. Para conseguir esto, se grabaron los audios de la base de datos varias veces, una vez con cada emoción, siendo del orden de varios centenares de frases grabadas con cada una.

4.8. Emociones

Para poder evaluar mejor el prototipo se partirá de locuciones de la misma frase con diferentes emociones. Esto nos permitirá juzgar mejor si se mantiene la prosodia del locutor fuente en el audio resultante, mientras que la identidad de dicho audio deberá pertenecer al locutor objetivo.

Si se convirtieran unas locuciones de entrada que fueran neutras sería más difícil de comprobar si se corresponde la prosodia ya que esta sería similar siempre. Además, considerando en la evaluación y estudio del sistema la posibilidad de tener entradas con prosodias totalmente diferentes, provenientes de diferentes emociones, se dotará de una mayor robustez a la herramienta, ya que las posibles entradas serán semejantes a las que se nos brindan en la vida real.

4.9. Dificultades durante la implementación

4.9.1. Alineamiento-recononocimiento

Debido a que se deben alinear la secuencia de fonemas generada desde el texto con la generada en el reconocimiento del audio, resultaron varias incompatibilidades. Según la emoción con la que el locutor fuente decía la frase, el reconocedor añadía más o menos silencios (el sistema trabaja como si el silencio fuera un fonema más) tanto al principio como al final o en medio de los fonemas, mientras que la secuencia de fonemas extraída del texto suele relacionarlos con las pausas escritas como comas y puntos.

Puesto que la herramienta a desarrollar en este proyecto no incluye el desarrollo de un reconocedor, si no que este viene dado, se invirtió más tiempo en lo referente a lo que es la conversión de voz en si. Finalmente se encontró solución ayudando al texto a la hora de puntuar las pausas. En una línea futura de avance se podría mejorar la compatibilidad entre el sistema previo de reconocimiento y el prototipo desarrollado.

4.9.2. Actualización de contextos

Según la metodología que impone HTS, cuando la función HMGenS lee las etiquetas (.lab) debe reconocer cada contexto. HTS dispone de unas listas de contextos donde almacena todos los contextos reconocidos durante el entrenamiento. El problema se da cuando se quiere generar una nueva frase, la cual contendrá contextos que no han sido entrenados, y HTS no reconocerá ese contexto como parte de su modelo y HMGenS nos devolverá un error.

Para evitar este error, necesitamos actualizar las definiciones de contextos gracias a la ayuda de la función HHEd de HTS. Para ello se creo la función mmfUpdate (ver 4.3) la cual realiza todos los pasos necesarios para incluir los contextos extraídos de la nueva etiqueta en las definiciones del modelo.

Capítulo 5

Resultados

Se debe tener en cuenta que es complicado juzgar objetivamente la calidad de los resultados. Evaluar según que características de un audio puede ser muy subjetivo para cada persona. La mejor manera de comprobar los resultados se presentará en la defensa de este proyecto cuando se podrá escuchar los audios obtenidos de la conversión de voz.

Aún con ello, al final de este capítulo se hará una valoración de la percepción tanto de la identidad del locutor objetivo, como de la prosodia del locutor fuente percibida dentro de los audios obtenidos.

Para comenzar, a continuación se presentarán resultados más objetivos como son la preservación de las duraciones del locutor fuente y la forma de su curva de pitch y energía, así como que los coeficientes que caracterizan el tracto vocal son los del locutor objetivo.

5.1. Parámetros espectrales y duración

La mejor manera de reconocer las duraciones y la energía es ver su forma de onda a lo largo del tiempo. En la figura 5.1 se puede comparar la forma de onda del audio original del locutor fuente con la conversión de voz al locutor teniendo en cuenta la preservación de la energía o no. En ambos casos se impusieron las duraciones de los fonemas, lo que es fácilmente reconocible si comparamos las formas de onda. Se puede comprobar que todas las pausas y unidades fonéticas están perfectamente alineadas y con la misma duración.

A su vez, se puede ver que cuando aplicamos el algoritmo para que se mantenga la energía a través de sustituir el primer coeficiente cepstrum por el del locutor fuente en cada trama, la dinámica resultante es la misma que el

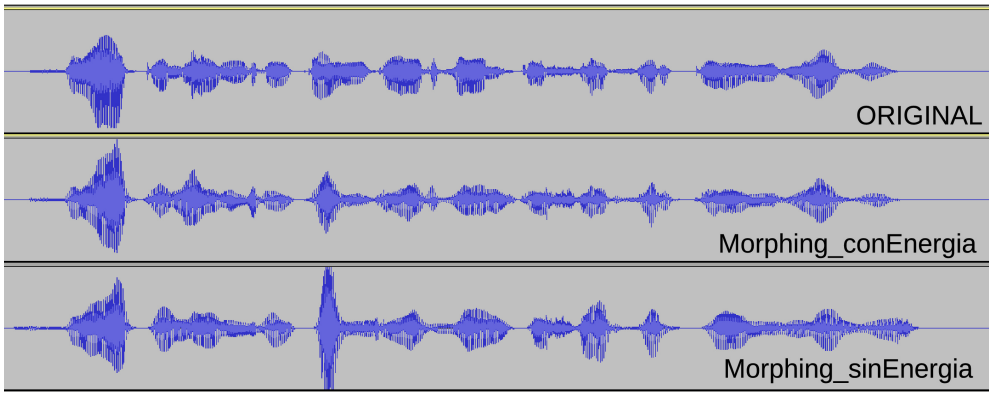

Figura 5.1: Forma de onda del audio del locutor objetivo (arriba), la conversión al locutor objetivo tieniendo en cuenta la energía de los cepstrums del locutor fuente (centro) y sin tenerla en cuenta (abajo)

del audio original. Extrayendo uno de los cepstrums resultantes (figura 5.2) se puede observar claramente como se respeta la energía del locutor fuente con el primer coeficiente y las características del tracto vocal del locutor objetivo con los demás.

Desde otro punto de vista, observando la forma de onda sin tener en cuenta la energía del locutor fuente de la figura 5.1, se ve que dejando la energía en manos del modelo estadístico del locutor objetivo, la dinámica puede tomar formas totalmente diferente a las deseadas.

Entra en discusión por lo tanto, si se prefiere dar algo de libertad a la dinámica que debería tener el locutor objetivo basado en su modelo estadístico para que parte de su identidad sea más reconocible, o por el contrario replicar totalmente la dinámica original del locutor fuente para que la prosodia sea exactamente la misma en cuanto a lo que la dinámica se refiere.

Tras la evaluación de los audios finales, se llegó a la conclusión de que sin tener en cuenta la energía del locutor fuente los resultados eran bastante peores. Por ejemplo, sabiendo que la duración venía impuesta por el locutor fuente, si dejamos la dinámica dependiendo del estadístico del locutor objetivo, este a veces puede dar demasiada energía a sonidos sordos que originalmente para él suelen ser más cortos, resultando en un sonido muy poco natural. Estas irregularidades se podrían interpretar como si se estuvieran estirando en el tiempo un sonido sordo, corto y seco, y por lo tanto no común en el habla.

Después de llegar a esta conclusión, para la obtención de los siguientes resultados siempre se tuvo en cuenta la energía del locutor fuente, siendo mucho

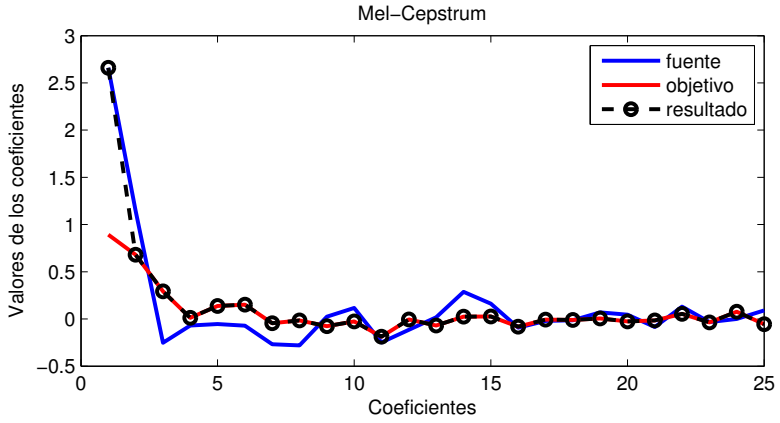

Figura 5.2: Cepstrum teniendo en cuenta el primer coeficiente del locutor fuente

más naturales y sin que ello impidiera que la identidad del locutor objetivo siguiera presente. En cuanto a la preservación de la prosodia del locutor fuente, en lo que a dinámica, pausas y duración se refiere, quedó comprobado que estaba claramente preservada y era totalmente reconocible en los resultados.

5.2. Pitch

El ajuste del pitch ha sido la tarea que más tiempo ha llevado para que sonara con naturalidad y a la vez se preservara tanto la identidad del locutor objetivo como la prosodia del locutor fuente, por ello fue necesario crear diferentes tipos de interpolación (descritos en la sección 4.4).

Los primeros resultados realizando la interpolación por traslación de medias basadas en el periodo no fueron muy buenos ya que debido a la no linealidad, el escalado del pitch en el dominio de la frecuencia estaba totalmente desproporcionado (ver figuras 5.4 y 5.7). Escuchando los audios generados, se constató que es mucho más importante tomar como referencia la curva en el domino de la frecuencia, por lo que el método de traslación por media aditiva trabajando con la frecuencia da mejores resultado que haciéndolo con el periodo.

Como se puede ver en la figura 5.3, haciendo la traslación aditiva los valores que toma el pitch pueden quedar totalmente fuera del rango del locutor objetivo dependiendo del rango de variabilidad del locutor fuente. Esto desencadenaba en unos resultados totalmente fuera de tono y para nada naturales en según que ocasiones.

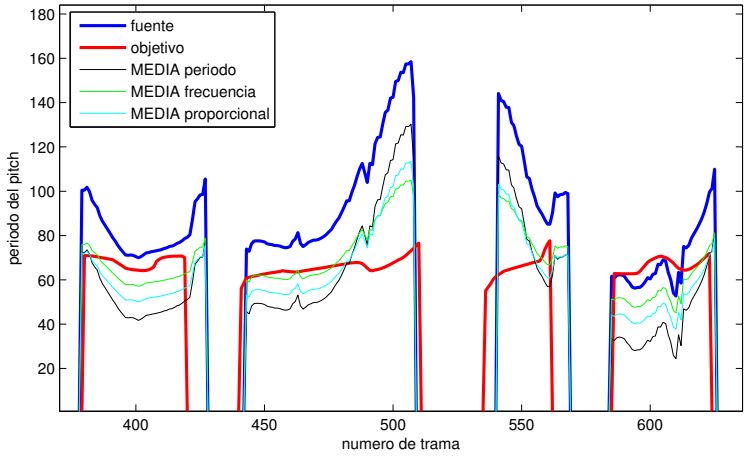

Figura 5.3: Interpolado del periodo del pitch (en muestras) teniendo en cuenta la media por traslación aditiva (MEDIA) del periodo o la frecuencia y por translación proporcional (MEDIA-P)

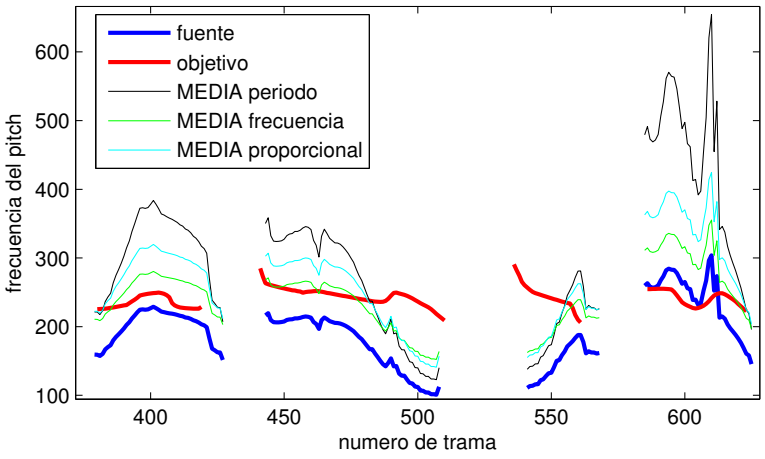

Figura 5.4: Interpolado en frecuencia del pitch (en Hz) teniendo en cuenta la media por traslación aditiva (MEDIA) del periodo o la frecuencia y por translación proporcional (MEDIA-P)

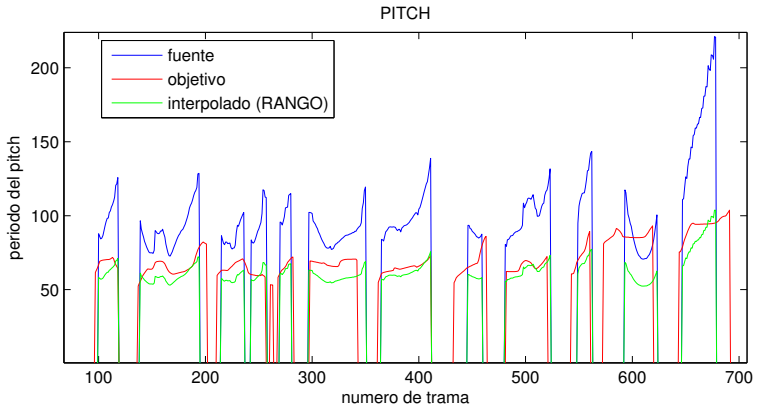

Figura 5.5: Interpolado del periodo del pitch (en muestras) teniendo en cuenta el rango dinámico

Tratando de mejorar los resultados de la media por traslación se realizó el método de interpolación por traslación proporcional de la media. Estos resultados nos daban una mejor aproximación a los valores tonales del locutor objetivo, pero aún así en muchos casos seguían estando fuera de tono en momentos puntuales.

En cuanto a la preservación de la prosodia del locutor, es obvio que interpolando por medias, la curva del pitch guarda la misma dinámica con el mismo rango y por lo tanto queda totalmente preservada. Pero en contraposición, esto no es totalmente compatible con la preservación de la naturalidad del locutor objetivo, ya que los resultados pueden estar fuera de tono cuando el rango del locutor fuente es mucho más amplio que el del locutor objetivo.

Con la intención de mantener una curva de pitch dentro de la tonalidad del locutor objetivo, se realizó la interpolación de pitch por rango. Como ya se sabía, esta opción era la más conservadora y los resultados tuvieron una naturalidad mucho mayor, sin estar fuera de tono. Aunque se podía apreciar la prosodia del locutor fuente, en contrapartida la curva de pitch de este se veía expandida en el caso de que el rango tonal del locutor objetivo fuera mayor que el del locutor fuente, por lo que se exageraba demasiado la prosodia. O el caso contrario, cuando el rango del locutor objetivo era menor que el del locutor fuente se comprimía la curva del pitch de este y la prosodia se veía atenuada (ver figura 5.5).

Para tener la posibilidad de poder jugar con el ajuste del rango, se desarrolló la opción de la interpolación por "rango proporcional". Con este ajuste, podemos darle manualmente el valor del porcentaje del rango tonal del locutor objetivo con el que queremos que varíen los límites del mismo y así con-

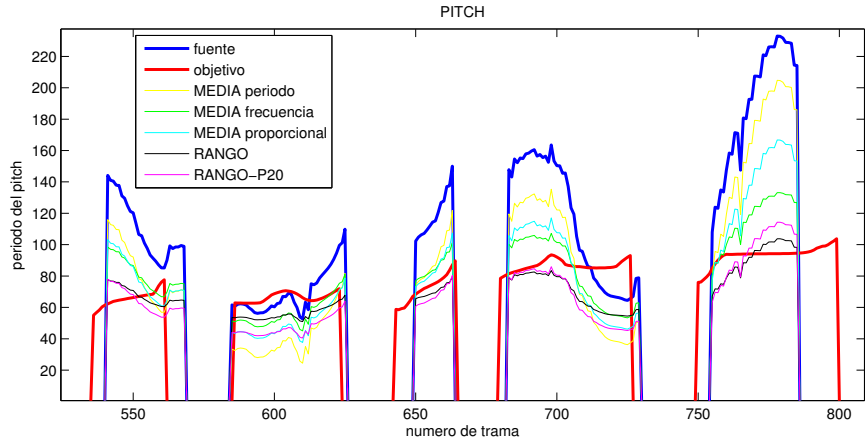

Figura 5.6: Zoom de la interopolación del periodo del pitch (en muestras) de voz masculina a femenina.

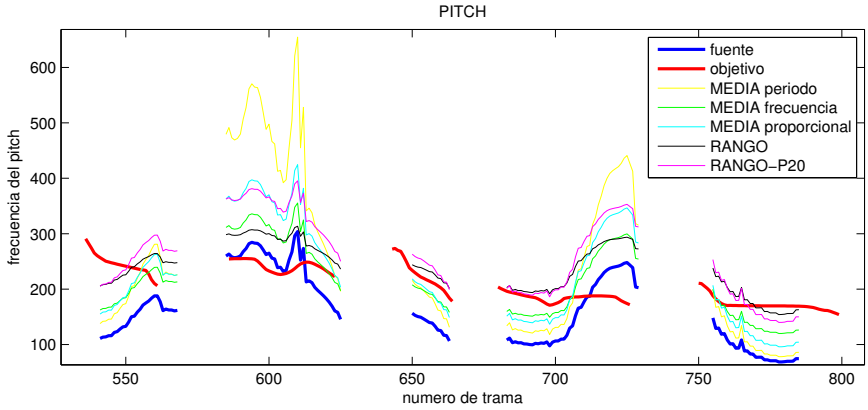

Figura 5.7: Zoom de la interopolación en frecuencia del pitch (en Hz) de voz masculina a femenina.

seguir el punto óptimo entre la preservación de la prosodia del locutor fuente y la identidad tonal del locutor objetivo.

Según se tomaban diferentes ajustes se obtenían varios resultados, unos ganaban en prosodia y otros en naturalidad. La decisión de la toma de unos ajustes u otros venía totalmente ligada a si la conversión de voz se realizaba entre locutores del mismo género (masculino o femenino) y si sus rangos tonales eran parecidos o no.

A continuación se hablará de la comparación de los resultados obtenidos con los cinco métodos de interpolación para la conversión de voz entre locutores tanto del mismo, como de diferente género.

Las figuras 5.6 y 5.7 muestran con más precisión la curva de pitch generada en la conversión de una voz masculina a una femenina para los diferentes métodos. En este caso, la media de la frecuencia del pitch del locutor objetivo es mayor[1] que la del fuente. Al generar los parámetros del locutor objetivo con una prosodia neutral, este presenta un rango tonal más pequeño de lo normal, mientras que el locutor fuente en este caso tiene una prosodia alegre con un rango más amplio. Buscando un punto óptimo en el ajuste del método por "rango proporcional", se amplió el rango un 20 por ciento más alto el nivel máximo y un 20 por ciento más bajo el nivel mínimo. Como se puede observar, a medida que damos más prioridad a la prosodia del locutor fuente, los resultados pierden la identidad tonal del locutor objetivo. Por ello, "objetivamente" se escogerían los métodos que utilizan el rango en vez de la traslación por media en la mayoría de los casos.

En cambio, en la figuras 5.8 y 5.9, donde la conversión de voz se realiza entre locutores del mismo género, estos tienen medias similares, siendo ligeramente menor la media de la frecuencia del locutor objetivo. Seguimos teniendo el mismo problema en cuanto al rango tonal generado para el locutor objetivo que en el caso anterior, por lo que se han utilizado los mismos ajustes el 20 por ciento para el caso del "rango proporcional". En este caso, ambos métodos tienen un pitch aceptable, sin valores fuera de tono si lo vemos en el dominio frecuencial, pero se atenúa un poco la dinámica de la curva del locutor fuente para el caso de la interpolación por rango. En cuanto a las interpolaciones realizadas por media, el caso de la media proporcional al ir de una media mayor a una menor, el rango de la interpolación disminuye respecto del del locutor fuente.

Viendo la evolución de las gráficas a primera vista, "objetivamente" al tratarse de una adaptación menos drástica, ya que los locutores tienen características similares, se podrían tomar cualquiera de los métodos. En cambio, tras escuchar los audios, se pueden observar diferencias y "subjetivamente" se puede apreciar que para conservar la naturalidad de la voz, la interpolación por rango da mejores resultados, aunque en desventaja se puede perder algo de dinamismo de la prosodia del locutor fuente.

En los resultados se ha priorizado en obtener unas tonalidades naturales y propias del locutor objetivo, pero intentado obtener el mayor grado de preservación de la prosodia del locutor fuente. Por ello, dependiendo de la diferencia tonal tanto en rango, como en media de los locutores implicados

[1]En las figuras, el pitch está expresado en función de su periodo, siendo este inverso a la frecuencia. Por lo tanto si la media de la frecuencia del pitch es menor, inversamente la media del periodo será mayor, y viceversa.

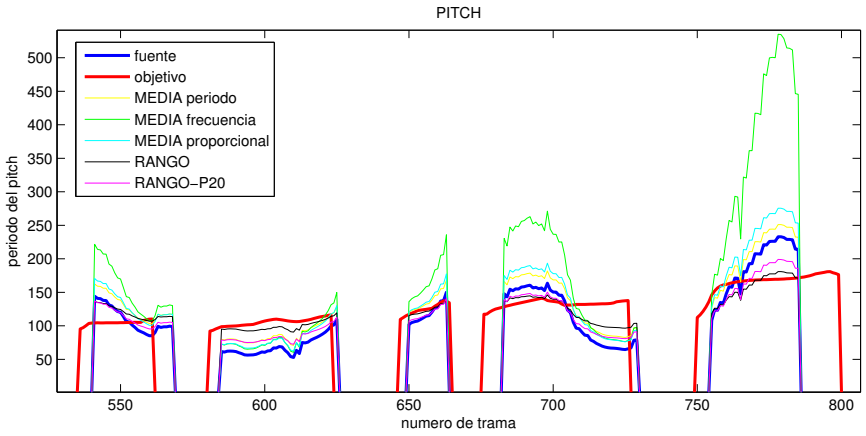

Figura 5.8: Zoom de la interopolación del periodo del pitch (en muestras) de voz masculina a masculina.

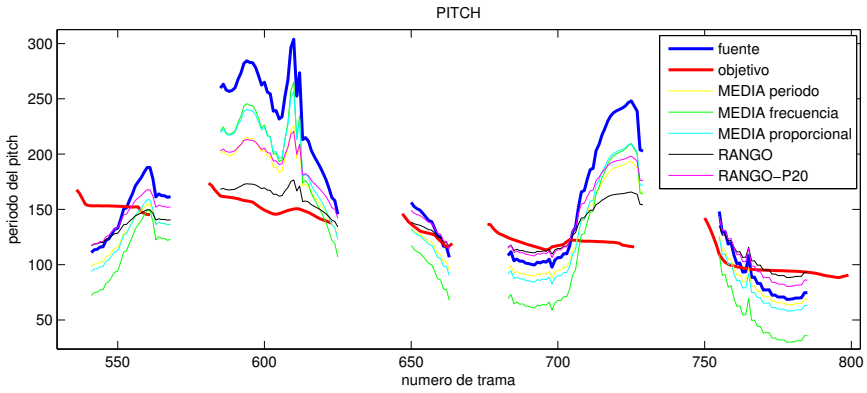

Figura 5.9: Zoom de la interopolación en frecuencia del pitch (en Hz) de voz masculina a masculina.

en la conversión de voz, a veces se han utilizado métodos más conservadores como la interpolación por rango para evitar sonidos fuera de tono, sacrificando un poco la variabilidad de la prosodia del locutor fuente. Y en otros casos, cuando los locutores son similares se puede aprovechar esta prosodia al máximo.

En conclusión, la elección de un método o otro, depende por lo tanto del nivel de prosodia que se está dispuesto a sacrificar para conseguir un mínimo de naturalidad. Por mucho que se intente buscar un método que resuelva el problema con un nivel óptimo entre preservación de prosodia del locutor fuente e identidad del locutor objetivo, esta búsqueda siempre irá ligada a la subjetividad del oyente. También, se debe tener en cuenta que un método puede ser mejor que otro para conversiones de voz entre diferentes tipos de locutores, y a su vez uno mejor que otro dependiendo de la entonación de una frase concreta, por lo que resulta muy difícil poder escoger uno de ellos de modo generalizado.

En cualquier caso, escogiendo el método de interpolación por rango, se puede garantizar la naturalidad de la voz resultante y en la escucha de la mayor parte de los audios se puede reconocer la prosodia del locutor fuente, aunque sea en mayor o menor grado.

5.3. Calidad y evaluación general

Como se ha comentado anteriormente, es muy complicado evaluar la calidad de los resultados de una manera objetiva ya que el reconocimiento de la identidad de un locutor, la calidad del audio, la inteligibilidad del mensaje que se quiere dar o el relacionar una prosodia con otra, depende en gran parte de la subjetividad del oyente que lo está evaluando.

En cuanto a las impresiones que se han obtenido, los resultados son totalmente inteligibles, se puede reconocer perfectamente la identidad del locutor objetivo y se mantiene la prosodia en mayor o menor grado del locutor fuente. Aunque el audio tiene la calidad suficiente, sin ruidos de fondo apreciables, ni añadidos puntualmente por saturaciones o por mala síntesis, si que se puede apreciar una cierta tendencia a lo que se llama “voz robótica”.

Esta “robotización” se puede notar más o menos dependiendo de las características de los locutores, si son más o menos semejantes entre ellos, pero no impiden el reconocimiento tanto de la prosodia del fuente como de la identidad del objetivo, el problema simplemente recae en que dependiendo de este “nivel de robotización” puede quedar en evidencia que esa voz ha

sido generada por ordenador en vez de haber sido grabada por un locutor humano.

Por tomar una referencia, podemos comparar "subjetivamente" la calidad alcanzada en este proyecto con la obtenida en otros casos, como por ejemplo, con las muestras disponibles de la Universidad de Cambridge [16], donde se realizó un proyecto de conversión de voz (sin emociones, voces con entonaciones neutrales) por medio de interpolaciones entre ambos locutores con un resultado que denominaban de alta calidad. Los resultados que hemos obtenido son bastante aceptables en cuanto a la percepción de la identidad del locutor objetivo y la prosodia del fuente, mientras que, por normal general pese a su peor calidad del audio, el efecto de robotización que ellos obtuvieron era menor.

Añadir, como ya se ha comentado en puntos anteriores, que la calidad en la síntesis va estrechamente ligada a la calidad y extensión de la base de datos utilizada. En nuestro caso hemos contado con la base de datos de Albayzin, que no es una base de datos pequeña, pero tampoco es demasiado extensa. Por lo tanto, se puede tener en cuenta que si se hubiera dispuesto de una base de datos mayor, se habrían conseguidos unos mejores resultados.

Capítulo 6

Conclusiones y líneas futuras

6.1. Desarrollo del proyecto

Este proyecto se realizó durante un tiempo "efectivo" de cinco meses los cuales, cuyo cronograma con las tareas desempeñadas viene adjuntado en el apéndice C. El tiempo invertido se puede desglosar en:

- Tiempo dedicado a documentación, tanto a la referente por parte teórica del estudio del estado del arte como a la escritura de esta memoria.
- Tiempo dedicado a la "puesta a punto" necesaria para realizar el proyecto, como era familiarizarse con las herramientas de trabajo, su instalación y correcto funcionamiento.
- La fase de análisis en la que se investigó como funcionaban las herramientas desde dentro, que se podía aprovechar, que se necesitaba construir y como serían los nexos entre ello.
- La fase de implementación y mejora donde se realizaron todas las funciones y scripts de la herramienta.
- La fase de obtención y evaluación de resultados, ligada a la fase de implementación.

6.2. Análisis de objetivos

Se ha finalizado el proyecto con la construcción de un prototipo de una herramienta de conversión de voz tal y como se propuso. Se ha comprobado

con ello que el sistema que se quería construir basado en el enfoque teórico de los Modelos Ocultos de Markov para conversión de voz es implementable.

El prototipo funciona correctamente y se han obtenido resultados satisfactorios tal y como se ha expuesto en el capítulo anterior. Esto no quita que a su vez estos resultados sean mejorables debido a que se puede percibir cierto grado de robotización que puede dejar entrever que no se trata de una voz humana original, si no que se ha generado digitalmente a través de un ordenador.

Este sistema abre una nueva vía de mejoras en la conversión de voz muy prometedora, la cual no necesita del audio del locutor objetivo tras el entrenamiento, ni de una gran capacidad de almacenamiento ya que no es necesario almacenar ninguna base de datos.

Este nuevo aporte deja a un lado el tratamiento de la señal basado en una función de transformación, como si de un filtrado de señal se tratara. En vez de eso, partiendo del análisis de un archivo audio y gracias a las tecnologías de conversión texto-audio y la potencia y flexibilidad que aportan los Modelos Ocultos de Markov, se puede generar una nueva locución sin la necesidad de que esta sea la modificación de otra señal anterior.

6.3. Líneas futuras

Partiendo de este prototipo, o el nuevo concepto que el aporta, se puede realizar varias mejoras. La más importante sería mejorar la calidad de la sintesis para intentar eliminar ese grado perceptible de robotización del que hemos hablado.

En cuanto a la interpolación del pitch se podría dedicar un solo proyecto como este a lograr una herramienta que sea capaz de ajustar automáticamente la mejor curva de pitch teniendo en cuenta los factores de preservación de la prosodia del locutor fuente y la identidad del locutor objetivo.

En el reconocimiento y alineamiento, tal y como se ha comentado en la sección 4.9.1, se podría lograr una mejor coordinación entre los fonemas proporcionados por el reconocedor del audio y el los fonemas generados del texto, lo cual haría al sistema más robusto.

Enfocando la parte de síntesis hacía otra herramienta como “hts-engine”, la cual también aparece en la demo de HTS, se podría evaluar una nueva forma de generar los resultados. Para ello habría que reestructurar el sistema según el formato de archivos utilizados por esta herramienta.

Se podría ampliar el prototipo con una función que sólo necesitara de una base de datos reducida en la fase de entrenamiento. Esto se podría conseguir basándose en el concepto de adaptar la voz de los locutores con los que se va a trabajar a la de un locutor medio ya entrenado en el sistema, reduciendo así el tamaño de la base de datos necesaria para un nivel mínimo de calidad [2].

En conexión con este concepto, también se podría realizar un estudio sobre que método es mejor para construir una base de datos, que tamaño debería tener, que vocabulario y que características debería contener para lograr un determinado nivel de calidad.

Este prototipo trabaja por comandos a través de consola y tomando como entradas y salidas archivos que se deben colocar en unas determinadas carpetas que han sido estructuradas con una determinada jerarquía. Una mejora podría ser una interfaz gráfica que le de al usuario una mejor usabilidad de la herramienta, facilitando el uso de menús y opciones de una manera más simplificada y accesible a todo el mundo.

Por último, otro gran aporte sería el de realizar la conversión de voz en tiempo real. Para ello, se debería estudiar los tiempos de los microprocesadores necesarios para trabajar dentro una latencia mínima.

Apéndice A

Modelos Ocultos de Markov

Para entender mejor los Modelos Ocultos de Markov se va a proceder a desarrollar un ejemplo extraido de [17]:

Sean un conjunto de estados $S = \{s_1, s_2, ..., s_N\}$ con ciertas probabilidades de transición $A = \{a_{ij}\}$ entre ellos, los cuales producen unos determinados resultados observables $X = \{x_1, ..., x_N\}$, y una secuencia de variables aleatorias u observaciones $O = \{o_1, ..., o_T\}$ que pueden tomar alguno de los valores en X. La secuencia forma una cadena de Markov de orden 1 si cumple la propiedad de Markov, esto es, si dado el estado actual, los estados pasados y los futuros son independientes.

$$P(S_{t+1} = s' | S_1 = s_1, ..., S_t = s) = P(S_{t+1} = s' | S_t = s) \tag{A.1}$$

Una cadena de Markov de orden m sería aquella en la que la probabilidad

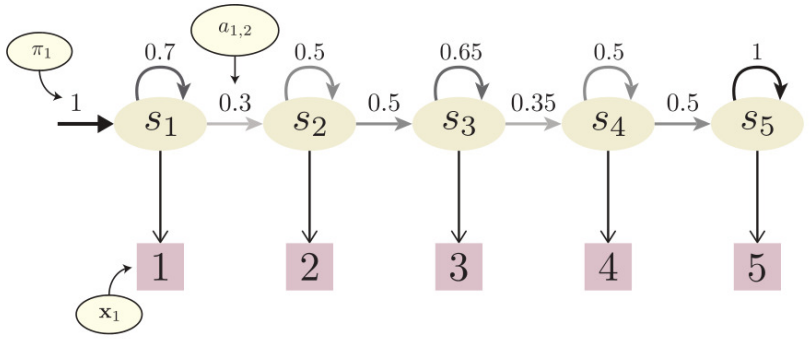

Figura A.1: Ejemplo de cadena de Markov de 5 estados

de ocurrencia de un estado depende de los m estados pasados, pero a partir de ahora se tomarán en consideración únicamente modelos de orden 1. La figura A.1 representa una cadena de Markov con N = 5, probabilidad 1 de comenzar en el estado s 1 y distintas probabilidades de transición.

Para tal cadena de Markov, la probabilidad de observar la secuencia $O = \{1; 1; 2; 3; 3\}$ sería de:

$$P(O) = \pi_1 a_{1,1} a_{1,2} a_{2,3} a_{3,3} = 6{,}825 \times 10^{-2} \tag{A.2}$$

A diferencia de las cadenas de Markov, donde observando los estados puede determinarse la verosimilitud de una secuencia, los estados de un HMM no son directamente observables sino que producen unos resultados observables u otros con una cierta probabilidad. De esta forma, la secuencia observada no se corresponde directamente con una secuencia de estados, sino que lo hace con una cierta probabilidad.

Un HMM se define como $\lambda = (\underline{\underline{A}}, \underline{\underline{B}}, \pi)$, donde $\underline{\underline{A}}$ es la matriz que contiene las probabilidades de transición entre estados, $\underline{\underline{B}}$ la matriz con los estadísticos de cada observable y π el conjunto de las probabilidades de estar inicialmente en cada estado del modelo.

La verosimilitud de un vector de características o_t para un estado s_i es:

$$b_i(o_t) = \mathcal{N}(o_t; \mu_i, \Sigma_i) \tag{A.3}$$

Donde μ_i y Σ_i hacen referencia a su media y covarianza respectivamente.

El ejemplo de la figura A.1 se convierte en un HMM si cada estado s_n lleva asociada una distribución Gaussiana de media n y varianza 0.5 (ver A.2).

En este caso la probabilidad de que una secuencia de entrada O haya sido producida por la secuencia de estados S se calcula de la siguiente manera:

$$P(O|S)P(S) = \pi_1 \; b_1(o_1) \prod_{i=2}^{5} a_{i-1,i} \; b_i(o_i) \tag{A.4}$$

Si por ejemplo se busca hacer reconocimiento, se buscará la secuencia que maximiza esta probabilidad. En el modelo usado para el lenguaje, cada estado de una cadena de Markov es una palabra, definiendo así las relaciones entre estas y la probabilidad de las posibles secuencias. El modelo acústico es un HMM donde cada estado representa una unidad de sonido, en este caso el

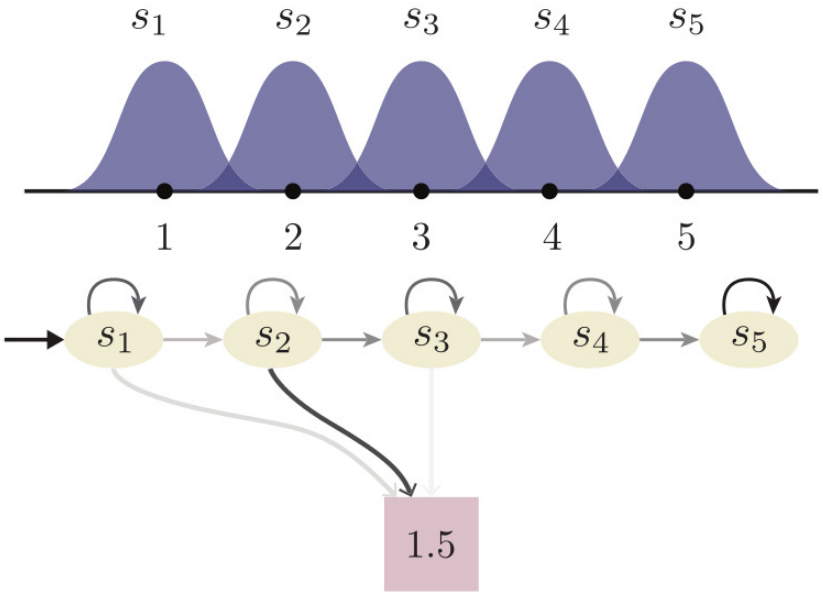

Figura A.2: Probabilidades de distribución (arriba) y de observación (abajo) de los distintos estados

fonema con contexto. Cada fonema se modela como tres estados, incluyendo así información sobre el fonema (o silencio) que lo precede y el que lo sigue dentro de una palabra.

Los parámetros estadísticos de las redes utilizadas se calculan generalmente mediante una estimación de máxima verosimilitud usando el algoritmo iterativo EM (Expectation Maximization) [18] a partir de una base de datos con ejemplos de cada tipo de sonido.

La figura A.3 muestra un ejemplo de una red de palabras y otro de una de fonemas. La red final puede verse como una red por capas, producto de la composición de las redes de sendos modelos. De esta forma, dos estados de la capa inferior pueden compartir el mismo modelo estadístico, pero ser aún así distintos por pertenecer a palabras diferentes en la red superior.

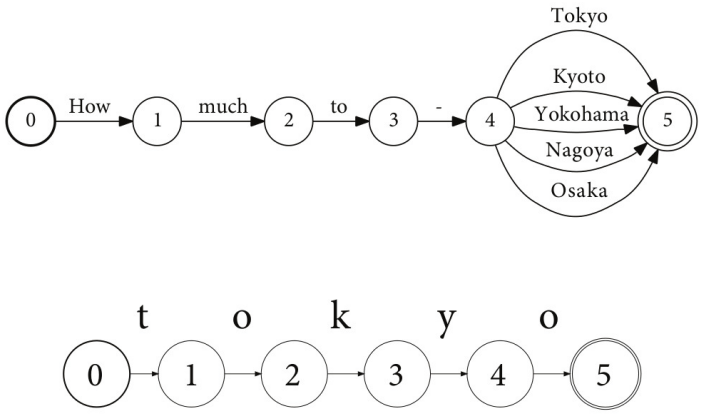

Figura A.3: Ejemplo de una red de palabras y fonemas [19]

Apéndice B

Funciones/Scripts

En este anexo se adjuntan las funciones más relevantes creadas en este proyecto.

B.1. Morphing

Este es el script general escrito en BASH, que se usa para la conversión de voz y genera un archivo wav como resultado. Su uso se compone de los siguientes comandos:

Morphing [Archivo audio locutor fuente] [locutor objetivo] [cepstrum: conEnergia/sinEnergia] [pitch: mediaPeriodo/mediaFrecuencia/mediaProporcional/rango/rangoPorcent

Donde se selecciona el archivo de audio del locutor fuente, el locutor objetivo deseado, si se quiere tener en cuenta la energía del locutor fuente o no en la conversión y que tipo de interpolación para el pitch se quiere usar de los que se han explicado en 4.4.

PSEUDO-CÓDIGO:

```
# Extraer f0 del audio
GetF0 ${NAME}

# Extraer cepstrums del audio
if [ "$CEPSTRUM" = "conEnergia" ]; then
GetMgc ${NAME}
fi

# Extraer duraciones .(rec) del audio
gen_rec_file -s ${NAME}.ses


# :::::::::: TXT --> UTT :::::::::: (TARGET)

# Pasar el .txt a script para festival

echo "${TEXTO}" >> script_festival.scm

# Generación de .utt con festival

festival --batch script_festival.scm
rm script_festival.scm


# :::::::::: UTT --> LAB-FULL :::::::::: (TARGET)

utt2lab ${NAME}

#listar el la etiqueta .lab en la cola de generacion
echo "ruta/${NAME}.lab" > ruta/colaGeneracion.scp

# actualizar contextos en mmf
cp ruta/labels/${NAME}.lab ruta/HTS-Albayzin-mono_${TARGET}/...
...data/labels/gen/${NAME}.lab

mmfUpdate ${TARGET}


# ::: (MORPHING) Mezclar duraciones del fuente con objetivo. :::
```

```
MixDur raw/${NAME}.rec labels/${NAME}.lab ...
...labels/${NAME}.fullstate.lab

mv labels/${NAME}.lab labels/${NAME}.old.lab
mv labels/${NAME}.fullstate.lab labels/${NAME}.lab


# :::::::::: LAB --> .LF0 (.PIT) + .MGC  ::::::::::


HMGenS -A -B -C syn.cnf -D -T 1 -s -S colaGeneracion.scp ...
...-t 1500 100 5000 -c 1 -H re_clustered_all.mmf.1mix -M tiedlist


# INTERPOLAR PITCH

cat ${NAME}.objetivo.lf0 | x2x +fa |
awk '{printf "%s\n",$1<0.0?0.0:16000/exp($1)}' |
x2x +af > ${NAME}_${SPKR}.objetivo.pit


cat ${NAME}.fuente.lf0 | x2x +fa |
awk '{printf "%s\n",$1<0.0?0.0:16000/exp($1)}' |
x2x +af > ${NAME}_${SPKR}.fuente.pit


if [ "${PITCH}" = "media" ]; then
InterpolarPitchMedia ${NAME}_${SPKR}.fuente.pit ...
...${NAME}_${SPKR}.objetivo.pit ${NAME}_${SPKR}.pit
elif [ "${PITCH}" = "mediaP" ]; then
InterpolarPitchMediaP ${NAME}_${SPKR}.fuente.pit ...
...${NAME}_${SPKR}.objetivo.pit ${NAME}_${SPKR}.pit
elif [ "${PITCH}" = "rangoP20" ]; then
InterpolarPitchP20 ${NAME}_${SPKR}.fuente.pit ...
...${NAME}_${SPKR}.objetivo.pit ${NAME}_${SPKR}.pit
else
InterpolarPitch ${NAME}_${SPKR}.fuente.pit ...
...${NAME}_${SPKR}.objetivo.pit ${NAME}_${SPKR}.pit
fi
```

```
# MEZCLAR ENERGIA

cd /home/edu/Escritorio/proyecto/GEN/generados

if [ "$CEPSTRUM" = "conEnergia" ]; then
MixEnergia ${NAME}_${SPKR}.fuente.mgc ...
...${NAME}_${SPKR}.objetivo.mgc ${NAME}_${SPKR}.mgc
else
mv ${NAME}.mgc ${NAME}_${SPKR}.mgc
fi

#::::: SINTESIS::::

SintetizarAlbayzin ${NAME}_${SPKR}.pit
```

B.2. MixDur

Como se ha explicado en la sección 4.2, este código toma las duraciones extraidas del reconocimiento del audio del locutor fuente gracias al script dado por el ViVoLAB de la EINA y se las impone a la etiqueta (.lab) generada con secuencia de fonemas y sus contextos.

Esta escrita en C y toma como entradas el archivo (.rec) del reconocimiento, el archivo (.lab) con los contextos y devuelve un archivo (.lab) con la imposición de las duraciones.

PSEUDO-CÓDIGO:

```
/* MixDur in.rec in.lab mix.lab */

/* LEER ARCHIVO */

while ( (read = getline(&linea_rec, &longitud, in_rec)) != -1 )
{
    inicio_leido = strtok(linea_rec," ");
    fin_leido = strtok(NULL," ");
    estado = strtok(NULL," ");

    if ( strncmp(estado, "s2", 2) == 0 )
    {
        getline(&linea_label, &longitud , in_lab);
        label = strtok(linea_label," ");
        label = strtok(NULL, " ");
        label = strtok(NULL, " ");
        /* ESCRIBIR LINEA */
        fprintf(out_mix, "%d %d [2] %s", inicio, fin, label);
    }
    if ( strncmp(estado, "s3", 2) == 0 )
        fprintf(out_mix, "%d %d [3] %s", inicio, fin, label);

    if (  strncmp(estado, "s4", 2) == 0 )alineamiento*/
        fprintf(out_mix, "%d %d [4] %s", inicio, fin, label);

       /* seguir con el resto de estados... */
}
```

B.3. InterpolarPitch

Como se ha visto en la seccion 4.4 tenemos varias formas de interpolar el pitch. Estos códigos están escritos en C y toman como entrada los archivos (.pit) que contienen el periodo del pitch de ambos locutores y devuelve un archivo (.pit) con los resultados interpolados. Como ejemplo se adjuntan los pseudo-códigos para hacerlo por media o por rango.

PSEUDO-CÓDIGO por RANGO:

```
/* InterpolarPitch (rango) entradaFuente.pit ...
...entradaObj.pit salidaInterp.pit */

/* LEER ARCHIVO pit */

leidosFuente = fread (pitchFuente_in, sizeof(float),...
...sizeFuente, fFuente_in );

leidosObj = fread(pitchObj_in,sizeof(float),sizeObj,fObj_in);

/* busqueda de maximos y minimos */

x=0;
while(pitchObj_in[x] == 0)
{
    x++;
}
/* inicializamos con los primeros valores distintos de 0*/
maxObj = pitchObj_in[x];
minObj = pitchObj_in[x];

for (x=0;x<sizeObj;x++)
{
   if( pitchObj_in[x] != 0 )
   {
        if (pitchObj_in[x]>maxObj)
           maxObj = pitchObj_in[x];
        if (pitchObj_in[x]<minObj)
           minObj = pitchObj_in[x];
   }
}
```

```
x=0;
while(pitchFuente_in[x] == 0)
{
    x++;
}
/* inicializamos con los primeros valores distintos de 0*/
maxFuente = pitchFuente_in[x];
minFuente = pitchFuente_in[x];

for (x=0;x<sizeFuente;x++)
{
    if( pitchFuente_in[x] != 0 )
    {
       if (pitchFuente_in[x]>maxFuente)
           maxFuente = pitchFuente_in[x];
       if (pitchFuente_in[x]<minFuente)
           minFuente = pitchFuente_in[x];
    }
}

/* INTERPOLACION */

rangoObj = maxObj - minObj;
rangoFuente = maxFuente - minFuente;
proporcion = rangoObj/rangoFuente;


for (i=0;i<sizeOut;i++)
{
    if (pitchFuente_in[i] == 0)
    {
        pitch_out[i] = pitchFuente_in[i];
    }
    else
    {
       aux = (pitchFuente_in[i] - minFuente);
       aux = aux*proporcion;
       pitch_out[i] =  minObj + aux ;
    }
}
```

```
/* ESCRIBIR ARCHIVO pit */

fwrite (pitch_out, sizeof(float), sizeOut, fInterp_out);
}
```

PSEUDO-CODIGO por MEDIA:

```
/* InterpolarPitch entradaFuente.pit...
... entradaObj.pit salidaInterp.pit */

/* LEER ARCHIVO pit */

leidosFuente = fread (pitchFuente_in, sizeof(float),sizeFuente, fFuente_in );
leidosObj = fread (pitchObj_in, sizeof(float), sizeObj , fObj_in );

/* MEDIA */

suma = 0;
contador = 0;
for (i=0;i<sizeFuente;i++)
{
    if( pitchFuente_in[i] != 0 )
    {
        suma = suma + pitchFuente_in[i];
        contador++;
    }
}
mediaFuente = suma/contador;

/* ...lo mismo para la media del objetivo */

traslacion = mediaObj - mediaFuente;

/*TRASLACION*/

for (i=0;i<sizeOut;i++)
{
   if (pitchFuente_in[i] == 0)
   {
      pitch_out[i] = 0;
```

```
   }
   else
   {
        pitch_out[i] = pitchFuente_in[i] + traslacion ;
   }
}

/* ESCRIBIR ARCHIVO pit */

fwrite (pitch_out, sizeof(float), sizeOut, fInterp_out);
```

B.4. MixEnergia

Como se ha visto en la seccion 4.5.1 se requiere de tomar el primer coeficiente de cada cepstrum del locutor fuente para incorporarlo en los cepstrum del locutor objetivo con la finalidad de preservar la curva de energía trama a trama en la conversión final. Este código está escrito en C y toman como entrada los archivos (.mgc) que contienen el los coeficientes mel-cepstrum de los locutores y devuelve un archivo (.mgc) con los resultados de la mezcla.

PSEUDO-CÓDIGO:

```
/* MixEnergia fuente.mgc objetivo.mgc out.mgc */

/* LEER ARCHIVO */

leidosFuente = fread (mgcFuente_in,...
...sizeof(float),sizeFuente, fFuente_in );
leidosObj = fread (mgcObj_in, sizeof(float),...
...sizeObj , fObj_in );

/* MIX energia*/

for (j=0;j<sizeOut;j++)
{
   if( (j%25) == 0 )
   {
       mgc_out[j] = mgcFuente_in[j];
   }
   else
   {
      mgc_out[j] = mgcObj_in[j];
   }
}

/* ESCRIBIR ARCHIVO pit */

fwrite (mgc_out, sizeof(float), sizeOut, fMix_out);
```

Apéndice C

Calendario de la elaboración del proyecto

<table>
<tr><th></th><th colspan="5">TAREAS</th></tr>
<tr><th>MES</th><th>DOCUMENTACIÓN</th><th>PREPARATIVOS</th><th>ANÁLISIS</th><th>IMPLEMENTACIÓN / SCRIPS</th><th>RESULTADOS</th></tr>
<tr><td rowspan="2">1</td><td rowspan="3">Documentación y estudio del estado del arte</td><td>Familiarización con LINUX, BASH, PYTHON, PERL...</td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td rowspan="3">Instalación y correcto funcionamiento de los toolkits (HTK, HTS, Festival, SPTK)
+
Entrenamiento de la DEMO HTS
(en inglés)</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Análisis del funcionamiento de la DEMO HTS
(Training.pl)</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td></td></tr>
<tr><td rowspan="2">3</td><td rowspan="2"></td><td rowspan="2"></td><td>Manejo de los formatos de los toolkits
(.utt, .lab, .wav, .lfo, .pit, .ses, .raw...)</td><td>HMGenS</td><td>Generación de parámetros desde una etiqueta de la base de datos</td></tr>
<tr><td>Base de datos Albayzin
(en español)</td><td rowspan="2"></td><td>HMGenS
+
imposición de duraciones</td><td>Generación de parámetros imponiendo las duraciones</td></tr>
<tr><td rowspan="4">4</td><td rowspan="5">Memoria</td><td rowspan="2"></td><td rowspan="4"></td><td>Generador/ Síntesis
+
mmfUpdate</td><td>Generación de audio (partiendo de texto) que no pertenezca al Corpus de entrenamiento</td></tr>
<tr><td>Reconocedor
(ViVoLAB)</td><td>Obtención de la secuencia de estados y duraciones del locutor fuente</td></tr>
<tr><td rowspan="2">Base de datos SEV-Joaquín
(en español,
con emociones)</td><td>MixDur</td><td>Conversión de voz en duraciones</td></tr>
<tr><td>GetF0 /
InterpoladorPitch</td><td>Conversión de voz en pitch</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2"></td><td rowspan="2"></td><td>GetCepstum /
mixEnergía</td><td>Conversión de voz en energía</td></tr>
<tr><td>Morphing</td><td>Primeros audios de conversión de voz completos</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Pruebas con diferentes interpolaciones de pitch y energías.
Mejoras en los scripts y en su usabilidad</td><td>Búsqueda de posibles mejoras / perfeccionamiento de los resultados</td></tr>
<tr><td></td><td></td><td></td><td></td><td>SCRIPTS DEFINITIVOS</td><td>RESULTADOS DEFINITIVOS</td></tr>
</table>

Bibliografía

[1] A. V. Oppenheim and R. W. Schafer, "A history of the cepstrum," *IEEE*, vol. 106, 2004.

[2] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, pp. 1234–1252, May 2013.

[3] T. Masuko, "Hmm-based speech synthesis and its applications," *Technical report*, November 2002.

[4] P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department, University of Hawaii at Manoa, Honolulu, USA*, 2008.

[5] S. Young and H. YE, "High quality voice morphing," *Int Conference Acoustics Speech and Signal Processing*, 2004.

[6] Y. Nambu, M. Mikawa, and K. Tanaka, "Voice morphing based on interpolation of vocal tract area functions using ar-hmm analysis of speech.," *Interspeech*, 2009.

[7] Y. Nambu, M. Mikawa, and K. Tanaka, "Flexible voice morphing based on linear combination of multi-speakers' vocal tract area functions," *18th European Signal Processing Conference (EUSIPCO-2010)*, August 2010.

[8] S. Young and H. YE, "Perceptually weighted linear transformations for voice conversion," *Eurospeech 2003*, 2003.

[9] H. Ye and Steve, "Quality-enhanced voice morphing using maximum likelihood transformations," *YEYO06*, 2006.

[10] J. Yamagishi, "An introduction to hmm-based speech synthesis," *Technical report*, October 2006.

[11] S. Young, J. Odell, D. Ollason, V. altcho Valtchev, and P. Woodland, "Htkbook," *Technical report of Cambridge University Engineering Department*, 2005.

[12] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameters generation algorithms for hmm-based speech synthesis," *ICASSP2000*, 2000.

[13] Y. Wang and J. Tao, "Implementation of parameter generation in hmgens," *Technical report*, January 2015.

[14] D. Braga, P. Silva, J. Freitas, D. Monterde, and M. S. Dias, "Building an hmm-based spanish tts system for albayzin 2010 challenge," *FALA 2010*, 2010.

[15] R. Barra-Chicote, J. M. Montero, J. Macias-Guarasa, S. Lufti, J. M. Lucas, F. Fernandez, L. F. D'haro, R. San-Segundo, J. Ferreiros, R. Cordoba, and J. M. Pardo, "Spanish expressive voices: Corpus for emotion research in spanish," *6th international conference on Language Resources and Evaluation. Marrakech (Morocco)*, 2008.

[16] S. Young and H. YE, "Cued research project: Voice morphing. university of cambridge. http://svr-www.eng.cam.ac.uk/ hy216/voicemorphingprj," *University of Cambridge's website*, 2004.

[17] J. Vallés, "Paralelización del algoritmo de búsqueda de un reconocedor automático de voz," *PFC Escuela de Ingeniera y Arquitectura de Zaragoza*, 2014.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, 39(1):1–21*, 1977.

[19] P. Dixon and S. Furui, "Introduction to the use of wfsts in speech and language processing," *APSIPA Conference*, 2009.