Universitat Ramon Llull

TESI DOCTORAL

Títol HMM-based speech synthesis applied to Spanish and English, its applications and a hybrid approach

Realitzada per Xavier Gonzalvo Fructuoso

en el Centre Escola Tècnica Superior d'Enginyeria Electrònica i Informàtica La Salle

i en el Departament Comunicacions i Teoria del Senyal

Dirigida per Dr. Joan Claudi Socoró Carrié
Dr. Ignasi Iriondo Sanz

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es
www.url.es

Abstract

Nowadays, Human Computer Interface (HCI) is one of the most studied disciplines in order to improve real human interactions with machines on the present time and for the incoming future. More and more electronic devices of the daily life are used by more people. This electronic incursion is mainly due to two reasons. On the one hand, the undoubted increasing of the economical accessibility to this technology but on the other hand, the more friendly interfaces that allow an easier and more intuitive use. As a matter of fact, nowadays it is only necessary to observe the personal computer interfaces, pocket size computers and even mobile telephones. All these new interfaces let little experienced users make use of cutting edge technologies. Moreover, the inclusion of speech technologies in these systems is becoming more usual since speech recognition and synthesis systems have improved their performance and reliability.

The purpose of speech technology is to provide systems with a natural human interface so the use can be extended to daily life. Text-To-Speech (TTS) systems are one of the main modules under intense research activity in order to improve their naturalness and expressiveness. The use of synthesizers has been extended during the last times due to the high-quality reached in real limited domain applications and the good performance in generic purposes applications. However, there is still a long way to go with respect to quality and open domain systems.

This work will present a TTS system based on a statistical framework using Hidden Markov Models (HMMs) that will deal with the main topics under study in recent years such as voice style adaptation, trainable TTS systems and low print databases. Moreover, a cutting edge hybrid approach combining concatenative and statistical synthesis will also be presented. Ideas and results in this work show a step forward in the HMM-based TTS system field.

Resumen

Hoy en día, la Interacción Hombre-Máquina (IHM) es una de las disciplinas más estudiadas con el objetivo de mejorar las interacciones humanas con sistemas reales para el presente y para el futuro venidero. Más y más dispositivos electrónicos son usados por más gente en la vida diaria. Esta incursión electrónica se debe principalmente a dos razones. Por un lado, el indudable aumento en la accesibilidad económica a esta tecnología pero por otra parte, unos interfaces más amigables que permiten un uso más fácil e intuitivo. Simplemente hace falta observar hoy en día los ordenadores personales, las computadoras de bolsillo e incluso los teléfonos móviles. Todos estos nuevos dispositivos admiten que usuarios poco experimentados puedan hacer uso de las tecnologías más punteras. Por otra parte, la inclusión de las tecnologías del habla está llegando a ser más común gracias a que los sistemas de reconocimiento y de síntesis de voz han estado mejorando su funcionamiento y fiabilidad.

El objetivo final de las tecnologías del habla es crear sistemas tan naturales como los seres humanos para que su uso se pueda extender a cualquier rincón de la vida diaria. Los conversores de Texto-a-Voz (o sintetizadores) son de los módulos que más esfuerzo investigador han recibido con el objetivo de mejorar su naturalidad y la expresividad. El uso de los sintetizadores se ha ampliado durante los últimos tiempos debido a la alta calidad alcanzada en usos de dominio restringido y el buen comportamiento en aplicaciones de propósito general. De todas formas, todavía queda un largo camino por recorrer por lo que respecta a la calidad en aplicaciones de dominio abierto. Además, algunas de las tendencias de los sistemas sintetizadores conllevan reducir el tamaño de las bases de datos, sistemas flexibles para adaptar locutores y estilos de locución y sistemas entrenables.

Esta tesis doctoral presentará un sintetizador de voz basado en el entorno probabilístico de los Modelos Ocultos de Markov (MOM) que lidiará con los principales temas estudiados en la actualidad tales como adaptación del estilo de locutor, sistema conversores de voz entrenables y bases de datos de tamaño reducido. Se describirá el funcionamiento convencional de los algoritmos y se propondrán mejoras en varios ámbitos tales como la expresividad. A la vez se presenta un sistema híbrido puntero que combina modelos estadísticos y de concatenación de voz. Los resultados obtenidos muestran como las propuestas de este trabajo dan un paso adelante en el ámbito de la creación de voz sintética usando modelos estadísticos.

Acknowledgements

I would like to warmly thank my supervisors Dr. Joan Claudi Socoró and Dr. Ignasi Iriondo for sharing their knowledge during the development of this work, for providing countless suggestions, fruitful discussions, important feedbacks and constant help and support while working abroad. To Dr. Jose Antonio Morán Moreno for his guidance into the research field. Moreover, I would like to thank Carlos Monzo for his friendly cooperation during a long time in many projects. I also want to thank the rest of the people directly or indirectly involved in the section of Theory and Signal processing of La Salle University starting with the head of Multimodal Research Group (GPMM) Elisa Martínez and other colleagues (Santi, Lluís, Xuti).

I would also like to thank Phonetic Arts Ltd. and its chief executive Paul Taylor for the support and opportunity to develop a lot of the exciting work presented in this thesis. I appreciate the kind support of Alex and Peter, Monika, Stefan and the rest of the great team: Ant, Ian, Matt, Yannis, Orla and many others.

First stage of this thesis. was funded by IntegraTV4all (FIT-350301-2004-2) and SALERO (EU-FP6-IST-507142). Final stage has been develop thanks to the support of Phonetic Arts Ltd.. I am thankful for this support, which is a crucial economic basis for any research.

I am heartily thankful to my family who all have given me a loving environment where to develop. I would like to express how grateful I am with my mother for her unconditional support who always encouraged me to do my best in all matters of life. My loving thanks are due to Elena for her infinite kind understanding and her support during my work abroad. Without their encouragement and understanding it would have been impossible for me to finish this work.

I would like to extend my gratitude to the nice city of Cambridge for being a lovely and inspiring place where I could develop most of the final efforts of this thesis.

Agradecimientos

En primer lugar quisiera agradecer a mis supervisores Dr. Joan Claudi Socoró y Dr. Ignasi Iriondo por compartir su conocimiento durante todo el desarrollo de este trabajo, por proporcionarme incontables sugerencias, fructíferas discusiones, importantes comentarios y por la ayuda y el soporte constante trabajando en el extrangero. Al Dr. José Antonio Morán Moreno por introducirme en el campo de la investigación. Por otra parte, me gustaría dar la gracias a Carlos Monzo por su amistosa cooperación durante ya muchos proyectos. También deseo agradecer al resto de la gente implicada directa e indirectamente en el Departamento de Comunicaciones y Teoría de la Señal de La Salle, empezando por la directora del Grup de Recerca Multimodal (GPMM) Elisa Martínez y sus otros colegas (Santi, Lluís, Xuti).

Me gustaría también dar las gracias a Phonetic Arts Ltd. y a su presidente Dr. Paul Taylor por el soporte y la oportunidad de poder desarrollar gran parte del excitante trabajo presentado en esta tesis. Aprecio el amable apoyo de Alex y Peter, Monika, Stefan y el resto del gran equipo: Ant, Ian, Matt, Yannis, Orla y muchos más.

La primera parte de esta tesis fue financiada por IntegraTV4all (FIT-350301-2004-2) y SALERO (EU-FP6-IST-507142). El periodo final ha sido desarrollado gracias al apoyo de Phonetic Arts Ltd. Estoy agradecido por esta ayuda, que es una base económica crucial para mi investigación.

Estoy sinceramente agradecido a mi familia por darme un ambiente agradable para poder desarrollar este trabajo. Deseo expresar cómo estoy de agradecido a mi madre por su apoyo incondicional, quién siempre me ha animado a dar lo mejor de mí en esta vida. Un tierno agradecimiento es para Elena por su infinita comprensión y su cercanía durante largas etapas alejados. Este trabajo es para ellos, ya que sin su ánimo y comprensión hubiera sido imposible para mí acabar este trabajo.

Me gustaría extender mi gratitud a la ciudad de Cambridge por ser un encantador e inspirador lugar donde poder elaborar los esfuerzos finales de esta tesis.

Contents

List of Tables

List of Figures

Acronym glossary

AG Accentual Group

APME Aperiodicity-based Mixed Excitation

AR Auto Regressive

ASR Automatic Speech Recognition

CBR Case Based Reasoning

CGV Contextual Global Variance

CMLLR Constrained Maximum Likelihood Linear Regression

CPSP Cross Correlation Phase Product

DOA Direction Of Arrival

DTW Dynamic Time Warping

EM Estimation Maximization

ESPRIT Estimation of Signal Parameters via Rotational Invariance Techniques

G2P Grapheme to phoneme

GMM Gaussian Mixture Models

GPMM *Grup de Processament Multimodal*

GV Global Variance

HCI Human Computer Interface

HMM Hidden Markov Model

HNM Harmonic plus Noise Model

HSMM Hidden semi-Markov Model

HTK Hidden Markov Model Toolkit

IG Intonational Group

LMGE Local Minimum Generation Error

LPC Linear Predictive Coding

LSP Linear Spectral Pair

MAP Maximum a Posteriori

MDL Minimum Description Length

MELP Mixed Excitation Linear Prediction

MFCC Mel Frequency Cepstrum Coefficients

ML Maximum Likelihood

MLSA Mel Logarithmic Spectrum Approximation

MLLR Maximum Likelihood Linear Regression

MGE Minimum Generation Error

MSD MultiSpace Distribution

NLU Natural Language Understanding

NLP Natural Language Processor

POS Part-Of-Speech

PSOLA Pitch-Synchronous Overlap and Add

SALERO Semantic AudiovisuaL Entertainment Reusable Objects

TTS Text-To-Speech

TC Text Classification

Co-TTS Concatenative Unit Selection-based Text-To-Speech synthesis

VFS Vector Field Smoothing

VoQ Voice Quality

VS Virtual Speaker

VSME Voicing strengths-based Mixed Excitation

VT Voice Transformation

Chapter 1

Introduction

This thesis concerns speech synthesis using Hidden Markov Model (HMM), defined as an HMM-based TTS system hereon. In particular, the thesis focuses on HMM-based TTS systems for Spanish and English and proposes different ways of improving naturalness and expressiveness of the synthetic speech. Furthermore, we show how different applications for video games were developed using this type of synthesis system. Finally, we propose a novel hybrid synthesis based on HMM.

1.1 General framework

TTS synthesis is one of the key technologies in speech processing. It is a technique for creating speech from given texts in order to communicate machines and people as part of a Human Computer Interface (HCI). Figure 1.1 depicts an example of video game application. This interface is a multimodal dialogue system framework where multiple input sources can be used simultaneously. In particular, an Automatic Speech Recognition (ASR) module converts speech into text and the Natural Language Understanding (NLU) block processes an input text to extract users' intentions (e.g., ask for a cinema ticket). The dialogue manager controls the action to be taken (e.g., answer a question) at each step of the conversation and the Natural Language Processor (NLP) module is the opposite to the NLU block converting intentions into natural text. Finally, speech reaches the final user through a TTS system. Firstly, this system produces neutral speech and then, an optional voice transformation technique can also be used in order to introduce extra effects (e.g., emotions).

Ideally, these additional input modes in the framework shown in Figure 1.1 can also be handwriting (Vielhauer and Scheidat, 2005), events from hardware interfaces (Young, 2007) (e.g., mouse movements) among others (e.g., motion capture). Today this is a reality in games such as EyePet (PlayStation3 ®) and on certain gaming devices (e.g., Nintendo®Wii).

Nowadays, these HCI have evolved so as to include media technologies to create virtual col-

laborative frameworks. In the past, some of these systems employed virtual actors (e.g., Virtual Speaker (VS) (Melenchón et al., 2003)) whereas these frameworks now include multiple speech technologies, 3D audio, image processing and virtual reality.

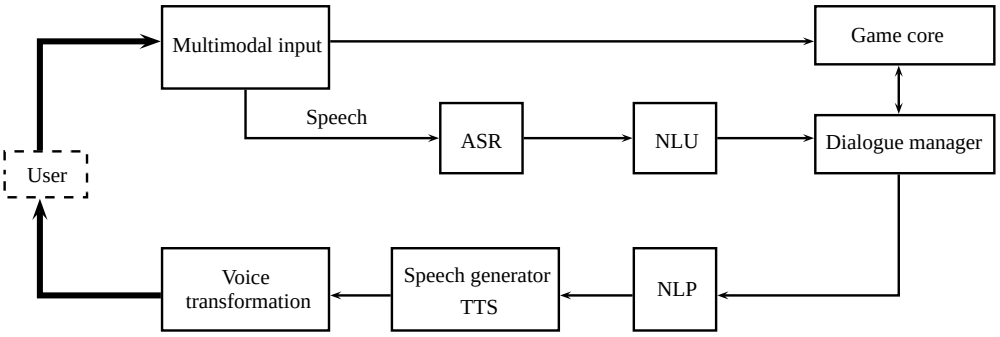


Figure 1.1: An example game engine application described as a dialogue interaction with multiple input sources, a dialogue manager and speech generated by a TTS system. Speech is converted to text by the Automatic Speech Recognition (ASR) module whereas Natural Language Understanding (NLU) and Natural Language Processor (NLP) blocks are used to convert text into the semantic level and viceversa, respectively.

TTS systems have been conventionally applied to applications where the purpose is focused on producing stable and intelligible speech (e.g., telephony), whereas naturalness has been often relegated to a second position. By contrast, TTS system applied to games differs on some aspects to that applied to conventional tasks (Rozak, 2007). Most computer games use recorded speech which produces perfect and natural speech. The production efforts are a very time consuming process which TTS systems can optimize. Nevertheless, the following constraints must be taken into account when TTS systems are applied to video games:

- Speech can be produced online or offline. On the one hand, Figure 1.1 shows an online speech production since the TTS system is integrated within the game engine so that the game can produce live speech. On the other hand, offline speech would be statically introduced into the game. Obviously, the possibility to create infinite content within the game and let the user create their own content makes the online option more attractive.
- Players demand the best visual and audio effects quality.
- Speech in games is very expressive and might contains multiple styles and emotions.
- Usually, several voices are required for each of the different characters of the game.

As we can see, video games are a particular application which requires TTS systems to generate natural speech with arbitrary speaker's voice characteristics and speaking styles. There exist different approaches to face those requirements and to produce stable and expressive speech. Currently there are basically three main trends: Concatenative Unit Selection-based Text-To-Speech synthesis

(Co-TTS), HMM-based TTS and hybrid systems. Each of these approaches have advantages and disadvantages as well as different purposes. A brief discussion and overview is described in the following sections.

1.2 Speech synthesis: an overview

As it has been described, the TTS system is responsible to generate synthetic speech from an input text. Usually the input of a conventional speech synthesis system is plain text, though a richer synthesis query might be also possible using a standard structured language (e.g., Speech Synthesis Markup Language (SSML) (W3C, 2004)). The idea is to control the synthesis system to produce not only a desired text but also to incorporate additional information (e.g., specific word intonation).

TTS systems have improved over the last years though it is still not possible to design the perfect synthesis for any application and all domains (Black, 2002). It is a matter of fact that the quality depends on the type of application, the amount of data and the affordable complexity of the resulting system. A compromise is needed between the target quality and the domain of the applications being designed. In speech synthesis, a domain refers to the semantic topic of the application (e.g., sports, politics or action dialogues in films). The wider the range of domains a TTS can handle maintaining the quality of the synthetic speech, the more generic the system is considered to be. Hence, a generic system can synthesize any domain with the same quality and naturalness. Generally, a generic synthesizer requires a very large corpus.

Usually, regardless of the type of synthesis system, two different types of applications can be designed. On the one hand, a limited domain system (Alías et al., 2005) which can produce a high-quality synthetic speech while it has a drastically decrease of performance when text is out of domain or when attempting to expand the domain itself. On the other hand, a generic TTS system requires a very large corpus and computationally expensive unit selection algorithms (Black, 2002). In addition, a so-called Multi-domain TTS was presented by (Alías et al., 2006a, 2008) to overcome the problems of a generic TTS when applied to a limited number of domains.

In the following sections, different speech synthesis approaches will be presented. In addition, their performance will be discussed along with the type of application they are intended to be used in.

1.2.1 Synthesis approaches

Speech production techniques can be classified into the following types (Huang and Hon, 2001; Taylor, 2009):

- **Formant-based techniques**. They make use of the acoustic-tube model in such a way that the control elements of the tube are easily related to acoustic-phonetic properties that can

easily be observed. The formant synthesizer is not an accurate model of the vocal tract, so the general assessment is that the quality is intelligible but far from natural. One of the most sophisticated system at this time was the Klatt synthesizer (Klatt, 1980).

- **Articulatory-based techniques**. They use a physical model of speech production that includes all the articulators. Some of the systems are based on fluids dynamic principles (Sinder, 1999) and some of the newest systems are based on the use of 3D models originally proposed by (Badin et al., 1998) and recently presented with new techniques by (Engwall et al., 2006) and also designed with HMMs (Zhang and Renals, 2008).

- **Sinusoidal model of speech**. This analysis and synthesis technique is based on a set of sine-wave components derived using the pitch frequency and voicing decisions. Synthetic phases are assigned to each respective sine wave. Usually, sine-wave amplitudes and phases are estimated by sampling a linear combination of frequency domain basis functions. The work presented by (Chazan et al., 2000) uses the frequency domain algorithm for the reconstruction of speech from the Mel Frequency Cepstrum Coefficients (MFCC). The basis function gains are determined such that the mel-frequency binned spectrum of the reconstructed speech is similar to the mel-frequency binned spectrum, obtained from the original MFCC vector by IDCT and antilog operations. Natural and intelligible sounding quality speech is obtained by this procedure. Furthermore, last advances by (Chazan et al., 2006) show an efficient sinusoidal modelling framework for high quality wide band speech synthesis and modification. This technique may serve as a basis for speech corpus compression in the context of small footprint concatenative TTS systems and it becomes simpler and considerably more efficient than STRAIGHT (see Section 3.3.5) since it outperforms it in speech quality for both speech reconstruction and transformation.

- **Source filter model**. It models speech as a sound source which is then modified by a vocal tract filter (see a detailed description in Section 1.2.4.2).

Depending on the process of the synthesis units, three main trends are being under development and research (Narayanan and Alwan, 2005; Taylor, 2009):

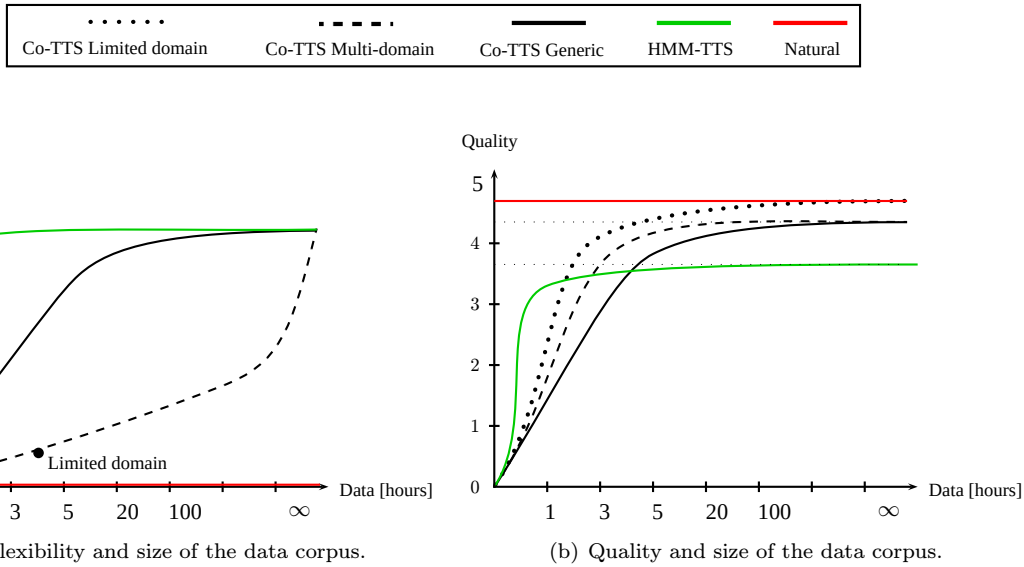
- **Concatenative Unit Selection-based Text-To-Speech synthesis (Co-TTS)**. It generates speech by concatenating unit segments. The main module is a unit selection system, whose basic premise is that one can synthesize new naturally sounding utterances by selecting appropriate sub-word units from a database (see more details in Section 1.2.2). It is possible to concatenate not only natural units but also parameterized units. It usually requires a signal processing module such as the Time-Domain Pitch Synchronous Overlap Add Method (TD-PSOLA) (Moulines and Verhelst, 1995; Moulines and Charpentier, 1990).

- **Statistical speech synthesis based on HMMs (HMM-based TTS system)** (Tokuda et al., 1994a, 2000, 2002b; Black et al., 2007). This type of speech synthesis system selects or

generates the parameters from a probabilistic model, usually an HMM. Although HMMs were originally designed for speech recognition, modelled units are also a good representation for synthesis purposes. Statistical speech synthesis based on HMM is described in Chapter 2 and is briefly introduced in Section 1.2.4.

- **Hybrids**. In this work, this term includes any system that uses both statistical and concatenative approaches in a single framework. The aim of those systems is to overcome the problems of any isolated approach while emphasizing their advantages (a complete description is presented in Chapter 4).

As we will see, a trade-off between quality, data and flexibility is the essence of TTS system design. Nowadays, Co-TTS and statistical synthesis systems based on HMM using a source-filter model are the two main basic trends for speech synthesis. Each of them has some advantages and disadvantages related with these three issues that are discussed through Figure 1.2.



(a) Flexibility and size of the data corpus.

(b) Quality and size of the data corpus.

Figure 1.2: Speech synthesis development trade-off schematics.

In these Figures, *Flexibility* refers to the possibility of using vocabulary and sentences not employed for training during synthesis time. That is, if the synthesizer can produce any synthesis for any input text without degrading the output quality, the system is considered to have a high flexibility. On the contrary, natural speech is considered to have no flexibility since each sentence is recorded. *Quality* of a TTS is usually measured by means of subjective experiments such as Comparision Mean Opinion Scores (CMOS) (Taylor, 2009). Regardless of the technique, the quality refers to how natural the synthetic speech is with respect to a reference baseline. This reference can be natural speech or another TTS system.

Figure 1.2(a) is an extension of the original figure presented for natural units of variable lengths (Yi and Glass, 1998), then used for concept-to-speech synthesis (Taylor, 2000) and finally extended to include Multi-domain synthesis (Alías et al., 2008). In the Figure, Co-TTS (in black) and HMM-based TTS (in green) systems are compared along with natural speech (in red).

The conventional approach to a Generic Co-TTS tends to strive the flexibility at the expense of quality. Systems focused on specific domains are constrained to improve firstly the quality and then the flexibility. An HMM-based TTS system is considered to fit in the former approach since the system produces a very flexible voice even with less data than a Co-TTS.

Using the same comparison philosophy, Figure 1.2(b) depicts the relation between quality and data size. Note that in the extreme case of having a large amount of data (and assuming that it would be possible to manage), current HMM-based TTS and Co-TTS systems would differ in the final quality. On the one hand, it is a matter of empirical fact that Co-TTS systems reach the highest quality when they use natural units. Nevertheless, any system can guarantee the perfect concatenation for any application anytime. In fact, this problem becomes worse as the number of domains increase and it becomes a problem for generic TTS systems specially because the probability of having errors in the corpus is more likely. Some of the common errors which degrade the quality of the concatenation are due to incorrect phoneme segmentation, sparsity data problems or errors in the unit selection algorithm. On the other hand, statistical systems produce an stable quality even with less data than the Co-TTS system. Unlike the increase of the data has a positive effect for Co-TTS systems, HMM-based TTS systems are not affected in the same manner because the lack of naturalness is not produced by insufficient data but because of statistical processing and data modelling (see Section 2.6 for a detailed description about the over-smoothing effect).

In the following sections, Co-TTS and source-filter model HMM-based TTS systems are briefly described.

1.2.2 Concatenative Text-To-Speech synthesis (Co-TTS)

Concatenative systems using unit selection are the most used systems since they can produce a very natural synthetic voice with a low computational cost. Unit selection is a natural extension to solve the problem of managing a large number of units for concatenative systems. A unit is defined by means of a specification, which contains information about linguistic features and prosodic parameters (Taylor, 2009).

The standard to unit selection was established with the formalism presented by (Hunt and Black, 1996). A cost function is defined in order to select the best units by minimizing a total cost. The unit selection technique must satisfy a target cost (i.e., how close a database unit is to a desired unit) and a join cost (i.e., how well two adjacent selected units join together). The unit selection process is designed to optimally minimise both target and join costs.

The design of a Co-TTS requires to address the following aspects:

- **Speech segments to use**. The smaller the units, the easier it might be to have coverage over the whole acoustic phonetic space as each unit may provide better sharing of contexts. Common units are diphones used as half-phones (Iriondo et al., 2003) in order to concatenate by the stable part of the signal. Smaller units have been used, for example 5 ms segments using HMM states (Hirai et al., 2007).

- **Design of the corpus**. It is a matter of fact that more data yields to a better synthesis since more units can be selected so a better joint is more likely. However, computational cost for selecting the best units increases as well. The synthesis of a system strongly reflects the style and coverage of the recorded databases. Domain specific databases can be built for specific applications (e.g., weather forecast (Alías et al., 2006a)) while open domain applications require a well balanced corpus to tackle a wide range of possible inputs. Some open systems such as BOSS (Breuer and Hess, 2010) or Festival (Black et al., 1999) are focused on multi-functionality and multi-linguality.

- **Unit selection algorithm**. From a large set of possible units to concatenate, the best ones need to be chosen from the recorded database. Two cost functions are defined: the target cost $C_t(\mathbf{f}_n, u_n)$ is used to estimate the mismatch between the target specification vector $\mathbf{f}_n$ (which is usually linguistic, prosody and spectrum contexts) and the candidate unit u_n; the concatenation cost $C_c(u_i, u_{i+1})$ is used to estimate the smoothness of the acoustic signal when concatenating units u_i and u_{i+1}.

  Given a sequence $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$ of specification vectors, the cost for one possible sequence of units $\mathbf{u} = \{u_1, u_2, \ldots, u_N\}$ is:

$$C(\mathbf{u}, \mathbf{F}) = \sum_{n=1}^{N} C_t(\mathbf{f}_n, u_n) + \sum_{n=2}^{N} C_c(u_{n-1}, u_n)$$

  The unit selection can then be formulated as the problem of finding the optimal sequence of units $\mathbf{u}^*$ from multiple candidate units $\mathbf{u}$ that minimizes the total cost,

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} C\left(\mathbf{u}, \mathbf{F}\right) \tag{1.1}$$

Unfortunately, although the advantages of this approach, it also has some drawbacks. Generally, this kind of system suffers of quality variability when applied to a non-limited domain application. In order to control the possible problematic concatenation artifacts, corpus size is likely to increase (Black, 2002) or domain can be limited or classified such in the next section.

1.2.3 Multi-domain synthesis

Depending on the domain, three types of Co-TTS systems can be defined: limited domain, multi-domain and generic purposes systems.

According to the discussion of Figure 1.2, the final goal of a TTS system is to produce natural speech for any input text. Two different approaches are commonly used (use Figure 1.2(a) as a reference). On the one hand, an approach is to design a generic TTS to be used in any application. On the other hand, a constrained design is focused on the domain where the TTS system works in order to produce the best quality for a specific domain. A multi-domain system belongs to the second category where N domains are used as a pre-cluster of the units. In this case, the unit selection algorithm reduces its search space. It is shown by (Alías et al., 2008) that a Co-TTS system presents a better performance (i.e., quality) within its own domain. In addition, by pre-selecting the range of units, the system also guarantees the most appropriate prosody for the input text.

In order to automatically categorize the input text in one of the N domains, a text classification (TC) technique is used. Conceptually, automatic TC is a discipline that arises from the intersection of information retrieval (representation of the used data) and artificial learning (techniques to model the information). The aim of the TC is to cluster text into a specific domain. The classification assumes the natural word sequentiality in the text. An Associative Relational Net (ARN) (Rennison, 1994) can be used to represent all the words of the modelled text whose connections are defined as the number of times that these matched words appear within the text. Note that in the context of a TTS system, TC has an extra difficulty due to the reduced lengths of the texts (e.g., a 5 words sentence).

1.2.4 HMM-based TTS system

In the following section, HMM-based TTS systems are introduced and the source-filter model is also described as the speech production approach for this type of synthesis system.

1.2.4.1 Brief introduction

HMM-based TTS system is a technique for generating speech from trained statistical models where information of basic speech units are modelled in a single framework (e.g., vocal tract, pitch and durations) (Yoshimura et al., 1999). This type of synthesis approach appeared in the 90s in order to overcome most of the limitations of the Co-TTS systems. Its main difference is the use of a statistical model as part of the speech production system.

One of the main interests of TTS systems is to achieve the naturalness of the real speech. As we have seen in Section 1.2.2, Co-TTS systems can reach a very good quality for limited domain applications whereas it presents a set of disadvantages when applied to a different domain. Briefly, most problems are related to corpus errors (e.g., linguistic labelling or unit segmentation) and in the nature of the concatenative algorithms (e.g., missing linguistic features or concatenation smoothing). Moreover, some other problems arise when attempting to expand a specific corpus (e.g, different recording sessions might have different audio levels). Also, a Co-TTS system uses natural units which provides a very high quality although, typically, it also implies a very large corpus.

In contrast, unlike the concatenative approach, HMM-based TTS systems do not use natural units during synthesis time but a model learned during a training stage. This capability makes these systems more suitable to synthesize different speaker features, styles and emotions. Synthesizing different speaking styles through concatenative speech synthesis still requires large databases in contrast to HMM which can obtain better results with smaller databases (Yamagishi et al., 2005). However, voice transformation techniques (Erro, 2008) offer a solution in order to adapt Co-TTS systems without using a corpus-based approach although HMM-based adaptation has also been shown to be very efficient transforming speech models (Yamagishi et al., 2009).

Some interesting voice transformation approaches using HMM were presented using speaker adaptation (Tamura et al., 1998), an eigen-voice technique (Shichiri et al., 2002) or interpolation models (Yoshimura et al., 2000).

The main problem of HMM systems is the over-smoothing effect due to the statistical processing of the training data. This produces muffled speech and flat intonation. Many techniques have been proposed in order to alleviate this problem. The idea is to enhance the HMMs in order to introduce part of the missing variability lost during the training stage. As described further in Section 2.6, a common approach is to use a global variance model during synthesis, the so-called Global Variance (GV) described in Section 2.6.1.

Language is a key topic in the design of a TTS system. HMM-based TTS system uses a decision tree-based context clustering which works as a unit selection system since it can characterize synthesis units using a set of linguistic features. In other words, the scheme of the HMM-based TTS system is based on contextual factors for clustering and can be adapted to any language (see Section 2.8 for a description of the unit selection approaches used in HMM synthesis). By the time of this dissertation, HMM-based TTS systems were applied to around 15 languages (e.g., Catalan (Bonafonte et al., 2008), English (Tokuda et al., 2002b) or Portuguese (Maia et al., 2003)). As a result of the present work, a Castilian Spanish HMM-based TTS system was firstly presented by (Gonzalvo et al., 2007b) and is described in Section 2.8.3.

The conventional unit used for an HMM-based TTS system is the phoneme. The use of the minimal unit in this kind of system is not as critical as for a concatenative approach because speech is synthesized using parameters generated from HMMs and in consequence, it does not suffer from segmentation and concatenation problems. However, some hybrids systems that combine Co-TTS and HMM-based TTS systems might use diphones as the basic unit [1].

The HMM-based TTS system presented in this work is based on a source-filter model (Section 1.2.4.2) approach to generate speech directly from the HMM itself using a Maximum Likelihood (ML) criterion (Section 2.5.1 describes the parameter generation algorithm in detail).

Current challenge for HMM synthesis is to increase naturalness and expressiveness. Unlike other synthesis approaches, quality is still an issue for this type of system due to the vocoder. As it will

[1] Usually Co-TTS systems use diphones as the base unit due to their advantages during concatenation (Lambert and Breen, 2004). As described in Chapter 4, concatenation-driven hybrid systems might force the HMM system to use diphones in order to build a consistent design.

be described, new modelling techniques with more complex mixed excitations or sinusoidal models are being investigated. Nevertheless, HMM synthesis systems are a very promising approach due to its stability and performance. Hybrid systems based on HMMs are also a very attractive solution in order to take advantage of the HMM properties (e.g., spectral transition smoothness).

1.2.4.2 Source filter model

The source filter model (see Figure 1.3) is a type of speech production technique where speech is comprised of a source component (or excitation signal) modified by the vocal-tract. In particular, this excitation signal ($e[n]$) contains voiced and unvoiced parts which are shaped by the vocal-tract filter coefficients ($h[n]$). Excitation signal is a representation of the residual signal ($r[n]$). This signal is the prediction error after codifying input signal $s[n]$ with the filter coefficients. Theoretically, when the residual signal is filtered with these filter coefficients, the same input signal is obtained $s[n]$. Otherwise, an estimation of the input signal is produced $\hat{s}[n]$.

The reconstruction of speech is performed in a different way depending if it is voiced or unvoiced. The source $e[n]$ in a voiced segment of speech tries to reproduce the vibration of the vocal folds (i.e., a periodic signal). In unvoiced speech frames, the source is not a regular vibration but rather caused by turbulent airflow due to a constriction in the vocal tract (i.e., noise). Depending on the complexity of this excitation signal, different qualities can be achieved (see Section 3.5). Basically, an excitation can be: a simple pulse and noise, a multiband mixed excitation or other approaches. All these approaches are discussed in the aforementioned section.

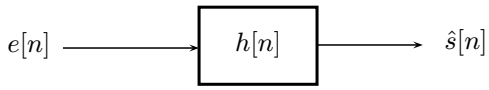


Figure 1.3: Source filter model scheme. Filter coefficients $h[n]$ are set to encode an input signal $s[n]$. When the excitation $e[n]$ is filtered with this filter, an estimated $\hat{s}[n]$ is obtained.

The basic problem of a Co-TTS system is the discontinuities in joint points. The source filter model is able to reduce this effect by smoothing the spectral discontinuities, though they cannot be completely eliminated. The use of source filter models undertake the processing of the source and the filter separately offering more control over the resulting synthesized signal.

One important advantage of using this representation is that several voice-quality effects can be achieved relatively easily (e.g. whisper). In addition, as we will see in Chapter 2, the source filter approach is convenient for voice conversion techniques (e.g. emotion adaptation) since it separates speech in source and vocal tract.

Synthesis systems based on this model are not able to use the true excitation signal. In contrast, they usually use a simplification, some form of periodic pulses and noise signal. The HMM-based TTS system presented in this work uses this approach and statistically models the spectrum information and the fundamental frequency contour to build an excitation signal.

1.3 Motivation

At this point, HMM-based TTS systems have been introduced and briefly discussed along with other synthesis approaches. So, why developing a synthesis system using HMM? Let's analyze the advantages and disadvantages of the HMM technique when applied to the framework described in Section 1.1. On the one hand,

- Unlike other synthesis systems which require enormous amounts of storage, HMM voices have a low footprint.
- During synthesis time, system performance is very high (i.e., it is not computational expensive, thus the system can be very fast in terms of time real time),
- Unlike Co-TTS systems, parameters are generated by an algorithm using a smoothing constraint, so there are no concatenation errors. This is a significant advantage as the HMM-based TTS system can steadily produce stable speech (e.g., content created by users).
- HMM can be modified by statistical techniques. Adaptation techniques can be applied to produce different speaking styles or emotions using only a few amount of target data.
- The system can be automatically trained.

On the other hand, regarding the main disadvantages:

- Synthetic speech sounds buzzy, muffled and unnatural when a conventional source filter model is used (described in Section 1.2.4.2). Therefore, one of the objectives would be to propose an improved excitation in order to overcome the problem in the source-filter approach and improve the naturalness.
- Synthetic speech tends to be mostly flat with respect to expressiveness (i.e. part of the prosody of the natural speech is not reproduced in the synthetic speech). Since the goal of the statistical process is to generalize the data in the HMM, part of the characteristics of the original speaker are lost. This would require a different perspective in order to take advantage of the stability of the HMM prosody while increasing the expressiveness.
- Apart from the previous problems, it is important to note that synthetic speech generated from HMMs significantly differs from the natural speech reference. This is the price of low footprint voices and smoothness. Although over-smoothing compensation techniques introduced in Section 1.2.4.1 (e.g., GV) can alleviate this problem, we believe that a step forward is needed and a novel synthesis system could be proposed using advantages from Co-TTS and HMM systems. Such a system could produce a better quality, incorporate HMM main characteristics (e.g., adaptation) and maintain a relatively small memory consumption. In other words, a hybrid system primarily guided by an HMM background.

One can see that HMM-based TTS systems are a promising approach to produce speech synthesis. In principle, it is limited to applications where stability rather than naturalness is the main issue. However, if we could overcome these three main disadvantages, HMM-based TTS system could become an even more interesting technique. Next section defines the objectives of this thesis in order to face these challenges.

1.4 Objectives

As it has been described, HMM synthesis was originally proposed in order to overcome most of the problems of the conventional speech synthesis systems. Nevertheless, HMM-based TTS system faces itself a set of limitations which affect the number of application it can be applied to.

In order to give a solution to some of these problems, this thesis. describes a set of advances and applications for the HMM-based TTS system. The main objective here is to **enhance the quality of the state-of-the-art of HMM-based TTS systems**. The purpose of this thesis is broken down as follows:

1. **Adapt the HMM-based TTS system to the Spanish Castilian language**. This basically involves proposing linguistic settings for Spanish. In addition, study the baseline of the conventional HMM-based system performance for that language.

2. **Improve naturalness**. Lack of naturalness in the HMM-based TTS systems is due to the use of a simple excitation in the source-filter model (introduced in Section 1.2.4.2). In order to improve the quality of the system, we propose to use a mixed excitation based on a multiband approach.

3. **Improve expressiveness**. The idea is to use a new technique in order to make the system more expressive by proposing an alternative prosody enhancing technique. In particular, the goal is to improve the F0 contour generated by the HMM introducing natural prosody estimated from an external system.

4. **Apply adaptation techniques to real applications**. The purpose is to use adaptation for speaking styles (i.e. emotions) and speaker identity.

5. **Investigate the future of HMM systems and propose a hybrid approach**. Propose a hybrid system as a composite TTS system in which HMM and concatenative techniques are used simultaneously in order to take advantage of their properties. The objective is to design a novel hybrid system based on an HMM-driven approach

1.5 Contents of this Ph.D.

Since this thesis is about speech synthesis and more specifically about HMM-based TTS systems, introduction in this Chapter 1 presented the general framework, an overview of the state-of-the-art of TTS systems (Section 1.2) and a brief discussion comparing synthesis approaches in order to give general idea of the advantages and drawbacks of each technique.

The HMM-based TTS system is described in **Chapter 2**. Firstly, it briefly introduces other uses of HMMs in speech synthesis. It then difines HMM in order to give the necessary background for a better comprehension of each of the parts of an HMM-based TTS system. HMM training for synthesis is presented in Section 2.3. In this section, training is broken down in a list of steps. The synthesis stage is described in detail in Section 2.5 with special emphasis on parameters generation (see Section 2.5.1), over-smoothing problem and some solutions (see Section 2.6). The proposed F0 enhancing technique introduced in the objectives is defined in Section 2.6.3. This chapter also describes HMM adaptation (see Section 2.7) which will be used for emotion and speaker adaptation applications further in the experiments of Sections 5.4.1 and 5.4.2. Unit selection using decision-tree-based clustering is detailed in Section 2.8 where details of the Castilian Spanish peculiarities are also described in Section 2.8.3.

Data parameterization is described in **Chapter 3**. Basic vocal tract coders are introduced and compared by their performance for HMM synthesis purposes (see Section 3.3). Moreover, parameterization is also presented for fundamental frequency (see Section 3.4) and different types of excitations (see Section 3.5). Proposed excitation is presented in Section 3.5.2.

Chapter 4 is entirely dedicated to hybrid systems. Firstly, an introduction of the state-of-art describes all existing techniques. There are basically two types of hybrids, concatenation-driven and HMM-driven. A novel hybrid approach based on an HMM-driven technique in presented in Section 4.3 using a simplified version of one of the concatenation-driven approaches described in Section 4.2.

Finally, **Chapter 5** presents objective and subjective experiments to show improvements to the HMM-based TTS system baseline. It firstly gives details of corpora used (Section 5.1). Experiments have been classified according to its purpose. Summarizing, there are four types of experiments: experimental tests (see experiments in Section 5.2), proposed work and baseline improvements (experiments in Section 5.3), applications (Section 5.4) and overall TTS performance (see Section 5.5). Among other experiment, this chapter details Spanish and English HMM-based TTS system performance, expressiveness and naturalness improvements, results for the proposed hybrid system and two applications: an emotional HMM-based TTS systems and a speaker identity adaptation.

To sum up, **Chapter 6** concludes with the work done as well as some remarks for the future work.

There are also five additional annexes. Annex A introduces the basics of prosody prediction estimation and describes a F0 estimator used in the proposed F0 enhancing technique. In Annex B,

the Minimum Generation Error (MGE) algorithm is reviewed in detail. Also, Annex C.1 refers to the distance between two HMMs. A preliminary formulation for contextual GV is given in Annex D. In addition, Annex E presents contributions to projects and the use of tools involved during the development of this work and finally, Annex F lists all conference contributions.

Chapter 2

Hidden Markov Model-based speech synthesis

The general approach to produce synthetic speech is to use a concatenative system in the so-called data-driven approach. In this chapter, an alternative to this approach is presented. HMM are described as a machine-learning technique to infer the parameters of a statistical model from the training data. Unlike the concatenative approach, statistical systems are not memorising the data but performing a learning process. Main advantages arising from that definition regard to low footprint voices, the possibility to adapt the models using linear regression techniques and a more stable system immune to concatenation errors. Firstly, a brief introduction describes conventional use of HMMs for speech synthesis. In the next sections, HMMs are described in detail including the novel proposed F0 enhancing technique and the Spanish HMM system.

2.1 The use of Hidden Markov Models for speech synthesis

Hidden Markov Models are a very useful statistical technique used in a wide range of applications. Originally, HMM became very popular to model speech sequences in order to recognize continuous speech. As it will be described in this chapter, HMM has been successfully applied to produce sequences of parameters to generate synthetic speech. However, the use of HMM for speech synthesis is not only focused to this application but to many others. In fact, the use of HMM for speech synthesis systems can be categorized as follows (Zen et al., 2004):

- **Speech corpus preparation**. Speech databases need to label each unit (e.g., phonemes) in order to be able to use them appropriately. HMMs are used to automatically perform the segmentation of the transcription of the recorded utterances. This involves:

  - Speech segmentation. The purpose is to set the time boundaries of each unit in the audio data (see Section 2.1.1).
  - Grapheme to phoneme (G2P) conversion. One of the possible approaches to tackle G2P is to use discrete HMMs to model sequences of phonemes and graphemes (see Section 2.1.2).

- **Synthesis**.
  - Parameter generation. Synthetic speech is created by generating parameters from an HMM. This type of synthesis system can use almost any of the speech production models presented in Section 1.2.1 such as a source filter model (Tokuda et al., 2002b) or a sinusoidal model (Hemptinne, 2006). Statistical synthesis is the main topic described in this thesis and it is briefly introduced in Section 2.1.3 and further described from Section 2.2 onwards.
  - Hybrids. Another possibility is to be part of a composite system where HMM are intended to produce an inventory of units which can be used by a Co-TTS (see Chapter 4). These kind of system were originally defined as trainable TTS systems (Donovan and Eide, 1998).

2.1.1 Speech segmentation

Speech segmentation is used to set the times boundaries of all the units for concatenative inventory production. Usually, those databases are used by Co-TTS systems which need very accurate labels in order to appropriately concatenate units during the selection process.

There exist many techniques to address this problem (Adell and Bonafonte, 2006) although the most common one is HMM. As described by (Taylor and Isard, 1991), the process consists on performing a recognition task over the recorded voice. Considering the phone sequence is already known, only an HMM sequence is allowed and models' transitions give us the boundaries of the phones.

The basic procedure to compute a proper phone alignment is described as follows:

1. **Prepare data**. Create observations to be modelled by the HMMs from speech frames. This process is identical to the one used for speech synthesis and MFCC are commonly preferred (see Chapter 3).
2. **Prepare model**. Usually a non-ergodic left-to-right topology is used where each phoneme is modelled by an HMM (e.g., 5-states).
3. **Model initialization**. If no preprocessed data is available (i.e., no hand labelled boundaries) a flat start must be performed. This implies initializing HMMs using a global variance and running multiple instances of embedded re-estimation (using the Estimation-Maximization -EM- algorithm) (Young et al., 2006).

4. **Initialize silences**. Correct time boundaries for silences are crucial for the speech synthesis quality. A common solution is to firstly consider short pauses after each word and if their durations become longer than a threshold promote them to a proper silence in subsequent iterations. Because of the similarities of the silence and short pause models, these models share the central state parameters.

5. **Unit alignment**. This step is closely related to conventional speech recognition decoding. The main difference in this case is that the process forces the alignment to a set of known phonemes so the only unknown information are the time boundaries.

2.1.1.1 Error detection

The main problem of speech segmentation (and evidently of speech recognition too) is how to detect misrecognitions that produce incorrect phoneme boundaries. In order to do that, confidence scores output by the recognizer can be used. These scores are employed by the decoding algorithm but they do not represent any absolute measure of the match and are meaningful only in comparison to other hypotheses produced for the same utterance.

There exist different possibilities depending on the use of these scores. On the one hand, scores could be normalized in order to take into account overall recognition goodness. This solution is not trivial and the rate of false alarm can be very high (Young, 1994). On the other hand, an utterance verification algorithm offers a more interesting solution. Basically, each sentence is assigned a confidence score and those falling below a threshold can be discarded or manually inspected and corrected.

If we assume an acoustic observation O is produced by a sequence of words W, then the goal of the speech recognition system is to determine the most probable word sequence $\hat{W}$, given the observed acoustic signal O. This is represented by the Bayes' equation as,

$$\hat{W} = \arg\max_{W} \left(p(W|O)\right) = \arg\max_{W} \left(\frac{p(O|W)P(W)}{P(O)}\right) \tag{2.1}$$

In most recognisers the denominator (acoustic probability of the observation) is assumed equal for all observations and is not considered in calculations. As we have noted earlier in this section, this means that the likelihoods of the recogniser are not absolute measures for the probability of O but relative measures used to compare different utterance hypotheses.

If we intend to normalize these scores, it would be possible to use the denominator of the fundamental equation to help in computing the confidence information for an hypothesis since the ratio will be an absolute measure of the probability of the word sequence. $P(O)$ can be approximated by general purpose recognizers based on filler networks (Akyol and Erdogan, 2004).

2.1.1.2 Utterance verification

Utterance verification (UV) is a post-processing stage to examine the reliability of the hypothesized recognition result (used in ASR or corpus segmentation). Under the framework of UV, two complementary hypotheses are proposed, namely the null hypothesis H_0 and the alternative hypothesis H_1. Hypothesis are tested using the Likelihood Ratio Test (LRT):

$$LRT_i = \frac{p_i(O|H_o)}{p_i(O|H_1)} \underset{H_1}{\overset{H_o}{\gtrless}} \tau_i \tag{2.2}$$

The classification of the observation O as belonging to phoneme i is deemed to be correct (H_o) or incorrect (H_1) depending on the value of an estimated likelihood ratio relative to the threshold τ_i.

2.1.1.3 Alternative hypothesis

The major difficulty with LRT is how to model the alternative hypothesis which usually represents a very complex and composite event, where the true distribution of data is unknown. In practise the same HMM structure is adopted to model alternative hypothesis, which can be:

- A general background model.
- Hypothesis-specific anti-model.
- A set of competing models.
- A combination of all the above.

So-called anti-models are often used as models to test the incorrect classification. The anti-model may be trained specifically or they can use a combination of competitor scores. The latter offers a good performance though it requires calculation of all the models in the space for each hypothesis, thus it can be very computationally expensive. However, since the values of most competing models are mainly sources contributing to alternative hypothesis score, only a small number of competing models can be considered

2.1.2 Grapheme to phoneme conversion (G2P)

The G2P problem tries to determine the best sequence of phonemes for a word when only its graphemes are known. There are basically three main approaches to the G2P problem. On the one hand, hand-written rules is the conventional system which explicitly requires an expert developer to design the system response (a good example is SinLib described in Section E.3 which has been

used in this work to perform all the linguistic analysis for Spanish). On the other hand, a data-driven approach can use an algorithm to learn those rules from the data. Within this category, the statistical approach is one of the possible systems among others such as neural networks (Damper et al., 1999).

The use of HMM for grapheme to phoneme conversion was proposed by (Taylor, 2005). In this approach, phonemes are the hidden states, the transitions between the phonemes describe the probability that one phoneme will follow another and the graphemes are observations. Just like speech recognition, this system will find the most probable sequence of phonemes that could generate the input grapheme observations.

Each individual HMM represents one phoneme which can generate up to four graphemes. The internal topology of the HMM does not require looping states because what is being modelled does not have a time dependency. As you can see in Figure 2.1, this is a left-to-right model with 4 emitting states.

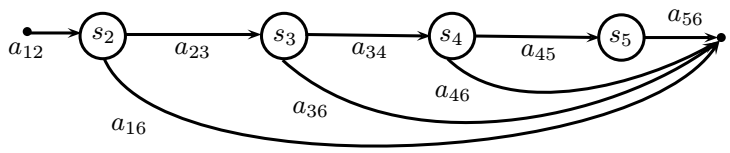


Figure 2.1: HMM topology for G2P.

By using EM training (Young et al., 2006), graphemes are aligned with the phonemes and transition and observation probabilities are estimated in the same space. The training process is summarized as follows:

- Generate data for grapheme observations. Since discrete model probabilities are used, each grapheme is mapped with a unique binary code.
- Produce a prototype definition according to Figure 2.1.
- Re-estimate models using observations.

As just explained, the basic system will attempt to perform the recognition of phonemes using only information of the current observation. It is obvious that prior information is very useful in this problem. For this reason, language models such as n-gram and context-sensitive modelling (i.e., triphones) can be applied to increase the accuracy of the grapheme to phoneme conversion.

2.1.3 HMM synthesizer

HMM-based TTS systems has been briefly introduced in Section 1.2.4. This system simultaneously models different speech parameters and is based on a source-filter model approach. Originally, some

of the first HMM synthesizers were reported in the late 80s and 90s (Ljolje and Fallside, 1986; Giustiniani and Pierucci, 1991; Fukada et al., 1994). These firsts attempts showed discrete HMMs mapping phonemes to the acoustic states and reconstruction of F0 contours.

Later on, a trainable IBM synthesizer system was proposed by (Donovan and Woodland, 1995; Donovan and Eide, 1998; Donovan and Woodland, 1999). It used the HMM state sized segments as its basic synthesis units and a dynamic programming search to optimize a perceptually motivated cost function. It is not until the work proposed by (Tokuda et al., 1995; Masuko et al., 1996a) when an algorithm for speech parameter generation from continuous mixture HMMs included delta and delta-delta parameters. Similar approaches had been applied for other similar systems such as the one proposed by Microsoft (Huang et al., 1997; Hon et al., 1998).

The conventional HMM-based TTS system takes the form of a complete synthesizer with the work proposed by (Yoshimura et al., 1999). In that system, spectrum, pitch and state duration are simultaneously modelled in a unified HMM framework. Later, different parameter generation algorithms were proposed by (Tokuda et al., 2000).

Subsequently, interest was focus on linguistic features firstly for English (Tokuda et al., 2002b) and later this served as the baseline for many other languages (e.g., Portuguese (Maia et al., 2003) and Spanish (Gonzalvo et al., 2007a)). In order to improve the naturalness, a multiband mixed excitation and a postfiltering technique were proposed by (Yoshimura et al., 2001). Basically, the purpose was to tackle the buziness. In addition, over-smoothing has been alleviated with approaches such as a GV model (Toda and Tokuda, 2005), a MGE technique applied during HMM training (Wu and Wang, 2006) and trajectory HMMs (Zen et al., 2004).

One of the first overviews was presented by (Black et al., 2007) where the first attempt to describe advantages and disadvantages of HMM systems were also described. In addition, hybrid systems using HMMs were also established as a new synthesis trend in this paper. Finally, and up-to-date overview can be found by (Zen et al., 2009).

2.2 Hidden Markov Models (HMMs)

The following section defines the HMM framework introducing training and synthesis procedures.

2.2.1 Definition of an HMM

An HMM is a widely used statistical models to treat sequential sources. It has successfully been applied to speech recognition systems since (Rabiner, 1990), with special corpus for ASR (Price et al., 1988) and last advances related to novel approaches to train continuous-density HMM (Jiang et al., 2006), audiovisual recognition (e.g., gestures (Nam and Wohn, 1996), office activities modelling (Oliver and Horvitz, 2005), behaviour modelling (Oliver and Pentland, 2000)), audiovisual synthesis (Tamura et al., 1999) and speech synthesis (described in the following sections of this

chapter).

In the following sections, a description of the basic theory of HMMs is presented. As we will see, the basic modelling is based on the likelihood of the trained model with respect to the data being modelled. Please note that all vectors are columns vector unless stated otherwise.

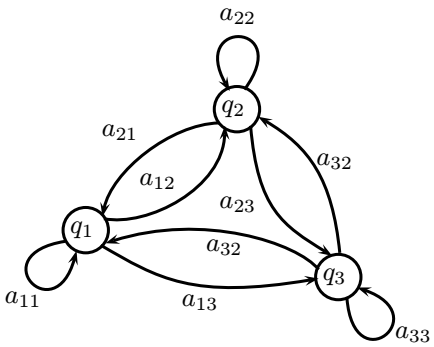


Figure 2.2: An ergodic HMM.

An HMM is a finite state machine (see Figure 2.2) which generates a sequence of discrete time observations $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$, where $\mathbf{o}_t$ is the column observation vector at time t. At each time unit (i.e., frame), the HMM changes its state according to a state transition probability distribution a_{ij} (the probability to go from current state i to next state j). It generates an observation $\mathbf{o}_t$ at time t according to the output probability distribution of the current state $b_i(\mathbf{o}_t)$.

An N-state HMM is defined in compact notation in equation 2.3, where $\mathbf{A} = \{a_{ij}\}_{i,j=1}^{N}$ is the state transition probability distribution, $\mathbf{B} = \{b_j\}_{j=1}^{N}$ is the output probability distribution and the initial state probability distribution is $\mathbf{\Pi} = \{\pi_i\}_{i=1}^{N}$.

$$\lambda = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi}) \tag{2.3}$$

The output probability distributions $b_j(\mathbf{o}_t)$ for state j can be discrete or continuous depending on the observations. Output distributions for a continuous density HMM are modelled by a mixture of multivariate Gaussian distributions,

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} \omega_{jm} \cdot \mathcal{N}\left(\mathbf{o}_t | \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm}\right) \tag{2.4}$$

where M is the number of mixture components, ω_{jm} is the weight of the m-th mixture, $\boldsymbol{\mu}_{jm}$ and $\mathbf{U}_{jm}$ are the column mean vector and a covariance matrix of mixture component m of state j, respectively.

A multivariate Gaussian distribution is defined by $\mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm})$ with mean vector $\boldsymbol{\mu}_{jm}$ and

covariance matrix $\mathbf{U}_{jm}$, that is,

$$\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jm}, \mathbf{U}_{jm}) = \frac{1}{\sqrt{(2\pi)^L|\mathbf{U}_{jm}|}} \exp\left(-\frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_{jm})^{\mathrm{T}}\mathbf{U}_{jm}^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_{jm})\right) \tag{2.5}$$

where L is the dimensionality of $\mathbf{o}_t$, $(\cdot)^{\mathrm{T}}$ stands for the transpose operator, $|\cdot|$ is the determinant and $(\cdot)^{-1}$ is the inverse matrix.

Multiple data streams [1] can be used to combine multiple data sources in a single HMM. This is used to distinguish different types of parameters during training (e.g., vocal tract and fundamental frequency). Equation 2.6 shows the output probability distribution of the observation at time t as the product of Gaussian mixture densities in order to combine S independent data streams,

$$b_j(\mathbf{o}_t) = \prod_{s=1}^{S}\left[\sum_{m=1}^{M_s}\omega_{jsm}\cdot\mathcal{N}(\mathbf{o}_{st}|\boldsymbol{\mu}_{jsm}, \mathbf{U}_{jsm})\right] \tag{2.6}$$

where $\mathbf{o}_{st}$ is the observation for stream s at time t, ω_{jsm} is the mixture weight and M_s is the number of mixtures in stream s. Note from this equation that streams are independent events unlike a mixture of Gaussians where the output distribution is as a weighted combination of elements (see Equation 2.4).

2.2.2 Basic Problems for HMMs

There are three basic problems that can be solved in order to use the HMMs (Rabiner, 1990). These problems are described in the following paragraph assuming a single Gaussian ($M = 1$). A mixture of Gaussians can be developed straightforwardly.

1. **Evaluation**. Given a model and a sequence of observations, the problem tries to compute $P(\mathbf{O}|\lambda)$, the probability of the observation sequence to be produced by the model, that is, scoring how well a given model matches a given observation sequence. Given a hidden state sequence $\mathbf{Q} = \{q_1, q_2, \ldots, q_T\}$ and an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$, the total likelihood of generating $\mathbf{O}$ from this HMM λ is calculated by summing $P(\mathbf{O}, \mathbf{Q}|\lambda)$ for all possible state sequence,

$$P(\mathbf{O}|\lambda) = \sum_{\forall\mathbf{Q}}\prod_{t=1}^{T} a_{q_t q_{t+1}} b_{qt}(\mathbf{o}_t) \tag{2.7}$$

   This likelihood is calculated using forward-backward procedures. The forward probability α takes into account the observation sequence until t and the backward probability β the observations from $t + 1$ until T. A further development can be found by (Rabiner, 1990).

[1] HTK library (Young et al., 2006) does this generalization in order to let observation vectors be a composite structure with different types of data (e.g., voval-tract and F0). This formulation is presented since most of the work in this thesis is based on this library.

Assuming that the first and last state of any HMM are non-emitting (Young et al., 2006),

$$\begin{aligned} \alpha_j(t) &= P\left(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, q_t = j|\lambda\right) \\ &= \left(\sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij}\right) b_j(\mathbf{o}_t) \end{aligned} \tag{2.8}$$

$$\begin{aligned} \beta_j(t) &= P\left(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \ldots, \mathbf{o}_T|q_t = j, \lambda\right) \\ &= \sum_{i=2}^{N-1} a_{ji} b_i(\mathbf{o}_{t+1})\beta_i(t+1) \end{aligned} \tag{2.9}$$

where N is the number of states. The total likelihood is given by

$$P(\mathbf{O}|\lambda) = \alpha_N(T) = \beta_1(1) \tag{2.10}$$

2. **Recognition**. Given a model λ and an observation sequence $\mathbf{O}$, this problem tries to find the optimal state sequence $\mathbf{Q}^*$. Typical use concerns to find the optimal state sequence associated with the given observation sequence (i.e., continuous speech recognition). The best state sequence can be obtained by a manner similar to the forward procedure, which is often referred to as the Viterbi algorithm (Forney, 1973).

3. **Training**. It adjusts the model parameters $\lambda = (A, B, \pi)$ to maximize $P(\mathbf{O}|\lambda)$, that is, observation sequences from a set of examples are used to train the HMM. Model parameters can be obtained using an iterative procedure such as the expectation-minimization (EM) which is often referred as the Baum-Welch algorithm.

   The essential problem is to estimate the means and covariances of an HMM. The state output distribution is considered to be a multivariate Gaussian distribution (Equation 2.4). Theoretically, if there were only one state and one mixture, the maximum likelihood estimates for $\boldsymbol{\mu}$ and $\mathbf{U}$ would be just the simple averages, that is

$$\hat{\boldsymbol{\mu}}_j = \frac{1}{T}\sum_{t=1}^{T} \mathbf{o}_t$$

   and

$$\hat{\mathbf{U}}_j = \frac{1}{T}\sum_{t=1}^{T} (\mathbf{o}_t - \boldsymbol{\mu}_j)(\mathbf{o}_t - \boldsymbol{\mu}_j)^{\mathrm{T}}$$

   However, in practice there are multiple states and mixtures so there is no direct assignment of observation vectors to individual states because the underlying state sequence is unknown. Thus, the training procedure is based on the probability of state occupation function $\gamma_j(t)$ which denotes the probability of being in state j at time t. Then, the maximum likelihood estimates of $\boldsymbol{\mu_j}$ and $\mathbf{U}_j$ are the weighted averages of Equations 2.11 and 2.12, which are the

Baum-Welch re-estimation formulae for the means and covariances of an HMM.

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{t=1}^{T} \gamma_j(t)\mathbf{o}_t}{\sum_{t=1}^{T} \gamma_j(t)} \tag{2.11}$$

and

$$\hat{\mathbf{U}}_j = \frac{\sum_{t=1}^{T} \gamma_j(t)(\mathbf{o}_t - \boldsymbol{\mu}_j)(\mathbf{o}_t - \boldsymbol{\mu}_j)^{\mathrm{T}}}{\sum_{t=1}^{T} \gamma_j(t)} \tag{2.12}$$

The probability of state occupation (being in state j at time t) is calculated using the Forward-Backward algorithm. By using Equations 2.8 and 2.9 as the forward and backward probabilities, respectively, and Equation 2.10 as the total likelihood based on these probabilities,

$$P(\mathbf{O}, q_t = j|\lambda) = \alpha_j(t)\beta_j(t)$$

$$\begin{aligned} \gamma_j(t) &= P(q_t = j|\mathbf{O}, \lambda) = \frac{P(\mathbf{O}, q_t = j|\lambda)}{P(\mathbf{O}|\lambda)} \\ &= \frac{1}{P(\mathbf{O}|\lambda)}\alpha_j(t)\beta_j(t) \end{aligned} \tag{2.13}$$

Equation 2.13 refers to the probability of state occupancy when a single Gaussian is used. In this case, the probability of mixture component occupancy would be equal to this probability. For an HMM with M_s mixture components in stream s, the probability of occupying the m-th mixture component in stream s and time t is straightforward. For $t > 1$,

$$\gamma_{jsm}(t) = \frac{\alpha_{jsm}(t)\beta_{jsm}(t)}{P(\mathbf{O}|\lambda)} b^*_{js}(\mathbf{o}_t) \tag{2.14}$$

where $\alpha_{jsm}(t)$ and $\beta_{jsm}(t)$ will take into account the multivariate Gaussian distribution formulae using the weight w_{jsm} and the output probability distribution $b_{jsm}(\mathbf{o}_{st})$. The second term of Equation 2.14 is an adjust for the rest of the streams (Young et al., 2006) defined as,

$$b^*_{js}(\mathbf{o}_t) = \prod_{k \neq s} b_{jk}(\mathbf{o}_{kt})$$

where each of the $b_{jk}(\mathbf{o}_{kt})$ are the total probability for all mixtures in state j and stream k.

4. **Synthesis**. Originally, HMMs were used for recognition purposes. However, HMMs are capable to model the speech in such a manner that the trained units are a good representation for synthesis as well. Hence, the use of the HMM in the field of speech synthesis takes to consider the generation of parameters from the HMM itself [2]. The aim of the synthesis problem is to obtain a speech parameter vector sequence $\hat{\mathbf{C}}$ which will be used to generate the speech for an

[2] A brief introduction is presented here while a further explanation of the generation of the parameters is in Section 2.5.1.

input text in the form,

$$\hat{\mathbf{C}} = \left[\hat{\mathbf{c}}_1^{\mathrm{T}}, \hat{\mathbf{c}}_2^{\mathrm{T}}, \ldots, \hat{\mathbf{c}}_T^{\mathrm{T}}\right]^{\mathrm{T}} \quad (2.15)$$

$$\hat{\mathbf{c}}_t = \left[\hat{c}(1), \hat{c}(2), \ldots, \hat{c}(L)\right]^{\mathrm{T}} \quad (2.16)$$

where L is the length of the feature vector. The parameter vector extracted from the sequence of HMMs is,

$$\mathbf{O} = \left[\mathbf{o}_1^{\mathrm{T}}, \mathbf{o}_2^{\mathrm{T}}, \ldots, \mathbf{o}_T^{\mathrm{T}}\right]^{\mathrm{T}} \quad (2.17)$$

where parameter vector $\mathbf{o}_t$ consists of the static feature vector $\mathbf{c}_t$ and dynamic feature vectors $\Delta\mathbf{c}_t$ and $\Delta^2\mathbf{c}_t$ (see Section 3.2) [3],

$$\mathbf{o}_t = \left[\mathbf{c}_t^{\mathrm{T}}, \Delta\mathbf{c}_t^{\mathrm{T}}, \Delta^2\mathbf{c}_t^{\mathrm{T}}\right]^{\mathrm{T}} \quad (2.18)$$

$$\mathbf{c}_t = \left[c(1), c(2), \ldots, c(L)\right]^{\mathrm{T}}$$

The optimal sequence $\hat{\mathbf{C}}$ can be obtained as the one that maximizes $P(\mathbf{O}|\lambda)$ with respect to the observations $\mathbf{O}$ and the sub-state sequence $\mathbf{Q}$:

$$P(\mathbf{O}|\lambda) = \sum_{\text{all } \mathbf{Q}} P(\mathbf{O}, \mathbf{Q}|\lambda) \quad (2.19)$$

where the sub-state sequence is defined as,

$$\mathbf{Q} = \{(q_1, s_1), (q_2, s_2), \ldots, (q_T, s_T)\} \quad (2.20)$$

and (q_t, s_t) represents being in state q_t and mixture s_t at time t.

There are different possible solutions in order to obtain the parameter vector sequence $\hat{\mathbf{C}}$, depending whether the sub-state sequence is given or it is part of the maximization constraints (see Section 2.5.1).

2.2.3 Types of HMM

Depending on the possible state transitions, an HMM can be ergodic or non-ergodic. In the former case, the model is fully connected, so each state can be reached from any state (see Figure 2.2). Although this model presents some advantages, a non-ergodic left-to-right HMM has been shown to present a better performance when used with speech signals. Figure 2.3 depicts this model adding two non-emitting states at the beginning and the end of the model in order to concatenate a sequence of HMMs.

Unlike the ergodic model, a left-to-right structure has the property to increase or keep the state

[3] As described in Section 3.2, each stream contains information of 3 windows (static, delta and delta delta), hence the total size of the observation is actually $3L$.

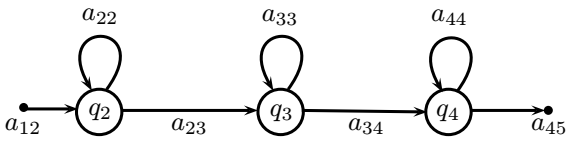


Figure 2.3: A non-ergodic HMM.

index as the time increases. This kind of HMM can readily model signals whose properties change over time in a successive manner, that is, speech.

There is no simple way of making an accurate choice of the model. The main parameters to take into account are the model size (number of states) and the observation symbols (number of multivariate Gaussians in each state). These choices must be made depending on the signal being modelled and its variability (e.g., more than one speaker would produce different information for a single vowel "a"). However, there are some basics considerations to take into account during the design of an HMM:

- The number of states (N) concerns the time axis. Each state of the HMM will model a region of certain units. Each state of the HMM is modelling a region of the time, that is, an integer number of frames. Typically, three or five states with a frame rate of 5ms to 10ms is used.
- The number of streams (S). Streams are used in order to simultaneously model different sources of information (e.g., vocal-tract and F0) in a single HMM framework. As an example of the purpose of modelling different types of data, streams were successfully applied to audio-visual speech recognition (Marcheret et al., 2004).
- The number of multivariate Gaussian distributions of each stream (M_s) depends on the diversity in each state. As the variation of the training data increases, so does M_s. This is crucial in speech recognition applications where the number of speakers tend to be very high for general applications. It becomes less important in speech synthesis where an speaker dependent application is usually needed. In practice, an HMM-based TTS system uses $M_s = 1$ though some experiments where conducted with more mixtures in a multilingual approach presented by (Latorre et al., 2006). In that system, the use of multiple mixture is justified since the application builds an average voice combining multiple speakers of different languages. The main drawback of increasing the number of mixtures is the complexity of the parameter generation during the synthesis stage (see Section 2.5.1).

2.3 HMM training for HMM-based TTS system

Like in any system involving HMMs, a training stage is needed (see Figure 2.4 which depicts the training workflow). The tool used for the training stage is (HTS, a), a modified version of HTK (Young

et al., 2006), a widely used toolkit originally developed for speech recognition.

Two phases are distinguished during training, the first one involves phonemes and the second one uses the so-called full contextual units, that is, a phoneme in a linguistic context (see Section 2.8 more a detailed description of the full context model).

Firstly, assuming that observations [4] have been extracted for each sentence, HMMs for isolated phonemes are initialized using labelled data. This step implies using a Viterbi-based estimation using the phoneme time boundaries [5].

Estimation of each of these models are used as a initialization for the full contextual units. From this point onwards, a embedded re-estimation process is used. Due to data sparsity, similar [6] re-estimated full context models are clustered by means of a decision tree using the contextual information and a set of binary linguistic questions (e.g., is right unit an "a" vowel? Is left unit an unvoiced consonant?). Embedded re-estimations guarantees the tuning of the HMMs and are used iteratively. Durations can be estimated in different ways depending on the type of duration model being used (see Section 2.4).

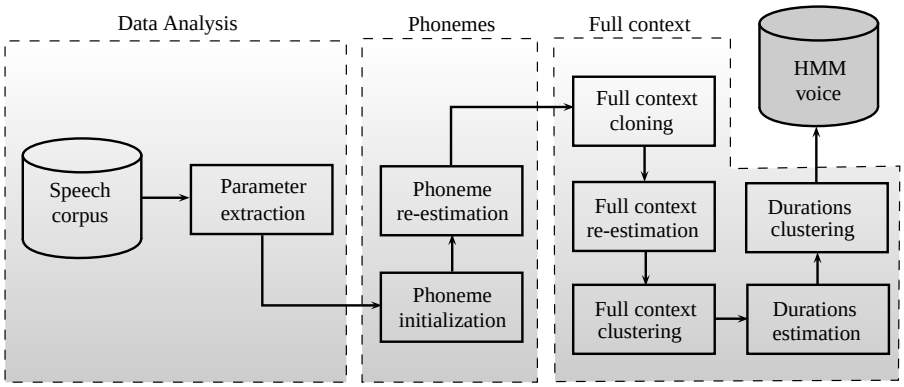


Figure 2.4: Training workflow of the HMM-TTS

Each full contextual HMM includes vocal tract (see Section 3.3), pitch (see Section 3.4) and mixed excitation parameters (see Section 3.5). Generally, each of these information parameters are divided in a different stream so they can be trained simultaneously in the same HMM (see Section 2.2.1). Table 2.1 summarizes the type of distribution for each of these streams where MGD stands for Multivariate Gaussian Distribution (which is a mixture of multidimensional Gaussians) and MSD for MultiSpace Distribution (see Section 3.4).

[4] In this case, observation refers to the set of parameters extracted from the speech as described in Chapter 3.

[5] There are two possible ways of initializing the models. On the one hand, a flat start would simply make use of an embedded re-estimation where time boundaries are not needed. This is often used in speech recognition where the number of models is much more smaller than the number of examples. In TTS systems, a flat start is not possible due to the sparsity of the full context models. In this case, each HMM is initialized individually using labelled data.

[6] Section 2.8 describes the clustering process by using decision trees. Basically, the tree is constructed in a top-down manner splitting the nodes choosing questions that maximize a probability until a stopping criterion is satisfied. Eventually, all models clustered under the same question will be therefore be similar and their mixtures will be shared.

Stream description	Parameters	Type of distribution
Vocal tract	mel-cepstral LPC LSP MFCC	MGD
Pitch	Fundamental frequency (F0) contour	MSD, MGD
Mixed excitation	Voicing strengths, aperiodicity, residual parameters	MGD

Table 2.1: HMM stream modelling. Each stream can modelled by any of the parameters using a certain type of distribution. Type of parameters are described in Chapter 3 while the type of distribution is always a MGD except for pitch (see Section 3.4).

A typical topology of the phonemes and full-context units is a 5 states left-to-right HMM with no-skips. According to table 2.1, output observations of each state are represented by 3 streams. Each type of parameter has a composite form containing their static vector along with their dynamic features (Δ and Δ^2 coefficients, see Section 3.2).

The training process depicted in Figure 2.4 is summarized as follows:

- **Prepare data**. Create observations for HMMs from speech frames. This process is identical to the one used for speech recognition or unit segmentation (a further description of data parameterization is in Chapter 3).

- **Prepare model**. 5 states left-to-right HMM with no-skips.

- **Model initialization**. At this stage there is an HMM for each phoneme. Using time boundaries extracted during unit segmentation, run Viterbi initialization and re-estimate.

- **Initialize full context units**. At this point, there is an HMM for every phoneme. Now we need to create an HMM for every full contextual unit. Since each contextual unit is a phoneme in a linguistic context, the corresponding phoneme HMM is used as the initial HMM model for every full contextual unit. Hence one phoneme HMM will be used by N contextual units.

- **Re-estimation**. Full contextual units are re-estimated using multiple instances of embedded EM algorithm.

- **Clustering**. Similar full contextual models are tied together so to make a robust training.

- **Duration estimation**. Depending on the duration model used (see Section 2.4), durations will be trained at this point (non-explicit model) or along with the previous stages (explicit modelling).

2.4 Duration modelling

Duration is a key aspect of an HMM-based TTS system because it controls the parameter generation algorithm (see Section 2.5.1). Once the duration is estimated, the synthesis system will use it as a count of the number of observations in every state.

State duration is not part of the previous model because this information is not modelled like the rest of the parameters. There are two basic constraints for the design of the duration model in synthesis. First, it should be dynamic so the system can set the speed of the synthetic speech maintaining the quality of the signal. Second, it should also be trainable. For these reasons and assuming that a full-context HMM has N states, durations are modelled by $N-2$ single Gaussian distributions, one for each emitting state.

In this section, different duration estimation algorithms are described. The simplest way is to model durations obtained by the Viterbi segmentation of the training data and model them with a single Gaussian distribution (Masuko et al., 1996b). This method is not very efficient since is very dependent on data sparsity and Gaussians can be poorly estimated if only a few observations are found in the training data. However, this method is useful to introduce other duration modelling approaches discussed here.

According to the definition of an HMM, the probability of a state sequence $\mathbf{Q} = \{q_1, q_2, \ldots, q_T\}$ being $q_t = \{q_t(i)\}$ the state i at time t is,

$$P(\mathbf{Q}|\lambda, T) = \prod_{k=1}^{K} p_k(d_k) \tag{2.21}$$

where $p_k(d_k)$ is the probability of being exactly d_k frames in state k and K is the total number of states visited during T frames:

$$\sum_{k=1}^{K} d_k = T \tag{2.22}$$

The inherent duration probability density $p_i(d_i)$ associated with state $q_t(i)$ with self transition probability a_{ii} is,

$$p_i(d_i) = (a_{ii})^{d_i - 1}(1 - a_{ii}) \tag{2.23}$$

where the probability of d_k consecutive observations in state $q_t(i)$ decreases exponentially with time.

From this equation one can see that the standard HMM does not differentiate between a self transition and a transition to another state when calculating the probability of an observation sequence. This yields an exponential probability function if the model remains in a certain state for a number of consecutive observations. Equation 2.23 is therefore inappropriate to model the temporal structure of speech signals (Theodoridis and Koutroumbas, 2006).

To control temporal structure appropriately, HMMs should have explicit state duration densities instead of self transition probabilities so the duration probability density $p_k(d_k)$ can be an arbitrary

distribution. From this point of view, depending on the type of duration model estimation, two approaches have been proposed in the literature: non-explicit and explicit state duration densities. The former technique (see Section 2.4.2) uses state occupancy at the last step of the embedded re-estimation and the latter (Section 2.4.3) uses the so-called Hidden semi-Markov Model (HSMM).

2.4.1 Gaussian modelling of duration for synthesis

Duration for state k is estimated by maximizing Equation 2.21 with respect to the sequence of states $\mathbf{Q}$ under the constraint of Equation 2.22. When the state duration probability density is modelled by a single Gaussian distribution,

$$p_k(d_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(d_k - \mu_k)^2}{2\sigma_k^2}\right)$$

the following equation is obtained by using Lagrange multipliers method (Masuko, 2002):

$$d_k = \mu_k + \rho\sigma_k^2 \tag{2.24}$$

where ρ controls the speaking rate of an utterance:

$$\rho = \frac{T - \sum_{k=1}^{K} \mu_k}{\sum_{k=1}^{T} \sigma_k^2} \tag{2.25}$$

In conclusion, the final duration of state k can be obtained using ρ in order to tune the speaking rate (Equation 2.24). When ρ is set to zero, the system uses the average speaking rate because durations will be based on the mean duration obtained during the training stage. When $\rho > 0$, the time in each state will increase so the speaking rate will be slower. The opposite effect can be applied for $\rho < 0$.

2.4.2 Non-explicit duration density

This technique (Yoshimura et al., 1998) [7] calculates Gaussian distributions of state durations on the trellis which is made in the embedded training stage, therefore the mean and variance of the Gaussian distribution are estimated from statistical variables obtained in the last iteration of the forward-backward algorithm.

Similarly to the EM algorithm used to estimate the HMM parameters, the mean ($\mu_d(i)$) and

[7] Although original paper by (Yoshimura et al., 1998) describes updating rules for means and variances, statistical representation of the state occupancy was corrected in the revision of the duration definition by (Zen et al., 2007b).

variance ($\sigma_d^2(i)$) for the duration of state i is obtained by the following Equations:

$$\mu_d(i) = \frac{\sum_{t_0=1}^{T}\sum_{t_1=t_0}^{T}\mathcal{X}_{t_0,t_1}(i)(t_1 - t_0 + 1)}{\sum_{t_0=1}^{T}\sum_{t_1=t_0}^{T}\mathcal{X}_{t_0,t_1}(i)} \tag{2.26}$$

$$\sigma_d^2(i) = \frac{\sum_{t_0=1}^{T}\sum_{t_1=t_0}^{T}\mathcal{X}_{t_0,t_1}(i)(t_1 - t_0 + 1)^2}{\sum_{t_0=1}^{T}\sum_{t_1=t_0}^{T}\mathcal{X}_{t_0,t_1}(i)} - \mu^{(d)}(i) \tag{2.27}$$

where $\mathcal{X}_{t_0,t_1}(i)$ is the probability of occupying state i from time t_0 to t_1,

$$\begin{aligned}\mathcal{X}_{t_0,t_1}(i) &= P(q_{t_0-1} \neq i, q_{t_0} = i, \ldots, q_{t_1} = i, q_{t_1+1} \neq i|\mathbf{O}, \lambda) \\ &= \frac{\left(\sum_{j\neq i}\alpha_j(t_0-1)a_{ji}\right)a_{ii}^{t_1-t_0}\prod_{t=t_0}^{t_1}b_i(\mathbf{o_t})\left(\sum_{k\neq i}a_{ik}b_k(\mathbf{o}_{t_1+1})\beta_k(t_1+1)\right)}{P(\mathbf{O}|\lambda)}\end{aligned} \tag{2.28}$$

2.4.3 Explicit duration density using Hidden semi-Markov Model (HSMM)

Using previous duration density estimation, speech is synthesized from HMMs with explicit state duration probability distributions but HMMs are still trained without them so there is an inherent inconsistency.

Multistream [8] Gaussian distribution for HSMMs (Zen et al., 2007b) can resolve this discrepancy, since they can be viewed as HMMs with explicit state duration probability density functions. It makes it possible to simultaneously re-estimate state output and duration probability density functions.

Using the conventional forward and backward algorithm (Equations 2.8 and 2.9), we redefine the forward and backward probabilities using the duration density (Rabiner, 1990; Zen et al., 2007b) as,

$$\alpha_j(t) = \sum_{\substack{i=2\\ i\neq j}}^{N-1}\sum_{d=1}^{t}\alpha_j(t-d)a_{ij}p_j(d)\prod_{s=t-d+1}^{t}b_j(\mathbf{o}_s) \tag{2.29}$$

$$\beta_j(t) = \sum_{\substack{i=2\\ i\neq j}}^{N-1}\sum_{d=1}^{T-t}\beta_j(t+d)a_{ij}p_j(d)\prod_{s=t+1}^{t+d}b_j(\mathbf{o}_s) \tag{2.30}$$

[8] Note that in this case, duration for each state is considered an independent process so instead of using multivariate Gaussian distributions, a multistream model is used. This is not a problem during synthesis time because multistream is treated as a multidimensional Gaussian for durations. Therefore Section 2.4.1 is still valid for HSMM.

where $p_j(d)$ is the state duration probability of the j-th state modelled by a Gaussian distribution. In addition, note that the sum is over all states and all possible state durations.

In consequence, by using these equations, the re-estimation of the model parameters must also be reformulated. Equations 2.11 and 2.12 now become,

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_j(t,d)\mathbf{o}_t}{\sum_{t=1}^{T}\sum_{t=1}^{T}\gamma_j(t,d)} \tag{2.31}$$

and

$$\hat{\mathbf{U}}_j = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_j(t,d)(\mathbf{o}_t-\boldsymbol{\mu}_j)(\mathbf{o}_t-\boldsymbol{\mu}_j)^{\mathrm{T}}}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_j(t,d)} \tag{2.32}$$

where $\gamma_j(t,d)$ is the probability of occupying state j, stream s and mixture m at time t during d so now Equation 2.14 takes into account the duration density.

2.4.4 Duration modelling: discussion

Both duration estimation techniques from previous sections use continuous Gaussian distributions. On the one hand, the technique described in Section 2.4.2 estimates the duration model during the last step of the re-estimation process estimating the duration from the state occupancy of each model. On the other hand, Hidden semi-Markov Model (HSMM) introduces the duration model into the HMM itself so the state duration probability density is part of the training stage.

Both system share the the following advantages:

- The speaking rate of the synthetic speech can be easily modified (by using ρ in Equation 2.25). The variance of the Gaussian distribution indicates the dynamic range centred to the mean. Durations are modified taking into account the range of values obtained during training in order to alleviate any possible distortion.
- Time boundaries are not needed since state duration densities can be estimated during training using the embedded re-estimation.

It seems obvious that the duration model obtained with HSMM will be more reliable since it will reflect the real duration behaviour and, in addition, the rest of the HMM parameters will be consistent with it. In fact, it improves the naturalness of the synthetic speech, not only for the duration but also for spectrum and F0 (Zen et al., 2007b). However, the main drawback of this approach is the increase of computationally load (forward-backward algorithm must be computed over all durations) and memory requirements (duration is an extra degree of freedom). The number of operations can be reduced by setting a maximum duration within each state. Nevertheless, during adaptation (see Section 2.7), Maximum Likelihood Linear Regression techniques can be applied straightforwardly to HSMM resulting in a better adaptation although its computational cost will be also higher.

Both systems have been used in different configurations for the experiments and it will be specified for each experiment in Chapter 5. It is noted that although every grain helps to make the tone, and every improvement in the HMM-based TTS system helps to make the synthesis more natural, explicit duration density using HSMM is not one of the fundamental parts studied in this thesis.

2.5 HMM-based speech synthesis

Figure 2.5 depicts the synthesis scheme. Once the system has been trained, an HMM voice consisting of a set of full context HMMs is produced.

In this stage, an arbitrarily given text to be synthesized is converted to a context based label sequence. Then, according to the label sequence, a chain of HMMs is constructed by concatenating the required context dependent HMMs. Decision trees guarantees that any label is linked with an existing full context HMM stage by descending the binary tree until a leaf node is reached.

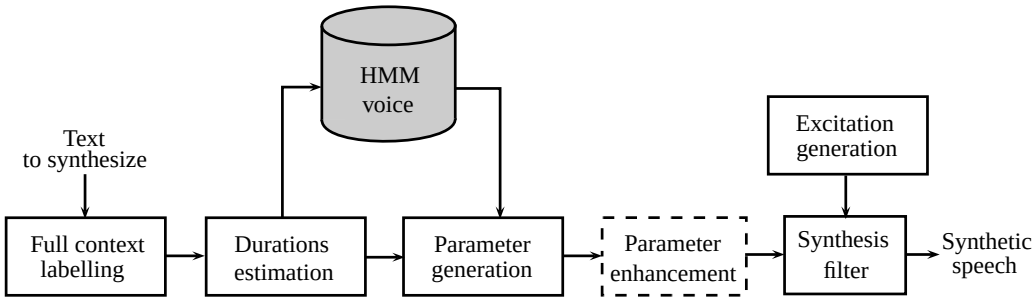


Figure 2.5: Synthesis workflow.

Firstly, HMM's duration are determined from the duration model (see Section 2.4). According to the duration of each state, a sequence of vocal-tract coefficients and pitch values including voiced/unvoiced decisions is generated from each HMM by using a speech parameter generation algorithm (see Section 2.5.1 and an illustrative example in Figure 2.6). The module in dotted lines is an optional step in which parameters from HMM are enhanced in order to alleviate the excessive muffling effect. A common weakness of this kind of system is the so-called over-smoothing and it is described in detail in Section 2.6.

Finally, speech is synthesized directly from the generated vocal-tract coefficients and excitation parameters using the synthesis filter (see Section 3.3.3). The excitation is built using the pitch and optionally some spectrum parameters (see how the excitation is created and types excitations in Section 3.5). In the source-filter model, this excitation is filtered and the vocal-tract coefficients are used as the filter coefficients.

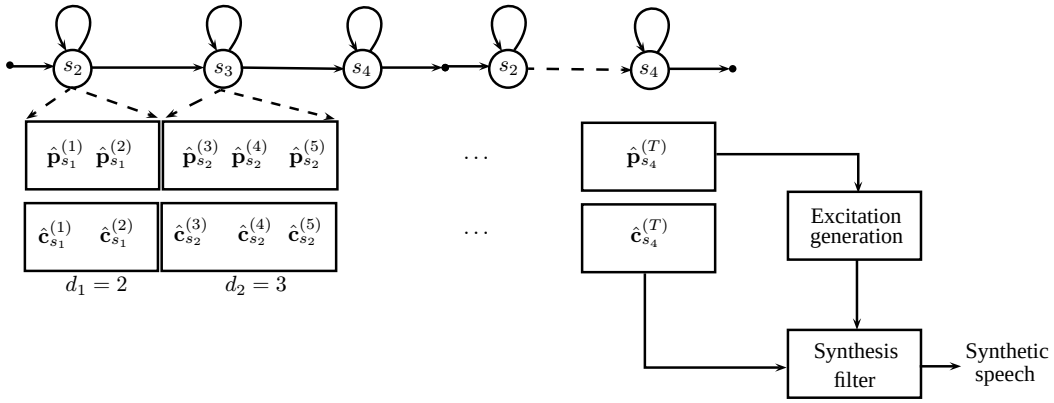


Figure 2.6: Illustrative example of a synthesis process extracting a certain number of observations from each state of the HMM. Pitch parameters for state s_i at time t of the current HMM stand for $\hat{\mathbf{p}}_{s_i}^{(t)}$ while vocal-tract is $\hat{\mathbf{c}}_{s_i}^{(t)}$.

2.5.1 Speech parameter generation

In order to convert the HMM parameters (i.e., means and covariances) into a sequence of speech samples, a parameter generation algorithm is used. In the current source-filter model, parameters generated from the HMM are vocal-tract, F0, mixed excitation and duration coefficients. These parameters are produced according to the static and dynamic features.

As we have introduced in Section 2.2.2, the problem of generating the parameter vector sequence $\hat{\mathbf{C}}$ is solved by maximizing the probability $P(\mathbf{O}|\lambda)$ in Equation 2.19. The solution is similar to the Viterbi algorithm and it employs a Maximum Likelihood Estimation (MLE) Criterion. Depending on the constraints of the system, three methodologies can be used (Tokuda et al., 2000) [9]:

Case 1. Maximize $P(\mathbf{O}|\mathbf{Q}, \lambda)$ for a given state sequence $\mathbf{Q}$ and model λ with respect to $\mathbf{O}$. This is the probability of observing a sequence of acoustic vectors given the state sequence. Its theoretical value is expressed as the probability of observing each observation sequence:

$$P(\mathbf{O}|\mathbf{Q}, \lambda) = b_{q_1 s_1}(\mathbf{o}_1) \cdot b_{q_2 s_2}(\mathbf{o}_2) \dots b_{q_T s_T}(\mathbf{o}_T) \tag{2.33}$$

in state q_t, mixture s_t at time t until the total number of observations T. Working with logarithmical probabilities Equation 2.33 becomes,

$$\log P(\mathbf{O}|\mathbf{Q}, \lambda) = \sum_{t=1}^{T} \log b_{q_t s_t}(\mathbf{o}_t) \tag{2.34}$$

Given the state sequence $\mathbf{Q}$, it is possible to assume without loss of generality that each

[9] Note that column vectors are assumed unless stated otherwise.

observation probability is a single [10] multidimensional Gaussian distribution of static and dynamic features of Equation 2.18,

$$\begin{aligned} b_j(\mathbf{o}_t) &= \mathcal{N}([\mathbf{c}_t^{\mathrm{T}}, \Delta\mathbf{c}_t^{\mathrm{T}}, \Delta^2\mathbf{c}_t^{\mathrm{T}}]^{\mathrm{T}}|\mathbf{\Gamma}_j, \mathbf{\Sigma}_j) \\ \mathbf{\Gamma}_j &= [\boldsymbol{\mu}_j^{\mathrm{T}}, \Delta\boldsymbol{\mu}_j^{\mathrm{T}}, \Delta^2\boldsymbol{\mu}_j^{\mathrm{T}}]^{\mathrm{T}} \\ \mathbf{\Sigma}_j &= \mathrm{diag}[\mathbf{U}_j, \Delta\mathbf{U}_j, \Delta^2\mathbf{U}_j] \end{aligned} \tag{2.35}$$

where diag[·] stands for the diagonal matrix and $\mathcal{N}(\cdot|\boldsymbol{\mu}_j, \mathbf{U}_j)$ is the generic representation of a Gaussian distribution for sub-state j with mean vector $\boldsymbol{\mu}_j$ and variance matrix $\mathbf{U}_j$. It is noted that $\mathbf{\Gamma}$ and $\mathbf{\Sigma}$ are composed of static and dynamic features.

$$\mathcal{N}(\mathbf{c}|\boldsymbol{\mu}_j, \mathbf{U}_j) = \frac{1}{\sqrt{(2\pi)^L|\mathbf{U}_j|}} \exp\left(-\frac{1}{2}\left(\mathbf{c}-\boldsymbol{\mu}_j\right)^{\mathrm{T}}\mathbf{U}_j^{-1}\left(\mathbf{c}-\boldsymbol{\mu}_j\right)\right) \tag{2.36}$$

Thus, logarithmic representation of Equation 2.35 is,

$$\log b_j(\mathbf{o}_t) = -\frac{1}{2}\left[L\log(2\pi) + \log|\mathbf{\Sigma}_j| + \left(\mathbf{o}_t-\mathbf{\Gamma}_j\right)^{\mathrm{T}}\mathbf{\Sigma}_j^{-1}\left(\mathbf{o}_t-\mathbf{\Gamma}_j\right)\right] \tag{2.37}$$

The relation between all observations ($\mathbf{O}$) and the static and dynamic features is obtained through a transformation matrix ($\mathbf{W}$):

$$\mathbf{O} = \mathbf{W}\mathbf{C} \tag{2.38}$$

Matrix $\mathbf{W}$ contains static and dynamic windows weights and it is defined as:

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T]^{\mathrm{T}} \tag{2.39}$$

where $\mathbf{w}_t$ is a $TL \times 3L$ matrix of at frame t defined as,

$$\mathbf{w}_t = \left[\mathbf{w}_t^{(0)}, \mathbf{w}_t^{(1)}, \mathbf{w}_t^{(2)}\right] \tag{2.40}$$

$$\begin{aligned} \mathbf{w}_t^{(n)} = \Bigg[& \underbrace{\mathbf{0}_{L\times L}}_{1st}, \ldots, \mathbf{0}_{L\times L}, \\ & \underbrace{w^{(n)}(-L_-^{(n)})\mathbf{I}_{L\times L}}_{(t-L_-^{(n)})-th}, \ldots, \underbrace{w^{(n)}(0)\mathbf{I}_{L\times L}}_{t-th}, \ldots, \underbrace{w^{(n)}(L_+^{(n)})\mathbf{I}_{L\times L}}_{(t+L_+^{(n)})-th} \\ & \mathbf{0}_{L\times L}, \ldots, \underbrace{\mathbf{0}_{L\times L}}_{T-th}\Bigg]^{\mathrm{T}} \end{aligned} \tag{2.41}$$

[10] For the purpose of this description, a single Gaussian distribution is assumed. This algorithm can be straightforwardly updated to use a mixture of M Gaussians.

where $n = 0$ stands for the static parameters, $n = \{1, 2\}$ stands for the dynamic features and $\mathbf{0}_{L\times L}$ and $\mathbf{I}_{L\times L}$ are the $L \times L$ zero matrix and the $L \times L$ identity matrix, respectively. It is assumed that $\mathbf{c}_t = \mathbf{0}_L$ for $t < 1$ and $t > T$ where $\mathbf{0}_L$ denotes the $L \times 1$ zero vector. Dynamic features and some examples of weights ($w^{(n)}(k)$) can be found in Section 3.2.

If we then substitute Equation 2.37 into Equation 2.34, we can generate an expression that gives us the log probability for the state sequence in terms of parameter vector $\mathbf{C}$.

$$\log P(\mathbf{O}|\mathbf{Q}, \lambda) = \underbrace{-\frac{TL}{2}\log(2\pi)}_{\text{1st term}} \underbrace{-\frac{1}{2}\sum_{t=1}^{T}\log|\mathbf{\Sigma}_{q_t s_t}|}_{\text{2nd term}} \underbrace{-\frac{1}{2}(\mathbf{WC}-\mathbf{\Gamma})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{WC}-\mathbf{\Gamma})}_{\text{3rd term}} \tag{2.42}$$

where:

- $\mathbf{W}$ is a $3TL \times TL$ matrix of dynamic window coefficients that defines the observations $\mathbf{O}$ (see Figure 2.7).
- $\mathbf{\Gamma} = \left[\mathbf{\Gamma}_{q_1 s_1}^{\mathrm{T}}, \mathbf{\Gamma}_{q_2 s_2}^{\mathrm{T}}, \ldots, \mathbf{\Gamma}_{q_T s_T}^{\mathrm{T}}\right]^{\mathrm{T}}$ is a $3TL \times 1$ vector of means
- $\mathbf{\Sigma} = \mathrm{diag}\left[\mathbf{\Sigma}_{q_1 s_1}, \mathbf{\Sigma}_{q_2 s_2}, \ldots, \mathbf{\Sigma}_{q_T s_T}\right]$ is the $3TL \times 3TL$ diagonal matrix of covariances.

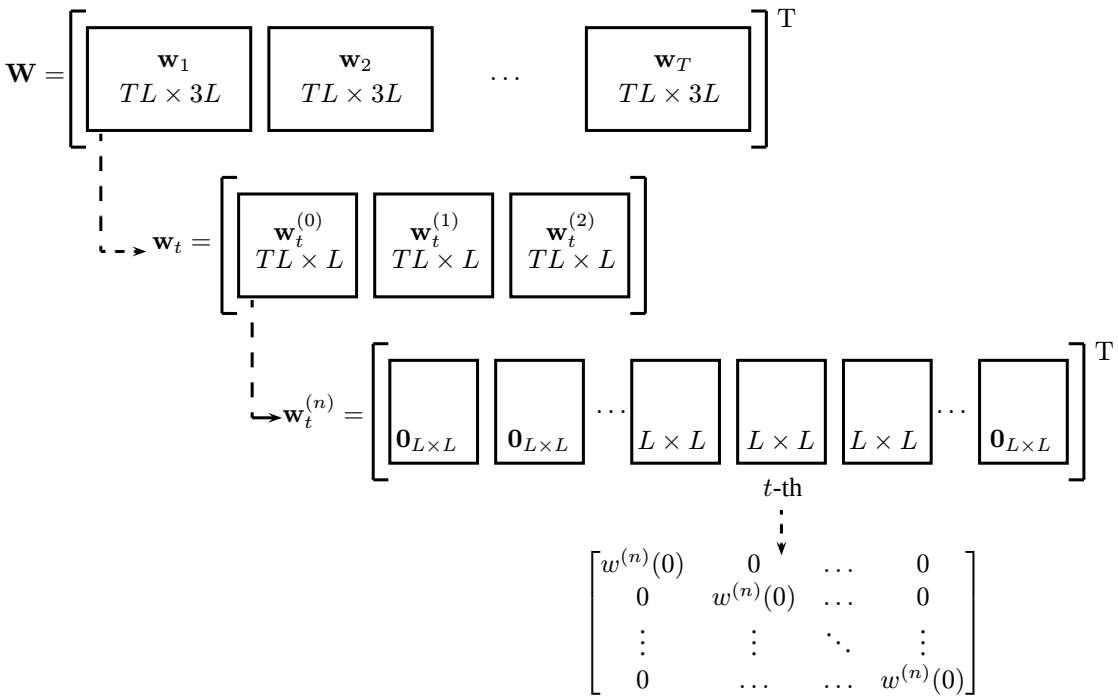


Figure 2.7: Matrix of static and dynamic features weights.

In order to find the feature vector sequence that maximizes the probability of Equation 2.42,

the following criterion must be satisfied,

$$\frac{\partial P(\mathbf{W}\hat{\mathbf{C}}|\mathbf{Q},\lambda)}{\partial \hat{\mathbf{C}}} = \mathbf{0}_{TL\times 1} \tag{2.43}$$

The third term of Equation 2.42 is the only one that depends on $\hat{\mathbf{C}}$ so,

$$\begin{aligned}\frac{\partial P(\mathbf{W}\hat{\mathbf{C}}|\mathbf{Q},\lambda)}{\partial \hat{\mathbf{C}}} &= -\frac{1}{2}\frac{\partial}{\partial \hat{\mathbf{C}}}(\mathbf{W}\hat{\mathbf{C}}-\mathbf{\Gamma})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{W}\hat{\mathbf{C}}-\mathbf{\Gamma}) \\ &= -\frac{1}{2}\frac{\partial}{\partial \hat{\mathbf{C}}}(\hat{\mathbf{C}}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}} - \hat{\mathbf{C}}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{\Gamma} - \mathbf{\Gamma}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}} + \mathbf{\Gamma}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{\Gamma}) \\ &= -\frac{1}{2}(2\mathbf{W}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}} - 2\mathbf{W}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{\Gamma}) \\ &= -\mathbf{W}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}} + \mathbf{W}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{\Gamma}\end{aligned} \tag{2.44}$$

The solution gives the following equation,

$$\hat{\mathbf{C}} = (\underbrace{\mathbf{W}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{W}}_{\mathbf{R}\ (TL\times TL)})^{-1}\underbrace{\mathbf{W}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{\Gamma}}_{\mathbf{r}\ (TL\times 1)} = \mathbf{R}^{-1}\mathbf{r} \tag{2.45}$$

If the dynamic features are not considered a constraint, the solution of equation 2.45 is directly $\hat{\mathbf{C}} = \boldsymbol{\mu}$, that is the mean values of the sub-state components. This would result on discontinuities in the generated spectral sequences between sub-states transitions (i.e., the generated parameters would look piece-wise rather than smooth, see an example in Figure 3.2).

Equation 2.45 can be solved by a efficient algorithm derived by (Tokuda et al., 1995; Tokuda, 1995) which can operate in a time-recursive manner. In practise, by using the special structure of $\mathbf{R}$, it can be solved by the Cholesky decomposition.

The Cholesky decomposition is mainly used for the numerical solution of linear equations such as $\mathbf{R}\hat{\mathbf{C}} = \mathbf{r}$ in Equation 2.45. Assuming that $\mathbf{R}$ is symmetric and positive definite, then we can solve the system by first computing the Cholesky decomposition $\mathbf{R} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, then solving $\mathbf{L}\mathbf{Y} = \mathbf{r}$ for $\mathbf{Y}$, and finally solving $\mathbf{L}^{\mathrm{T}}\hat{\mathbf{C}} = \mathbf{Y}$ for $\hat{\mathbf{C}}$.

Case 2. For a given λ, maximize $P(\mathbf{O},\mathbf{Q}|\lambda)$ with respect to $\mathbf{Q}$ and $\mathbf{O}$. In this case, the probability should be maximized for all possible state sequences $\mathbf{Q}$. However, this is impractical and a fast algorithm (Masuko, 2002) was developed for searching the optimal or sub-optimal state sequence keeping $\mathbf{C}$ optimal in the sense that $P(\mathbf{O}|\mathbf{Q},\lambda)$ is maximized with respect to $\mathbf{C}$.

Case 3. For a given λ, maximize $P(\mathbf{O},\lambda)$ with respect to $\mathbf{O}$. The algorithm is based on an EM method, which finds a critical point of the likelihood function $P(\mathbf{O}|\lambda)$. The algorithm is an iterative process that calculates a new features vector $\mathbf{C}'$, sets $\mathbf{C} = \mathbf{C}'$ and checks if a certain convergence condition is satisfied to stop. In each iteration, the occupancy probability is calculated with the forward-backward algorithm for the iterative modified $\mathbf{C}$.

Considering the HMM topology as an n-state left-to-right with no skips, the state sequence $\mathbf{Q}$ is straightforward to derive since each state appears in successive order and it is only necessary to estimate the duration of each sub-state. Besides of the increase of computational complexity of Case 2 and Case 3, it is not worth to apply these methods in an scenarios of a single Gaussian distribution. Therefore, Case 1 is the only method used in this work.

2.6 The problem of over-smoothing

All parameters needed to perform the synthesis using a source filter approach are generated using Case 1 explained in Section 2.5.1. Statistical averaging due to the modelling process improves the robustness against data sparseness and the use of the dynamic features constraint during the generation guarantees a smooth trajectory. However, the main disadvantage of the generation process is the lost characteristics of speech that cannot be recovered. The resulting speech sounds muffled and flat.

A wide range of solutions have been proposed in the literature to alleviate this problem. Basically, the main idea is to improve the generation of parameters in order to reduce the error (or increase the likelihood) between the generated parameters and the natural units. A further review of methods can be found by (Zen et al., 2009), and although they all have points in common, the following types can be categorized (Figure 2.8):

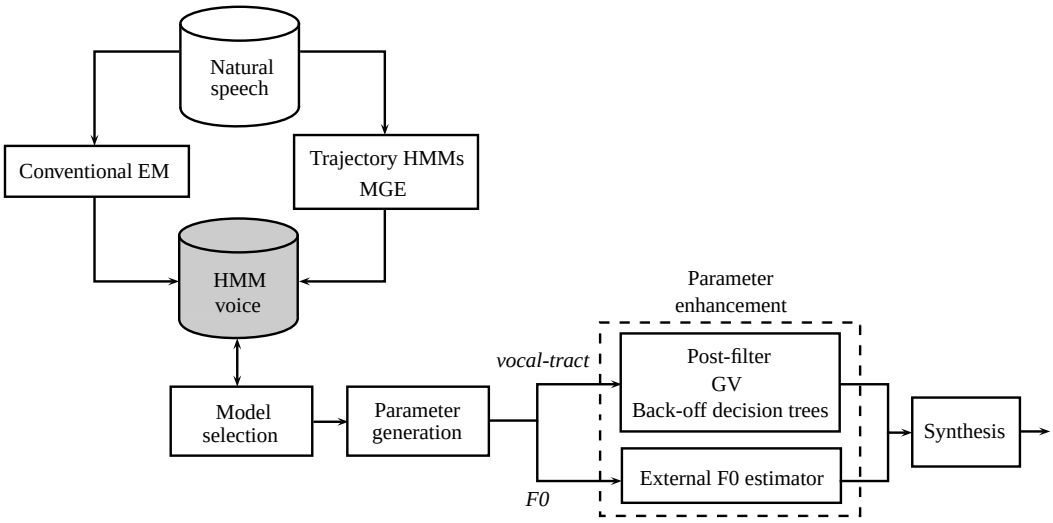


Figure 2.8: Categories to alleviate the over-smoothing problem.

- **Model of lost characteristics**. Assuming that the system models the vocal tract using a set of coefficients (i.e., any speech coding technique), the statistical training process generalizes the parameters through a model. Solutions in these category aim to improve the parameter

generation algorithm by modelling the information that is lost during the training in order to be reintroduced into the generated parameters. Depending on the type of information being modelled, the following systems have been proposed:

- **Post-filtering** (Yoshimura et al., 2001) is a known technique employed in many vocoders. The objective is to brighten the synthesized speech by emphasizing formants. Depending on the speech parameterization, a different post-filter mechanism will be used. See Section 3.3.3.1 for further details.
- **GV** (Toda and Tokuda, 2005, 2007) attempts to model the global variance of the coefficients of the natural units. This is one of the most successful methods in this category because of the low amount of extra information modelled and the low complexity during the synthesis process (see Section 2.6.1).
- **External F0 estimator** (Gonzalvo et al., 2007a). F0 and therefore expressivity is in fact one of the most affected parameters. The idea of this approach is to alleviate the over-smoothing by using an external F0 model. It is shown that this external model is more accurate than the F0 HMM in terms of absolute error with respect to the natural contours. This approach yields to a more expressive speech. In addition, it is also described an approach where F0 contours from the external model and the HMM model are blended. By merging both approaches, the system improves the expressivity while it guarantees the stability of the conventional F0 modelling with HMM (see Section 2.6.3 more a detailed description of the approach where the CBR system introduced in Section A.1 is used).
- **Back-off decision trees** (Kataoka et al., 2004) do not limit the size of the decision tree-based clustering during training considering that the lost of speech characteristics at the leaf nodes of these trees is zero. The system varies the size of phonetic decision trees dynamically at run-time according to the text to be synthesized undertaking the output probability of speech parameter trajectories.

- **Improved training**. Unlike the previous type that applies the solution during the synthesis stage, approaches in this category aim to enhance the training stage.
  - **Trajectory HMMs** (Zen et al., 2004). Although parameter generation in Section 2.5.1 works under the constraint of dynamic features, the HMM are trained without this assumption so there is an inconsistency [11]. Trajectory models introduce the dynamic features constraints into the training part of the HMMs.
  - **Minimum error training** (Wu and Wang, 2006; Wu et al., 2007; Wu and Tokuda, 2009). The idea is to modify the training stage in order to modify the Maximum Likelihood (ML) criterion introducing a new constraint with the sake of minimizing the error between the generated parameters and the real ones. A so-called MGE (Minimum Generation Error)

[11] Note that although dynamic features are part of the training observations, this does not imply that the EM algorithm differentiates which parameters correspond to each dynamic feature.

algorithm will update the HMM parameters by introducing the synthesis stage into the training process (see Section 2.6.2).

- **Hybrid systems**. Unlike the previous categories, solutions within this group make use of a complete different system. The resulting structure make use of an HMM system and usually a concatenative system (more information can be found in Chapter 4).

2.6.1 Global Variance (GV)

The GV algorithm is an extension of the conventional parameter generation algorithm described in Section 2.5.1 as Case 1. It is one of the most used solutions to tackle over-smoothing and it has been successfully applied to the HTS Nitech system in past and recent Blizzards (Zen et al., 2007a; Ling et al., 2006; Yamagishi et al., 2007).

The aim of GV is to improve the variability of the generated trajectories. In addition to the constraint of static and dynamic features in the conventional algorithm, GV also takes into account the global variance of the natural units. This algorithm is applied to vocal tract and F0.

The extended probability to be maximized taking into account the new constraint of the global variance is,

$$P(\mathbf{O}|\lambda,\lambda_v) = \sum_{\text{all } \mathbf{Q}} P(\mathbf{O}|\mathbf{Q},\lambda)^w \cdot P(\mathbf{v}(\hat{\mathbf{C}})|\lambda_v) \tag{2.46}$$

where:

- $\mathbf{v}(\hat{\mathbf{C}}) = [v(1), v(2), \ldots, v(L)]^{\mathrm{T}}$ is the vector of global variances of the generated parameters $\hat{\mathbf{C}} = \{\hat{\mathbf{c}}_1^{\mathrm{T}}, \hat{\mathbf{c}}_2^{\mathrm{T}}, \ldots, \hat{\mathbf{c}}_T^{\mathrm{T}}\}$ for a single utterance computed like:

$$v(l) = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{\mathbf{c}}_t(l) - \mu_{\hat{\mathbf{c}}(l)}\right)^2 \tag{2.47}$$

$$\mu_{\hat{\mathbf{c}}(l)} = \frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{c}}_t(l) \tag{2.48}$$

- λ_v is the global variance model and $P(\mathbf{v}(\hat{\mathbf{C}})|\lambda_v)$ is modelled by a single Gaussian distribution. Note that the global variance model will have a unique mean $\boldsymbol{\mu}_v = [\mu_v(1), \mu_v(2), \ldots, \mu_v(L)]^{\mathrm{T}}$ and diagonal covariance matrix $\mathbf{U}_v$ which are calculated from all natural units. It must not be confused with the variance of each utterance (referred in Equations 2.47 and 2.48).

$$P(\mathbf{v}(\hat{\mathbf{C}})|\lambda_v) = \frac{1}{\sqrt{(2\pi)^L|\mathbf{U}_v|}}\exp\left(-\frac{1}{2}\left(\mathbf{v}(\hat{\mathbf{C}}) - \boldsymbol{\mu}_v\right)^{\mathrm{T}}\mathbf{U}_v^{-1}\left(\mathbf{v}(\hat{\mathbf{C}}) - \boldsymbol{\mu}_v\right)\right) \tag{2.49}$$

- w is a constant that denotes the weight controlling a balance between the two probabilities. In practise, it is fixed to the ratio of the number of dimensions between vectors $\mathbf{v}(\hat{\mathbf{C}})$ and $\mathbf{O}$

(i.e., $1/3T$) as described by (Toda and Tokuda, 2007).

The likelihood to maximize is,

$$\mathcal{L} = P(\mathbf{O}|\mathbf{Q}, \lambda)^w \cdot P(\mathbf{v}(\hat{\mathbf{C}})|\lambda_v) \tag{2.50}$$

In order to determine $\hat{\mathbf{C}}$ that maximizes $\mathcal{L}$ in Equation 2.50, $\hat{\mathbf{C}}$ is iteratively updated with the gradient method, so for iteration $i+1$,

$$\hat{\mathbf{C}}^{(i+1)} = \hat{\mathbf{C}}^{(i)} + \alpha \cdot \delta\hat{\mathbf{C}}^{(i)} \tag{2.51}$$

where α is the step size which must be set to a small value.

The process of the GV algorithm can be summarized as follows:

1. First, parameters for the utterance to be synthesized are generated by the conventional algorithm resulting in $\hat{\mathbf{C}}$.

2. $\hat{\mathbf{C}}$ is updated in order to adapt its global variance $\mathbf{v}(\hat{\mathbf{C}})$ to the global variance of the model λ_v.

$$\begin{aligned} \hat{\mathbf{c}}_t^{(0)}(l) &= r \cdot \left(\hat{\mathbf{c}}_t(l) - \mu_{\hat{\mathbf{c}}(l)}\right) + \mu_{\hat{\mathbf{c}}(l)} \\ r &= \sqrt{\frac{\mu_v(l)}{\sigma^2_{\hat{\mathbf{c}}(l)}}} \end{aligned} \tag{2.52}$$

   where r is a ratio to convert the global variance of $\hat{\mathbf{C}}$ into the global variance of the model and $\sigma^2_{\hat{\mathbf{c}}(l)}$ is the global variance of the generated sequence $\hat{\mathbf{C}}$.

3. The iterative process runs for $i = \{1, \ldots, I\}$ using the initial value $\hat{\mathbf{C}}^{(0)}$. A gradient method is employed (Equation 2.51) in order to update the parameters until a stopping criterion is satisfied. Usually, the process stops if the number of steps exceeds a maximum value (I) or if the likelihood difference is below a certain threshold γ_g. Two methods are proposed by (Toda and Tokuda, 2007) in order to compute $\delta\hat{\mathbf{C}}^{(i)}$, one using the first derivative (steepest descent algorithm, Equation 2.58) and another using the second derivative (Newton-Raphson method, Equation 2.61). Basically, if the initial trajectory $\hat{\mathbf{C}}^{(0)}$ is to close to the optimum one, we may use the latter method.

   The first derivative can be obtained as follows. Firstly, by using logarithmic probabilities, Equation 2.50 becomes,

$$\begin{aligned} \mathcal{L} &= w\left(-\frac{1}{2}\hat{\mathbf{C}}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}} + \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}}\right) \\ &\quad -\frac{1}{2}\mathbf{v}(\hat{\mathbf{C}})^{\mathrm{T}}\mathbf{U}_v^{-1}\mathbf{v}(\hat{\mathbf{C}}) + \boldsymbol{\mu}_v^{\mathrm{T}}\mathbf{U}_v^{-1}\mathbf{v}(\hat{\mathbf{C}}) + \mathbf{K} \end{aligned} \tag{2.53}$$

where $\mathbf{K}$ are all the constants independent of $\hat{\mathbf{C}}$. The stop criterion using the threshold γ_g is the difference of likelihoods obtained using Equation 2.53 for each step,

$$\Delta\mathcal{L} = \mathcal{L}^{(i)} - \mathcal{L}^{(i-1)} < \gamma_g \tag{2.54}$$

Assuming that $\mathcal{L}$ in Equation 2.53 is composed by the likelihood of the HMM generation ($\mathcal{L}_h$) and the likelihood of GV ($\mathcal{L}_g$), $\mathcal{L}$ can be described as follows,

$$\mathcal{L} = \mathcal{L}_h + \mathcal{L}_g \tag{2.55}$$

$$\mathcal{L}_h = w\left(-\frac{1}{2}\hat{\mathbf{C}}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}} + \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}}\right) \tag{2.56}$$

$$\mathcal{L}_g(l) = -\frac{1}{2}(\sigma^2_{v(l)})^{-1}\sigma^2_{\hat{c}(l)}\left(\sigma^2_{\hat{c}(l)} - 2\mu_{v(l)}\right) \tag{2.57}$$

Using Equation 2.53,

$$\begin{aligned}\delta\hat{\mathbf{C}}^{(i)} = \left.\frac{\partial\mathcal{L}}{\partial\hat{\mathbf{C}}}\right|_{\hat{\mathbf{C}}=\hat{\mathbf{C}}^{(i)}} &= w\left(-\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}} + \mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}\right) \\ &\quad - \left(\mathbf{v}(\hat{\mathbf{C}})^{\mathrm{T}}\mathbf{U}_v^{-1}\frac{\partial\mathbf{v}(\hat{\mathbf{C}})}{\partial\hat{\mathbf{C}}} - \boldsymbol{\mu}_v^{\mathrm{T}}\mathbf{U}_v^{-1}\frac{\partial\mathbf{v}(\hat{\mathbf{C}})}{\partial\hat{\mathbf{C}}}\right) \\ &= w\left(-\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{W}\hat{\mathbf{C}} + \mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}\right) \\ &\quad \underbrace{- \left(\mathbf{v}(\hat{\mathbf{C}})^{\mathrm{T}} - \boldsymbol{\mu}_v^{\mathrm{T}}\right)\mathbf{U}_v^{-1}\frac{\partial\mathbf{v}(\hat{\mathbf{C}})}{\partial\hat{\mathbf{C}}}}_{\Upsilon_g}\end{aligned} \tag{2.58}$$

In order to obtain $\partial\mathbf{v}(\hat{\mathbf{C}})/\partial\hat{\mathbf{C}}$, we have to use the definition of Equation 2.47,

$$\frac{\delta v(l)}{\delta\hat{c}_t(l)} = \frac{2}{T}\sum_{t=1}^{T}\left(\hat{c}_t(l) - \mu_{\hat{c}(l)}\right) \tag{2.59}$$

then,

$$\Upsilon_g = -\frac{2(\sigma^2_{v(l)})^{-1}}{T}\left(\sigma^2_{\hat{\mathbf{c}}(l)} - \mu_{v_{(l)}}\right)\left(\sum_{t=1}^{T}\left(\hat{c}_t(l) - \mu_{\hat{\mathbf{c}}_{(l)}}\right)\right) \tag{2.60}$$

The second derivative is obtained as Equation 2.61 and it is fully described by (Toda and Tokuda, 2007).

$$\delta\hat{\mathbf{C}}^{(i)} = -\left(\frac{\partial^2\mathcal{L}}{\partial\hat{\mathbf{C}}\partial\hat{\mathbf{C}}^{\mathrm{T}}}\right)^{-1}\left.\frac{\partial\mathcal{L}}{\partial\hat{\mathbf{C}}}\right|_{\hat{\mathbf{C}}=\hat{\mathbf{C}}^{(i)}} \tag{2.61}$$

The GV algorithm improves the quality of the synthetic speech by approaching the generated parameters to the natural ones, specially the vocal tract coefficients. The resulting quality is increased beyond the performance of conventional post-filtering algorithm (see Section 3.3.3) while the expressiveness (that resulting of improving the F0 contour) is not affected in the same manner.

One possible reason is automatic F0 extraction errors, specially halving and doubling. Another possibility is that the GV criterion works on a frame basis while the F0 might be enhanced as a contour (this is the starting point of Section 2.6.3).

2.6.2 Minimum Generation Error (MGE)

MGE is another technique to minimize the over-smoothing effect. Unlike the GV algorithm presented in the previous section, MGE approach aims to minimize the error of the HMMs during the training stage (Wu and Wang, 2006; Wu et al., 2007). It attempts to solve two problems: (a) inconsistencies between training and synthesis since the conventional method for training is based on the ML criterion and it is not designed for synthesis applications; (b) a lack of mutual constraints between static and dynamic features which are only considered in the generation algorithm but not in the training procedure.

As we have just pointed out, MGE criterion was proposed to solve the issues related to ML-based HMM training in HMM-based speech synthesis. The approach described in this section uses the Euclidean distance to define the generation error between the original and generated LSP coefficients. However, recent advances improve the MGE criterion by imposing a log spectral distortion (LSD) (Wu and Tokuda, 2009). In addition, MGE has also been used to introduce an MGE linear regression based model adaptation algorithm, where the regression matrices used to transform source models are optimized so as to minimize the generation errors of adaptation data (Qin et al., 2008). Note that MGE has been mostly tested using LSP parameters (Wu and Wang, 2006) rather than mel-ceptral coefficients.

In the following paragraphs, two approaches will be described. The first one describes parameter updating and the next one deems the use of the MGE as a criterion for tree-based clustering.

2.6.2.1 Parameter updating

A first solution was proposed by (Zen et al., 2004) where a trajectory model was introduced into the HMM-based TTS synthesis system. Although the new training criterion implied a constraint between static and dynamic features, the HMM training was still under the ML framework.

The solution proposed in the MGE approach (Wu and Wang, 2006) is based on minimizing the error of the HMM prediction during the training stage. By incorporating the parameter generation algorithm into the training stage, the inconsistency between training and generation is eliminated and the constraints between static and dynamic features are considered in HMM training. This algorithm can either be used as a post-processing of the trained models using the conventional ML criterion and a distance definition (Wu and Wang, 2006) or it can also be extended as a criterion for tree-based clustering of context dependent HMMs (Wu et al., 2007).

MGE algorithm is described to work using a Generalized Probabilistic Descent (GPD) criterion (McDermott, 2000). The basic form of the GPD applied to update the HMMs has the following

form,

$$\lambda(n+1) = \lambda(n) - \varepsilon(n) \cdot \mathbf{S}_n \cdot \nabla \ell(\mathbf{C}_n, \lambda)|_{\lambda=\lambda_n} \tag{2.62}$$

where λ represents the HMM parameters, n is the sample counter (i.e., the current utterance), $\varepsilon(n)$ is the learning factor controlling the speed and accuracy of the convergence process, $\mathbf{S}_n$ is a definite positive matrix (which in practise can be set to the identity matrix) and $\ell(\mathbf{C}_n, \lambda)$ is the cost function defining the generation error.

Firstly, an appropriate objective measure for the cost function ℓ must be defined and it must be related with the generation error. Assuming we are using the optimal state sequence $\mathbf{Q}$, the cost function can be defined as the distance with respect to the original data. Usually, the Euclidean Squared distance is used,

$$\ell(\mathbf{C}, \lambda) = D(\hat{\mathbf{C}}, \mathbf{C}) = \left\| \hat{\mathbf{C}} - \mathbf{C} \right\|^2 = \left\| \sum_{t=1}^{T} (\hat{\mathbf{c}}_t - \mathbf{c}_t) \right\|^2 \tag{2.63}$$

By using the cost function of Equation 2.63 into Equation 2.62, we can define the corresponding updating criterion as,

$$\lambda(n+1) = \lambda(n) - \varepsilon(n) \left. \frac{\partial \ell(\mathbf{C}_n, \lambda)}{\partial \lambda} \right|_{\lambda=\lambda(n)} \tag{2.64}$$

The models during sample n are updated using the first derivative of the error with respect to the models. Assuming that the models use a single Gaussian distribution and considering Equation 2.45, we can derive the formulas for the mean and variance (see Section B for a complete description of this algorithm),

$$\frac{\partial \hat{\mathbf{C}}}{\partial \mu_{tj}} = \mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}} \mathbf{\Sigma}^{-1} \boldsymbol{\psi}^{(tj)} \tag{2.65}$$

$$\frac{\partial \hat{\mathbf{C}}}{\partial v_{tj}} = \mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}} \mathbf{\Psi}^{(tj)} \left(-\mathbf{W} \hat{\mathbf{C}} + \mathbf{\Gamma} \right) \tag{2.66}$$

where μ_{tj} and v_{tj} is the j-th dimension associated to the t-th frame of the mean and the inverse variance ($v_{tj} = {}^1\!/_{\sigma_{tj}^2}$), respectively. Selection of specific means and variances is done using $\boldsymbol{\psi}^{(tj)}$ and $\mathbf{\Psi}^{(tj)}$. The former is the elementary vector, one at $(3(t-1)L + j) \in [1, 3TL]$ and zero elsewhere. Similarly, the latter is the elementary matrix with dimension $3TL \times 3TL$.

If we then substitute Equations 2.65 and 2.66 into Equation 2.64 we obtain the updating rules for the mean and the variances:

$$\mu_{tj}(n+1) = \mu_{tj}(n) - 2\varepsilon(n) \left(\hat{\mathbf{C}} - \mathbf{C} \right)^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}} \mathbf{\Sigma}^{-1} \boldsymbol{\psi}^{(tj)} \tag{2.67}$$

$$v_{tj}(n+1) = v_{tj}(n) - 2\varepsilon(n) \left(\hat{\mathbf{C}} - \mathbf{C} \right)^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}} \mathbf{\Psi}^{(tj)} \left(-\mathbf{W} \hat{\mathbf{C}} + \mathbf{\Gamma} \right) \tag{2.68}$$

The learning rate for each utterance $\varepsilon(n)$ must be set to a small positive value fulfilling:

$$\sum_{n=1}^{\infty} \varepsilon(n) \rightarrow \infty \qquad \sum_{n=1}^{\infty} \varepsilon^2(n) < \infty \tag{2.69}$$

2.6.2.2 Tree-based context clustering

Since the parameter updating rules of the previous section reduce the error of the HMMs during the training stage, a step further is to use the MGE criterion within the clustering framework. The computational cost of applying MGE during clustering is computationally very expensive, so a set of simplified rules are presented by (Wu et al., 2007). By considering all training samples in a batch mode, the updating rule of Equation 2.64 becomes,

$$\lambda^{(new)} = \lambda^{(old)} - \varepsilon \sum_{n=1}^{N} \sum_{t=1}^{T_n} \varphi(n,t) \left. \frac{\partial \ell(\mathbf{C}_n, \lambda)}{\partial \lambda} \right|_{\lambda=\lambda_{old}} \tag{2.70}$$

where T_n is the total number of frames in utterance n and $\varphi(n,t)$ is used to activate the updating process when model λ is used for utterance n and frame t (see Section B.3 for a full description of this reduced approach).

Assuming that $\mathbf{WW^T}$ is a quasi-diagonal matrix and diagonal dominant matrix, the following approximation can be made, $\mathbf{WW^T} \approx \mathbf{I}$. With this assumption, the updating rules in Equations 2.67 and 2.68 can be simplified as:

$$\mu_{ij}^{(new)} = \mu_{ij}^{(old)} - 2\varepsilon \sum_{n=1}^{N} \sum_{t=1}^{T_n} \varphi_i(n,t) \left(\hat{o}_{nt}(j) - o_{nt}(j)\right) \tag{2.71}$$

$$v_{ij}^{(new)} = v_{ij}^{(old)} - \frac{2\varepsilon}{v_{ij}^{(old)}} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \varphi_i(n,t) \left(\hat{o}_{nt}(j) - o_{nt}(j)\right) \left(\mu_{ij}^{(old)} - o_{nt}(j)\right) \tag{2.72}$$

where μ_{ij} and v_{ij} are the j-th coefficient of the mean and the inverse variance of model i. Note that in this case, the parameters are not straightforwardly related to frame t since we are using N utterances simultaneously. Moreover, in this case, $\hat{o}_{nt}(j)$ and $o_{nt}(j)$ are the j-th dimension of observation from utterance n at frame t generated by the HMM and natural reference, respectively.

2.6.3 F0 enhancement through CBR external estimator

The over-smoothing problem does not only affect the vocal-tract but also any parameter being modelled and generated within the HMM framework. Global Variance in Section 2.6.1 improves the naturalness of the synthetic speech but not the expressiveness. This is because GV enhances the parameters by a frame basis although the F0 should be enhanced with respect to the whole utterance (that is, the whole contour).

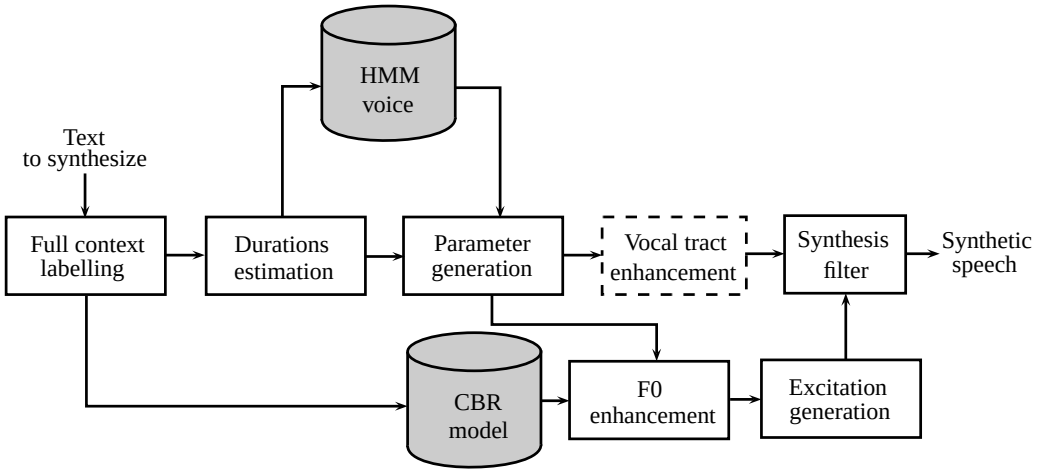


Figure 2.9: Synthesis workflow with an external F0 estimator using a CBR model.

In order to improve the expressiveness, another possible solution is to use an external F0 estimator during synthesis time. In this case, following what we planned as an objective, we propose an enhancing F0 technique using an external F0 CBR estimator (Gonzalvo et al., 2007a,b). CBR-based F0 estimator is a technique originally proposed by (Iriondo et al., 2006) in order to obtain reliable and expressive prosody estimations for emotion modelling. What makes CBR a better F0 estimator (refer to experiment in Section 5.3.1) can be explained by two reasons. Firstly, as we introduced in this section, HMM models F0 in a frame-by-frame basis whereas CBR models the F0 contour in longer blocks. In particular, as described in Section A.1, CBR estimates a F0 based on groups of syllables. Secondly, although CBR clusters similar training cases, it does not use the HMM-based clustering process which strongly averages the HMM parameters in a small set of groups. See Section A.1 to see further details of the CBR-based F0 estimator and its training procedure.

As a consequence, the synthesis scheme is modified and is depicted in Figure 2.9, where the F0 enhancement block merges the F0 generated by the HMMs and the one estimated from the external CBR module. The merging is performed using the idea from the F0 contour smoothing of Section A.2 and a single step of the process described for GV (see Section 2.6.1). Thus, the final F0 for frame t in an utterance of length T is,

$$f_0(t) = \underbrace{\tilde{f}_0^h(t) + \mu_h}_{\text{Default}} + \underbrace{\alpha_\sigma \left(\tilde{f}_0^c(t) - \tilde{f}_0^h(t) \right)}_{\text{Contour enhancement}} + \underbrace{\alpha_\mu \left(\mu_c - \mu_h \right)}_{\text{Mean tuning}} \tag{2.73}$$

where $\tilde{f}_0^h$ and $\tilde{f}_0^c$ denote HMM and CBR F0 contour variations respectively and μ_h, μ_c are the means

of the original contours defined as,

$$\tilde{f}_0^x(t) = f_0^x(t) - \mu_x$$
$$\mu_x = \frac{1}{T}\sum_{t=1}^{T} f_0^x(t)$$

where $x \in \{c, h\}$.

Equation 2.73 is divided in three terms. By default, the final F0 contour $f_0(t)$ is the HMM curve (defined by the first element of the Equation labelled as Default). The second and third terms adapt the variation and mean of the final contour, respectively, and the intensity of each one is controlled by two weights $\alpha_\sigma, \alpha_\mu \in [0, 1]$. Then, in order to enhance the contour, the weighted mixture of the HMM and the CBR curves is added. As with the contour, the offset of the final F0 contour can also be tuned.

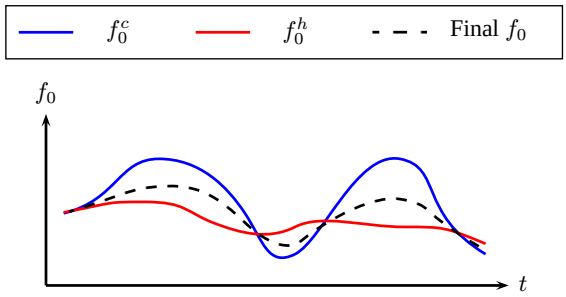


Figure 2.10: Example of a F0 contour merging with HMM and CBR systems.

Weights $\alpha_\sigma, \alpha_\mu$ can be empirically set from the training data. Obviously when $\alpha_\sigma = \alpha_\mu = 0$ the F0 curve will be fully generated from the HMM. For $\alpha_\sigma > 0, \alpha_\mu = 0$ only the shape variations of the contour will be enhanced (see Figure 2.10).

Note that the value of α_μ is used in order to restrict high values of F0, which can produce distortion in the HMM synthesis. The following balance can be used in order to avoid high pitch values (e.g., children, women or very expressive data): $\alpha_\mu < \alpha_\sigma$. Originally, HMM is clustering vocal-tract and F0 parameters. In this case, F0 is enhanced to external references which has not been simultaneously trained and in consequence, high F0 values can distort the synthetic speech producing a harsh quality. Equation 2.73 guarantees that means and variances can be modified independently so undesired artifacts can be alleviated without losing the expressiveness enhancement.

2.7 HMM adaptation

In general, speech synthesis systems are desirable to be used with arbitrary speaker characteristics and speaking styles. In the HMM-based speech synthesis method, we can easily change spectral and

prosodic characteristics of synthetic speech by transforming HMM parameters appropriately since speech parameters used in the synthesis stage are statistically modeled by using the framework of the HMM. The technique to perform this transformations is the so-called HMM adaptation. The effect of these transformations is to shift the components in the initial system sot that each state in the HMM system is more likely to generate the adaptation data.

Originally designed for robust speech recognition, HMM adaptation was primarily used to train average models that could be adapted to specific environments and speakers. Similarly, adaptation for speech synthesis was adopted. On the one hand, speaker adaptation with average voice model (see Section 2.7.3) becomes promising when available speech data of a target speaker is limited. On the other hand, the average voice model can be utilized to provide robust basis useful for synthesizing speech of the new target speaker. As a result, stable synthetic speech can be obtained even if speech samples available for the target speaker are very small. In addition, a synthesizer adaptation must be able to convert both voice characteristics and prosodic features such as F0 and phone duration. Therefore, an HSMM framework (See Section 2.4) is described for the adaptation algorithms.

The basic applications of adaptation can be classified in the following groups:

- **Speaker conversion**. This can either be style adaptation (e.g., emotions) or speaker identity conversion (e.g., speaker S_A adapted to speaker S_B). Both applications can be solved identically. An original voice is adapted to a target voice using a limited amount of adaptation data. A specific case of style adaptation is emotion transformations. Current state-of-the-art text-to-speech (TTS) systems are often able to produce intelligible and natural speech, but the speaking style is typically restricted to neutral style and so does not exhibit the full range of expressivity present in natural speech. Hence the development of more expressive and natural speech synthesis systems is becoming an important research area. Approaches to this reflect the two main synthesis techniques today. One approach is to use concatenative speech synthesis, which produces good quality but requires a large amount of data for each emotion or expressive style (Bulut et al., 2002; Montero et al., 1999). On the other hand, we can use a modelling approach and transform the synthesized speech into a target emotion by modelling changes in prosody and voice quality (VoQ) (Inanoglu and Young, 2007). Currently, significant recent improvements in HMM-based synthesis has been able to create a speaker independent voice, trained from multiple speakers, that can be later transformed to a single speaker's voice by using speaker adaptation techniques on a small amount of data (Tamura et al., 2001). These same techniques can also be used to adapt to a target emotion style (Yamagishi et al., 2004).

  In order to reproduce a speaking style for a specific emotion and maintain naturalness, it is necessary to control prosodic and spectral features, and so HMM speech synthesis (Yoshimura et al., 1999) is ideally suited to meet this criteria. An example of emotion conversion is shown in the Experiments Section 5.4.1.

- **Multilingual system**. There exists two main types of multilingual systems (Traber et al., 1999). On the one hand, a *multilingual* approach is described as a TTS system using different

languages simultaneously switching between languages on demand. On the other hand, a TTS is considered to be *polyglot* if it can synthesize several languages using the same voice identity with the appropriate pronunciation rules while maintaining the quality.

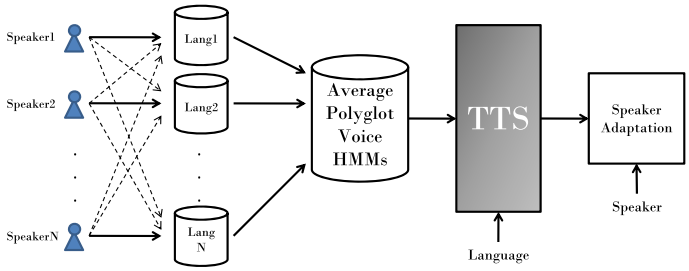


Figure 2.11: Polyglot HMM-TTS system based on voice adaptation. Several speakers are used to train different languages. Dotted lines from speakers stand for possible speakers being able to speak more than 1 language.

Hence, a polyglot system based on HMM (see figure 2.11) can synthesize language A (L_A) and language B (L_B) with the voice of a L_A speaker who does not speak L_B (Latorre et al., 2006). Polyglot HMM-based TTS systems combine corpora from several speakers and languages and adapts a voice to a target speaker. Taking advantage of the HMM properties, sounds of non-included units can be approximated between languages by those available in the polyglot training data.

- **Interpolation** (Yoshimura et al., 2000). The purpose is to synthesize speech with untrained speaker's characteristics by interpolating HMM parameters among some representative speakers' HMM sets.

Main adaptation techniques are Linear Regression (LR) (Leggetter and Woodland, 1995) and Maximum a Posteriori (MAP) (G. and Lee, 1994), whereas a common approach is to combine both approaches (Digalakis and Neumeyer, 1996; Yamagishi. et al., 2006) where MAP gives consistently to LR.

Maximum Likelihood Linear Regression (MLLR) (Leggetter and Woodland, 1995), is the basic model-space [12] linear regression algorithm for adaptation. The mean vectors of state output and duration distributions for the speaker are obtained by linearly transforming mean vector of state output and duration distributions of the average voice model. In some cases, it is not only necessary to adapt the means, but also the variances. In that case, constrained and unconstrained forms of transformations are distinguished depending to the form of variance transform. In the Constrained MLLR (CMLLR) (Gales, 1997; Yamagishi. et al., 2006), mean vectors and covariance matrices of the state output distributions are transformed simultaneously using the same matrix [13]. In the

[12] Usually in the literature, two types of linear transformations are distinguished. Adaptation applied either to model-space or feature-space. The former acts on the HMM parameters and the latter transforms the observations.

[13] In addition, as it will be shown in the description of the algorithm, in the constrained case of MLLR model-space and feature-space transformations are equivalent.

unconstrained case, the transformations of the mean and variances are unrelated to each other and the algorithm becomes increasingly complex.

2.7.1 CMLLR

Assuming the vector of observations and duration densities are modelled by multivariate Gaussian distributions, the adapted output distribution for the i-th state are defined as,

$$b_i(\mathbf{o}) = \mathcal{N}(\mathbf{o}; \hat{\boldsymbol{\mu}}_i, \hat{\mathbf{U}}_i) \tag{2.74}$$
$$p_i(d) = \mathcal{N}(d; \hat{m}_i, \hat{\sigma}_i^2) \tag{2.75}$$

where the transformed mean and covariance share a common transformation matrix $\boldsymbol{\zeta}'$ for the state output,

$$\hat{\boldsymbol{\mu}}_i = \boldsymbol{\zeta}'\boldsymbol{\mu}_i - \boldsymbol{\epsilon}' \tag{2.76}$$
$$\hat{\mathbf{U}}_i = \boldsymbol{\zeta}'\mathbf{U}_i\boldsymbol{\zeta}'^{\mathrm{T}} \tag{2.77}$$

and also a common transformation for the mean and covariances of the durations,

$$\hat{m}_i = \chi m_i - \nu \tag{2.78}$$
$$\hat{\sigma}_i^2 = \chi'\sigma_i^2\chi' \tag{2.79}$$

These transformations have the following structure:

- $\boldsymbol{\zeta}'$ is a $L \times L$ matrix used to transform both the mean vector and covariance matrix of the state output distribution.
- χ' is a scalar used to transform the mean and covariance of the duration distribution.
- The bias of the transformations is the vector $\boldsymbol{\epsilon}'$ of length L for the state output distribution and the scalar ν for the duration, respectively.

Although the CMLLR is defined as a model-space transformation technique, this technique is equivalent to a feature-space transform of the feature vector $\mathbf{o}$ and duration d in state i,

$$\begin{aligned} b_i(\mathbf{o}) &= \mathcal{N}(\mathbf{o}; \boldsymbol{\zeta}'\boldsymbol{\mu}_i - \boldsymbol{\epsilon}', \boldsymbol{\zeta}'\mathbf{U}_i\boldsymbol{\zeta}'^{\mathrm{T}}) = |\boldsymbol{\zeta}|\mathcal{N}(\boldsymbol{\zeta}\mathbf{o} + \boldsymbol{\epsilon}; \boldsymbol{\mu}_i, \mathbf{U}_i) = |\boldsymbol{\zeta}|\mathcal{N}(\mathbf{W}\boldsymbol{\xi}; \boldsymbol{\mu}_i, \mathbf{U}_i) \\ p_i(d) &= \mathcal{N}(d; \chi' m_i - \nu, \chi'\sigma_i^2\chi') = |\chi|\mathcal{N}(\chi d + \nu; m_i, \sigma_i^2) = |\chi|\mathcal{N}(\mathbf{X}\boldsymbol{\phi}; m_i, \sigma_i^2) \end{aligned} \tag{2.80}$$

where:

- $\boldsymbol{\xi} = [\mathbf{o}^{\mathrm{T}}, 1]^{\mathrm{T}}$ is the extended vector of observations with dimension $(L+1) \times 1$.

- $\boldsymbol{\phi} = [d, 1]^{\mathrm{T}}$ is the extended vector of durations observations.
- $\mathbf{W} = [\boldsymbol{\zeta}, \boldsymbol{\epsilon}]$ is the $L \times (L + 1)$ extended transformation matrix with a linear transformation $\boldsymbol{\zeta}$ and a bias vector $\boldsymbol{\epsilon}$.
- $\mathbf{X} = [\chi, \nu]$ is the transformation vector for durations.

2.7.1.1 Tying transformation matrices

In general, it is not always possible to estimate the linear regression transformation matrices $\mathbf{W}$ for every distribution because the amount of adaptation data of a target speaker is often small and the generalized inverse method drastically decreases the accuracy of the transformation matrix. Therefore, the first requirement to allow adaptation is to specify the set of the components that share the same transform.

Commonly, techniques to group similar distributions are based on binary trees. Specifically in this section, a regression class tree is used to group the Gaussians in the model set, so that the set of transformations to be estimated can be chosen according to the amount and type of adaptation data that is available. The tying of each transformation across a number of mixture components makes it possible to adapt distributions for which there were no observations at all. The regression class tree is constructed so as to cluster together components that are close in acoustic space, so that similar components can be transformed in a similar way (Young et al., 2006). Note that the tree is built using the original speaker independent model set, and is thus independent of any new speaker. In consequence, similar mixture components share the same transformation and transformations are estimated according with the amount of available target data.

In this tree structure (see Figure 2.12) each node specifies a particular cluster of distributions in the model, and those nodes that have a state occupancy count below a given threshold are placed in the same regression class as that of their parent node. The terminal nodes of the tree that specify the final component groupings are termed the base (regression) classes. Each Gaussian component of a model set belongs to one particular base class.

The tree is constructed with a centroid splitting algorithm, which uses an Euclidean distance measure and it is grown with a number of terminals or leaf nodes to cluster the mixture components of the model set. Algorithm 1 proceeds until the requested number of terminals has been achieved.

To apply the transformation, a top-down approach is used to traverse the regression class tree from the root node and progresses down the tree generating transforms only for those nodes which

1. have sufficient data and
2. are either terminal nodes (i.e. base classes) or have any children without sufficient data.

Algorithm 1 Algorithm to create a regression class tree.

1. Select a terminal node that is to be split.
2. Calculate the mean and variance from the mixture components clustered at this node.
3. Create two children. Initialise their means to the parent mean perturbed in opposite directions (for each child) by a fraction of the variance.
4. For each component at the parent node assign the component to one of the children by using a Euclidean distance measure to ascertain which child mean the component is closest to.
5. Once all the components have been assigned, calculate the new means for the children, based on the component assignments.
6. Keep re-assigning components to the children and re-estimating the child means until there is no change in assignments from one iteration to the next. Now finalise the split.

2.7.1.2 Parameter estimation

The objective is to estimate a set of transforms $\tilde{\Lambda} = \left(\tilde{\mathbf{W}}, \tilde{\mathbf{X}}\right)$ (see Equation 2.80) maximizing the likelihood of the adaptation data $\mathbf{O}$ of length T frames,

$$\tilde{\Lambda} = \arg\max_{\Lambda} P(\mathbf{O}|\lambda, \Lambda) \tag{2.81}$$

The parameters of the linear transformation [14] are found using an Estimation Maximization (EM) approach. In the estimation step of this algorithm, an auxiliary function $\mathcal{Q}(\hat{\lambda}, \lambda)$ is used where in this case λ and $\hat{\lambda}$ are the original and adapted model set, respectively.

$$\mathcal{Q}(\hat{\lambda}, \lambda) = \sum_{\text{all } \mathbf{Q}} P(\mathbf{Q}|\mathbf{O}, \lambda) \cdot \log P(\mathbf{O}, \mathbf{Q}|\hat{\lambda}) \tag{2.82}$$

As we have seen, the transform matrices are tied across a number of Gaussians. For a particular transform $\mathbf{W}_r$, the M_r Gaussian components will be tied together as determined by the class tree across a number of Gaussian distributions. In addition, considering that the set of original models is denoted as $\mathcal{M}$ and the set of transformed models sharing transformations is denoted as $\hat{\mathcal{M}}$, the auxiliary function $\mathcal{Q}$ can be rewritten as,

$$\mathcal{Q}(\hat{\mathcal{M}}, \mathcal{M}) = -\frac{1}{2} \sum_{r=1}^{R} \sum_{m_r=1}^{M_r} \sum_{t=1}^{T} \gamma_{m_r}(t) \left[K_{m_r} + \log(|\hat{\mathbf{U}}_{m_r}|) + (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_{m_r})^{\mathrm{T}} \hat{\mathbf{U}}_{m_r}^{-1} (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_{m_r}) \right] \tag{2.83}$$

where K_{m_r} subsumes all constants, $\hat{\boldsymbol{\mu}}_{m_r}$ and $\hat{\mathbf{U}}_{m_r}$ are the transformed mean and variance for

[14] The transformation of the state output distribution and duration is analogous, so in the next description, only the state output transformation is considered.

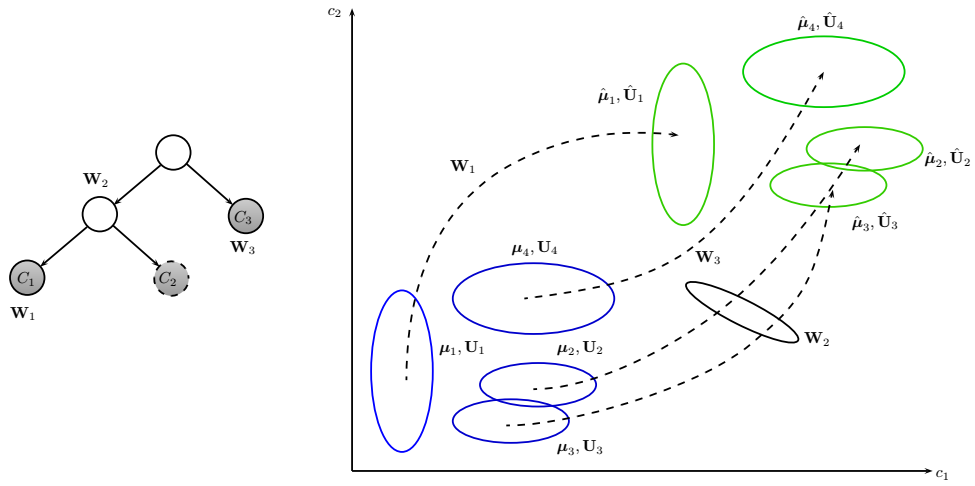


Figure 2.12: Example of a two-dimension HMM-based CMLLR adaptation and its regression class tree. Note that in this example, the regression class tree has three classes, C_1 and C_3 with sufficient data to have its own transforms $\mathbf{W}_1$ and $\mathbf{W}_3$ whereas C_2 uses the transformation of the parent node ($\mathbf{W}_2$).

component m_r, respectively and $\gamma_{m_r}(t)$ is the occupation likelihood defined as

$$\gamma_{m_r}(t) = P(q_{m_t}(t)|\mathcal{M}, \mathbf{O}_T)$$

where $q_{m_r}(t)$ indicates the Gaussian component m_r at time t and $\mathbf{O}_T = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ is the adaptation data.

The objective is to maximize this equation with respect to the transformation $\mathbf{W}_r$. So, substituting Equations 2.76 and 2.77 into Equation 2.83 and using property in Equation 2.80 and re-arranging,

$$\begin{aligned}\mathcal{Q}(\hat{\mathcal{M}}, \mathcal{M}) \quad = \quad & K - \frac{1}{2}\sum_{r=1}^{R}\sum_{m_r=1}^{M_r}\sum_{t=1}^{T}\gamma_{m_r}(t)[K_{m_r} + \log(|\hat{\mathbf{U}}_{m_r}|) - \\ & \log(|\boldsymbol{\zeta}_r|^2) + (\mathbf{W}_r\boldsymbol{\xi}_t - \hat{\boldsymbol{\mu}}_{m_r})^{\mathrm{T}}\hat{\mathbf{U}}_{m_r}^{-1}(\mathbf{W}_r\boldsymbol{\xi}_t - \hat{\boldsymbol{\mu}}_{m_r})]\end{aligned} \tag{2.84}$$

By ignoring all terms independent of matrix $\mathbf{W}_r$, considering that only diagonal covariances matrices are used and re-arranging,

$$\mathcal{Q}(\hat{\mathcal{M}}, \mathcal{M}) = K + \sum_{r=1}^{R}\left[\beta\log(\mathbf{p}_{ri}w_{ri}^{\mathrm{T}}) - \frac{1}{2}\sum_{j=1}^{L+1}\left(\mathbf{w}_{rj}\mathbf{G}_r^{(j)}\mathbf{w}_{rj}^{\mathrm{T}} - 2\mathbf{w}_{rj}\mathbf{k}_r^{(j)}\right)\right] \tag{2.85}$$

where β is an auxiliary variable defined in Equation 2.88.

Taking Equation 2.85 is possible to obtain the following maximization equation for $\mathcal{Q}$ as described by (Gales, 1997):

$$\frac{\partial \mathcal{Q}(\hat{\mathcal{M}}, \mathcal{M})}{\partial \mathbf{w}_{ri}} = \beta \frac{\mathbf{p}_{ri}}{\mathbf{p}_{ri}\mathbf{w}_{ri}^{\mathrm{T}}} - \mathbf{w}_{ri}\mathbf{G}_r^{(i)} + \mathbf{k}_r^{(i)} = 0 \tag{2.86}$$

where $\mathbf{w}_{ri}$ is the i-th row vector of matrix $\mathbf{W}_r$, $\mathbf{p}_{ri}$ is the extended cofactor row vector $[0, p_{ri_1}, \ldots, p_{ri_L}]$ of $\boldsymbol{\zeta}$ (being $p_{ri_j} = \mathrm{cof}\left(\boldsymbol{\zeta}_{ij}\right)$), $\mathbf{G}_{ri}$ is a $(L+1)\times(L+1)$ matrix which takes into account the covariance of the model and $\mathbf{k}_{ri}$ is a $L+1$ vector:

$$\mathbf{G}_r^{(i)} = \sum_{m_r=1}^{M_r} \frac{1}{\sigma_{m_r i}^2} \sum_{t=1}^{T} \gamma_{m_r}(t)\boldsymbol{\xi}_t\boldsymbol{\xi}_t^{\mathrm{T}}$$

$$\mathbf{k}_r^{(i)} = \sum_{m_r=1}^{M_r} \frac{\mu_{m_r i}}{\sigma_{m_r i}^2} \sum_{t=1}^{T} \gamma_{m_r}(t)\boldsymbol{\xi}_t^{\mathrm{T}}$$

Solving maximization in Equation 2.86 for row $\mathbf{w}_{ri}$ yields to,

$$\mathbf{w}_{ri} = (\alpha\mathbf{p}_{ri} + \mathbf{k}_r^{(i)})\mathbf{G}_r^{(i)-1} \tag{2.87}$$

where α is also an auxiliary variable satisfying:

$$\beta = \alpha^2\mathbf{p}_{ri}\mathbf{G}_r^{(i)-1}\left(\mathbf{p}_{ri}^{\mathrm{T}} + \mathbf{k}_r^{(i)\mathrm{T}}\right) = \sum_{m_r=1}^{M_r}\sum_{t=1}^{T}\gamma_{m_r}(t) \tag{2.88}$$

This is an iterative process which performs a row by row optimization. Each row is dependent of the other rows via its cofactor. The process is described in Algorithm 2.

2.7.2 Maximum A Posteriori (MAP)

This adaptation process is referred to as Bayesian adaptation (Lee, 1991). It is based on the fact that if we have knowledge about the parameters to be estimated, we can incorporate such prior knowledge into the prior distribution. Such a prior is often called an informative prior.

For MAP adaptation purposes, the informative priors that are generally used are the speaker independent model parameters. As MAP adaptation works at the distribution's parameter level using prior knowledge of the target's parameters, the update formula for the adapted mean in state j and mixture component m is,

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{N_{jm}\boldsymbol{\sigma}_{jm}'^2}{N_{jm}\boldsymbol{\sigma}_{jm}'^2 + \boldsymbol{\sigma}_{jm}''^2}\mu_{jm}'' + \frac{\boldsymbol{\sigma}_{jm}''^2}{\boldsymbol{\sigma}_{jm}''^2 + N_{jm}\boldsymbol{\sigma}_{jm}'^2}\boldsymbol{\mu}_{jm}' \tag{2.90}$$

where the speaker independent parameters are referred as μ' and σ'^2 and the estimation using only the adaptation data as μ'' and σ''^2. N_{jm} is the occupation likelihood of the adaptation data defined

Algorithm 2 Iterative process to compute a CMLLR transformation.

- Initialize transformation $\mathbf{W}_r^{(0)} = \text{diag}\,[a_1, \ldots, a_L]$ as a diagonal matrix where a_i is calculated using a reduced version of this algorithm (Gales, 1997).
- For each iteration it:

  1. For each row i

     (a) Get the best α using Equation 2.89. There are two possible solutions for α in Equation 2.88. The solution that yields the maximum increase in the auxiliary likelihood function is used. Thus, ignoring all terms independent of α in Equation 2.85 yields

     $$\mathcal{Q}(\hat{\mathcal{M}}, \mathcal{M}) = \beta \log(|\alpha a + b|) - \frac{1}{2}\alpha^2 a$$

     where

     $$\begin{aligned} a &= \mathbf{p}_{ri}\mathbf{G}_r^{(i)-1}\mathbf{p}_{ri}^{\mathrm{T}} \\ b &= \mathbf{p}_{ri}\mathbf{G}_r^{(i)-1}\mathbf{k}_r^{(i)\mathrm{T}} \end{aligned}$$

     and using the two maximum values of α

     $$\mathcal{Q}^{(it)}(\hat{\mathcal{M}}, \mathcal{M}) = \beta \log\left(\left|\frac{b \pm \sqrt{b^2 + 4a\beta}}{2}\right|\right) - \frac{a}{2}\left(\frac{-b \pm \sqrt{b^2 + 4a\beta}}{2a}\right)^2 \tag{2.89}$$

     (b) Compute cofactor of row i ($\mathbf{p}_{ri} = \text{cof}\left(\boldsymbol{\zeta}_{ij}\right)$).

     (c) Calculate row $\mathbf{w}_{ri}$ using Equation 2.87.

     (d) Compute likelihood for current iteration $\mathcal{Q}^{(it)}(\hat{\mathcal{M}}, \mathcal{M})$ using Equation 2.85.

     (e) Calculate initial likelihood using current cofactor ($\mathcal{Q}^{(0)}(\hat{\mathcal{M}}, \mathcal{M})$) and initial row i of matrix $\mathbf{W}_r^{(0)}$.

     (f) If $\mathcal{Q}^{(it)}(\hat{\mathcal{M}}, \mathcal{M}) > \mathcal{Q}^{(0)}(\hat{\mathcal{M}}, \mathcal{M})$, take current row estimate ($\mathbf{w}_{ri}$) for matrix $\mathbf{W}_r^{(it)}$.

as

$$N_{jm} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{jm}^{r}(t)$$

However, to ease mathematical tractability conjugate priors are used, which results in a simpler adaptation formula. In Bayesian theory (Raiffa, 1961) a conjugate prior for the mean of the Gaussian density is known to be a Gaussian density and assuming that the precision ($\tau = {}^{1}/_{\sigma''^2}$ is known and fixed), Equation 2.90 yields

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\tau \boldsymbol{\mu}_{jm} + N_{jm} \bar{\boldsymbol{\mu}}_{jm}}{\tau + N_{jm}} \tag{2.91}$$

where here $\bar{\boldsymbol{\mu}}_{jm}$ is the mean of the observed adaptation data defined as the extension of Equation 2.11 with T_r observations,

$$\bar{\boldsymbol{\mu}}_{jm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{jm}^{r}(t) \mathbf{o}_t^r}{\sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_{jm}^{r}(t)}$$

being $\boldsymbol{\mu}_{jm}$ the original mean, R the number of training examples, $\mathbf{o}_t^r$ the observation at time t for example r and $\gamma_{jm}^{r}(t)$ is the occupation likelihood using observations from example r calculated using Equation 2.13. Although τ in Equation 2.91 could be estimated from the adaptation data it is a constant devoted to control the a priori knowledge of the adaptation process. Note that when $\tau \to \infty$ the MAP estimate $\hat{\boldsymbol{\mu}}_{jm}$ is simply the prior mean.

MAP adaptation requires more adaptation data to be effective when compared to linear regression adaptations, although when larger amounts of adaptation training data become available, MAP begins to perform better than the linear regression. This is because when a large number of adaptation data is available $N_{jm} \to \infty$, thus the MAP estimate converges to the Maximum Likelihood estimate of the adaptation data asymptotically.

It order to make the whole adaptation process independent of the amount of adaptation data, MAP estimation is applied after CMLLR (Yamagishi and Kobayashi, 2007). By applying the MAP estimation to the model transformed by the linear regression, it is possible to improve the estimation for the distribution having sufficient amount of adaptation samples. Thus, using the linearly transformed means as the priors for MAP adaptation, components that have a low occupation likelihood in the adaptation data, (and hence would not change much using MAP alone) will be adapted using a regression class transform.

2.7.3 Speaker-independent HMM training

Speaker independent (SI) models (i.e, an average voice model) is trained using a set of different speakers in order to obtain a robust adapted voice with only some samples of the target speaker (see experiments in Section 5.4.2 for details of number of speakers and corpora size). There are two possibilities to build the average voice, the conventional and the adaptive training.

There are two techniques to perform SI training. On the one hand, a conventional SI-based

training (Yamagishi and Kobayashi, 2007) uses the same steps as the training for a speaker dependent system but using data from all speakers simultaneously. Afterwards, target data is used to adapt the HMMs using mean and covariance adaptation techniques described in Sections 2.7.1 and 2.7.2. On the other hand, a Speaker Adaptive Training (SAT) (Yamagishi et al., 2009) could be used in order to reduce the negative influence of speaker differences. SI model is trained so that the resultant model, obtained by the CMLLR-based speaker adaptation, maximizes the total likelihood for the respective training speakers. In the SAT paradigm, the regression matrix $\mathbf{W}$ is re-estimated in accordance with a standard EM algorithm and the mean vectors and the covariance matrices are re-estimated using the updated values of the regression matrices based on an extended EM algorithm. This re-estimation process is repeated until the convergence.

SI training used in the Experiments Section 5 uses the conventional SD training with multiple speakers. This technique favours a lower computational cost with respect to the SAT approach. Adaptation is performed concatenating CMLLR (see Section 2.7.1) and MAP (Section 2.7.2). In this case, mean and covariance matrices are obtained by simultaneously transforming all parameters.

2.8 Unit clustering and selection

Any Text-To-Speech system uses some sort of unit selection algorithm in order to choose a sequence of units to construct synthetic speech. Unit selection algorithm used by Co-TTS systems (see Section 1.2) use an optimization procedure based on minimizing a sum of weighted cost functions computed to find the optimal sequence of units to be concatenated from a speech corpus. In HMM-based TTS systems, each HMM represents a synthesis unit (e.g., a phoneme). As the realization of an isolated phoneme is heavily dependent on its surrounds, each synthesis unit is represented by a linguistic set of contextual features. Thus, instead of having a single model for phoneme, there are a large number of specialized models, each of which more precisely describes that phone within a context. The most common way of doing this is by using the so-called triphone model (extensively used in speech recognition (Schwartz et al., 1985)). In synthesis, the context has a composite form that uses a set of features to represent the units with characteristics related to phonemes (e.g., vocal or consonant), words (e.g., position in sentence) and utterance (e.g., number of words). See Figure 2.13 for an example of a full context unit.

In order to accurately capture the variations in real speech spectra, it is necessary to have a large number of robustly estimated HMMs. Hence, a minimum number of data samples must be used for each of these models. Unfortunately, there is rarely enough data to have a sufficient number of examples of every unit, specially if the context is composite. Hence, a method is needed to balance model complexity against data availability.

A common solution in speech recognition is to cluster and tie HMM states by means of a decision tree (Young et al., 1994), so parameters from well-trained models are shared for use in those that suffer from data sparsity. The main point is that while the features are used to measure similarity

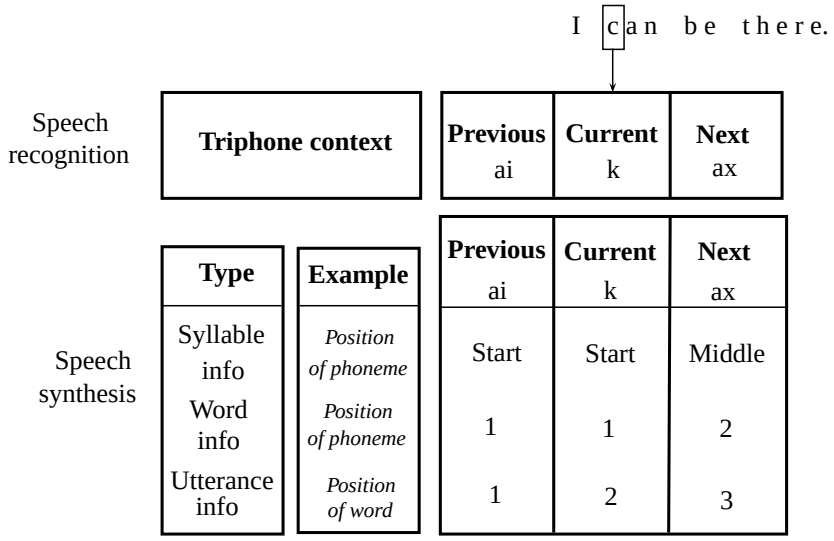


Figure 2.13: Example of a composite linguistic context for a phoneme in the sentence "I can be there". Unlike speech recognition, speech synthesis uses a larger set of linguistic features. Note that the information is often referred to syllable, word and utterance. In this case, 3 examples are presented: position of the phoneme in the syllable, position of the phoneme in the word and position of the word in the utterance. Current phoneme "k" is the start of the syllable, its position in the word is 1 out of 3 phonemes and the word containing this phoneme ("can") is the second in the utterance out of 3 in total. The rest of the positions can be described in a similar way.

between units, the actual data determines how close any particular feature combination actually is (Yoshimura et al., 1999).

2.8.1 Decision tree-based clustering for synthesis

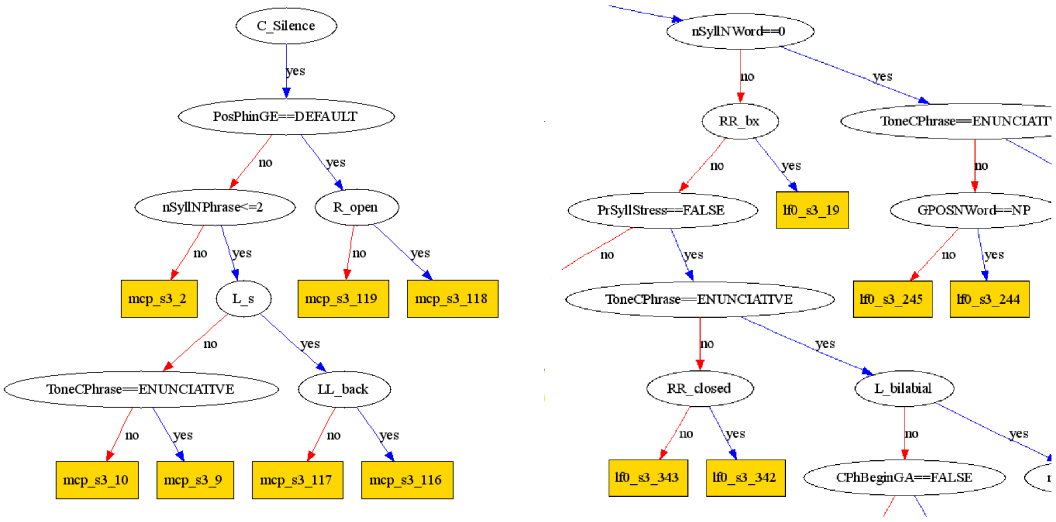
Decision trees for HMM-based speech synthesis are used for state clustering and state selection. On the one hand, a decision tree is constructed to cluster the model parameters of each state by tying those which are acoustically close. This way, sufficient data is guaranteed for all models so HMM training becomes more robust. On the other hand, constructed decision trees can be used to predict leaf nodes from context definitions even for models which are not in the training set.

Unlike recognition, synthesis HMMs will have not only information about vocal tract (e.g., mel-cepstral coefficients), but also about F0 and state durations (and any other information such as mixed excitation parameters). Each of these components are treated independently (i.e., trees are constructed separately) because they are affected by different contextual factors. For this reason, there are KS trees where K is the number of states in an HMM and S is the number of independent information in the HMM (i.e., vocal-tract or F0) [15].

[15] Obviously, S reflects the number of streams (detailed in Section 3.1) which usually counts vocal-tract, F0, mixed excitation and durations.

The decision tree for each state is a binary representation in which a yes/no question is attached to each node. By following the correct branches of the tree for a given context, it is possible to reach a leaf node. Each node represents a set of distributions which share the same parameters. Some examples of real questions are

- Is the current phoneme the start of the syllable?
- Is the next phoneme the end of the word?
- Is the number of syllables from the previous stressed syllables to the current syllable less or equal to 1?
- Is the position of the current word in current phrase below or equal 29?



(a) Vocal tract (b) F0

Figure 2.14: A partial example of two decision trees for vocal-tract and F0. In these trees, the names of the nodes are codified as explained in Table 2.4.

A real decision tree is depicted in Figure 2.14. In this example, different questions are used to cluster vocal-tract and F0. It is important to highlight that the spectrum tree (Figure 2.14(a)) has more information related to phonemes and their features (e.g. current vowel?, current vowel is "I"?, current consonant is unvoiced?) whereas the F0 tree (Figure 2.14(b)) is more likely to take into account information related to syllables, words and sentences (e.g. left syllable is accented?, current sentence has 12 words?). This is due to the different dependencies between linguistic attributes and acoustic parameters. Vocal-tract is highly dependent on phonemes identity whereas F0 variations are correlated with the utterance prosody contour (e.g., type of sentence, number of words or position of syllables).

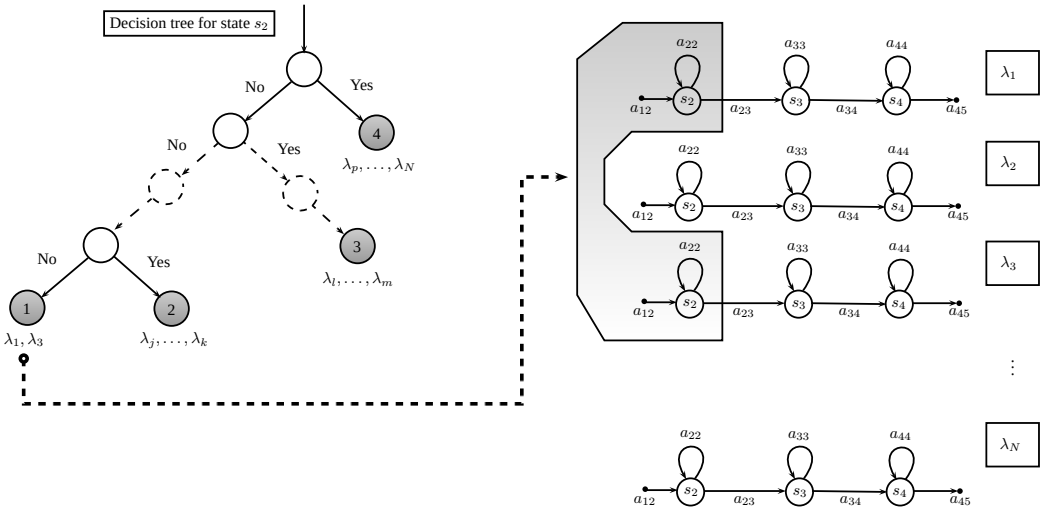


Figure 2.15: An Example of a decision tree clustering the first emitting state of a set of HMMs $\lambda_i, i \in [1, \ldots, N]$. In the picture, models λ_1 and λ_3 are clustered in the first leaf node while the rest of the nodes are clustering different subsets of model states. Eventually, any state s_2 belonging to any model λ_i is clustered in a leaf node.

There are some Co-TTS systems which use similar unit clustering approaches. They were conceived in order to avoid the necessity of defining a heuristic distance between contexts to measure the target cost and to reduce the search space and the selection process complexity (Tokuda et al., 2002b). A good example is a system based on clustering similar units (Black and Taylor, 1997). In this technique, the optimal path is obtained by performing the search for each group first and then refining the search within each of these clusters. Similarly, an HMM-based decision tree approach was used for the trainable IBM system by (Donovan and Woodland, 1999). The main difference of these approaches with the decision tree-based clustering technique used in the HMM-based TTS framework is that the former methods cluster contexts in advance and select each unit from a cluster whereas in the latter, each cluster is represented by statistics (e.g., for a tree, a leaf node is a state and its parameters represent any state within this group).

2.8.2 Decision tree construction

The aim to build a mixture tied HMM system is to ensure there is sufficient training data to robustly estimate each set of state output distribution parameters while retaining the important context-dependent acoustic distinctions within each class.

A decision tree is constructed in a top-down manner. First all nodes are a unique root node. Then, this node is iteratively split into a set of nodes selected to minimize a likelihood. Depending on the minimization criterion a different stop rule is used. For decision trees, this criterion is usually based on a Maximum Likelihood (ML) (Young et al., 1994) or a Minimum Description Length

(MDL) (Shinoda and Watanabe, 1997, 2000) approach.

Figure 2.16 shows an example of the iteration process during clustering. Let S_0 be a root node of the decision tree and U a model [16] defined for the leaf node set $U = \{S_1, S_2, \ldots, S_M\}$. Each node is assigned a Gaussian distribution which is the result of combining several other Gaussian distributions.

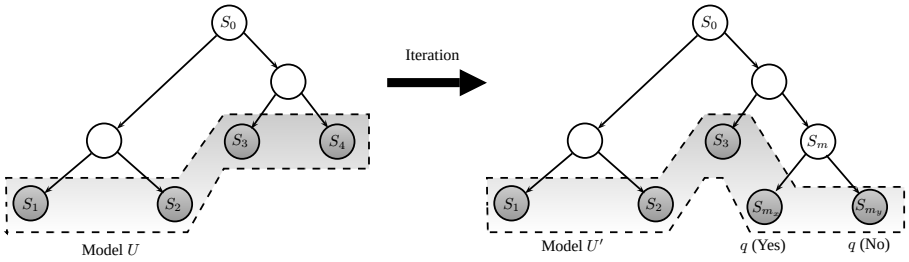


Figure 2.16: Example of the decision tree construction based on splitting nodes. A model U split from root node S_0 with $M = 4$ and a new model U' is constructed by splitting $S_{m=4}$ from model U using question q.

As you can see, a decision tree construction is a node splitting process. Before describing the clustering steps in detail, the following assumptions are made in order to reduce the computational cost (Shinoda and Watanabe, 2000):

1. The transition probabilities of HMMs can be ignored in the calculation of the likelihood function for a node set. Hence, the state splitting process does not change the frame or state alignment between the data and the model.

2. The function of the log-likelihood for each state can be given by the sum of the log-likelihood for each data frame weighted by the state occupancy probability for each state.

From the second assumption and considering that the training data is a sequence of T feature vectors $\mathbf{o} = [\mathbf{o}_1, \ldots, \mathbf{o}_T]^{\mathrm{T}}$, the likelihood $\mathcal{L}$ for node S_m is approximated by

$$\mathcal{L}(S_m) \approx \sum_{t=1}^{T} \gamma_m(t) \cdot \log\left(P(\mathbf{o}_t|\boldsymbol{\mu}_{S_m}, \boldsymbol{\Sigma}_{S_m})\right) \tag{2.92}$$

where $\gamma_m(t)$ is the state occupancy probability for node S_m calculated during the Forward-Backward algorithm (see Equation 2.14). Assuming that the output probability distribution $P(\mathbf{o}_t|\boldsymbol{\mu}_{S_m}, \boldsymbol{\Sigma}_{S_m})$ is a multidimensional Gaussian distribution $\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{S_m}, \boldsymbol{\Sigma}_{S_m})$ with mean $\boldsymbol{\mu}_{S_m}$ and covariance matrix $\boldsymbol{\Sigma}_{S_m}$, the likelihood yields

$$\mathcal{L}(S_m) \approx -\frac{1}{2} \sum_{t=1}^{T} \gamma_m(t) \cdot \left(L \log(2\pi) + \log(|\boldsymbol{\Sigma}_{S_m}|) + (\mathbf{o}_t - \boldsymbol{\mu}_{S_m})^{\mathrm{T}} \boldsymbol{\Sigma}_{S_m}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{S_m})\right) \tag{2.93}$$

[16] Note that a model here is not an HMM, but a set of leaf nodes in the decision tree.

where L is the dimension of the observation feature vector. Using the parameter re-estimation Equation 2.12 and considering the covariance matrix is assumed to be diagonal, it is possible to use the following property

$$\sum_{t=1}^{T} \gamma_m(t)(\mathbf{o}_t - \boldsymbol{\mu}_{S_m})^{\mathrm{T}} \boldsymbol{\Sigma}_{S_m}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{S_m}) = L \sum_{t=1}^{T} \gamma_m(t)$$

to simplify the total likelihood as,

$$\mathcal{L}(S_m) \approx -\frac{1}{2} \sum_{t=1}^{T} \gamma_m(t) \cdot (L \log(2\pi) + \log(|\boldsymbol{\Sigma}_{S_m}|) + L)) \tag{2.94}$$

The log-likelihood of the data for all the nodes in set U is

$$\mathcal{L}(U) \approx \sum_{m=1}^{M} \mathcal{L}(S_m) \tag{2.95}$$

In order to find the optimal model U (that is, the optimal set of leaf nodes), we would need to calculate the auxiliary function $\mathcal{L}$ for all possible models. Since this is computationally very expensive, a suboptimal solution based on differences of $\mathcal{L}(U)$ is used depending on the ML or MDL criterion.

The value of Σ_{S_m} is not calculated directly from the data but from the statistics from each unique context. A context is defined as the set of states clustered within a leaf node ($C(S_m)$). This reduces both the storage and computational requirements since the number of different contexts is (often substantially) smaller than the number of examples (Odell, 1995). In this case, the covariance matrix is computed as

$$\boldsymbol{\Sigma}_{S_m} = E[\mathbf{o}^2] - E[\mathbf{o}]^2 = \frac{\sum\limits_{c \in C(S_m)} \gamma_c \left(\boldsymbol{\Sigma}_c + \boldsymbol{\mu}_c \boldsymbol{\mu}_c^{\mathrm{T}} \right)}{\sum\limits_{c \in C(S_m)} \gamma_c} - \left(\frac{\sum\limits_{c \in C(S_m)} \gamma_c \boldsymbol{\mu}_c}{\sum\limits_{c \in C(S_m)} \gamma_c} \right) \left(\frac{\sum\limits_{c \in C(S_m)} \gamma_c \boldsymbol{\mu}_c}{\sum\limits_{c \in C(S_m)} \gamma_c} \right)^{\mathrm{T}} \tag{2.96}$$

where $C(S_m)$ is the set of contexts that are to be represented by the distribution of node S_m and $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the parameters for context c calculated using the conventional formulae of Section 2.2.2 (see Training) respectively.

2.8.2.1 Maximum Likelihood (ML) criterion

Suppose that node S_m in model U is split into two nodes S_{m_x} and S_{m_y} by using question q (see example of Figure 2.16) and let U' be the model obtained after splitting node S_m. The auxiliary

function for model U' is,

$$\begin{aligned}\mathcal{L}(U') &\approx \sum_{\substack{i=1\\ i\neq m}}^{M} \mathcal{L}(S_i) + \mathcal{L}(S_{m_x}) + \mathcal{L}(S_{m_y}) \\ &= \sum_{\substack{i=1\\ i\neq m}}^{M} \mathcal{L}(S_i) - \frac{1}{2}\sum_{t=1}^{T} \gamma_{m_x}(t)\cdot\big(L\log(2\pi) + \log(|\boldsymbol{\Sigma}_{S_{m_x}}|) + L\big) \\ &\quad -\frac{1}{2}\sum_{t=1}^{T} \gamma_{m_y}(t)\cdot\Big(L\log(2\pi) + \log(|\boldsymbol{\Sigma}_{S_{m_y}}|) + L\Big)\end{aligned} \tag{2.97}$$

Let $\delta_m(q)$ be the increase of the log-likelihood when node S_m is split into two in response to using question q (Shinoda and Watanabe, 1997),

$$\begin{aligned}\delta_m(q) &= \mathcal{L}(U') - \mathcal{L}(U) \\ &= -\frac{1}{2}\Big(\Theta_{m_x}\cdot\log(|\boldsymbol{\Sigma}_{S_{m_x}}|) + \Theta_{m_y}\cdot\log(|\boldsymbol{\Sigma}_{S_{m_y}}|) - \Theta_m\cdot\log(|\boldsymbol{\Sigma}_{S_m}|)\Big)\end{aligned} \tag{2.98}$$

where $\Theta_j = \sum_{t=1}^{T}\gamma_j(t)$. By using the difference $\delta_m(q)$ is possible to automatically construct a decision tree using Algorithm 3. Since the increase of likelihood is always positive, it is also essential to use an external parameter to control the number of clusters. Some methods apply a threshold value to the total occupancy Θ_j and/or to the log-likelihood increase $\delta_m(q)$. The optimization of this parameters requires a series of computationally expensive experiments and additional data (Young et al., 1994).

2.8.2.2 Minimum Description Length (MDL) criterion

Before relating the MDL criterion to the clustering process, let's briefly introduce this technique (Rissanen, 1984). Suppose we are given a sequence of N data points $\mathbf{x} = \{x_1, \ldots, x_N\}$. As an estimation problem, we could say that we are looking for the model that has generated this data. In other words, we try to estimate a vector of parameters $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_L]$ of a statistical model $P_{\boldsymbol{\theta}}(\mathbf{x})$ for the data $\mathbf{x}$. The MDL criterion is an effective way to select the optimal probabilistic model from among various models. In order to do that, it selects the statistical model with the minimum description length for the given data. The description length $\mathcal{D}_j(\mathbf{x})$ for data $\mathbf{x}$ of an underlying probabilistic model j is given by,

$$\mathcal{D}_j(\mathbf{x}) = -\log\Big(P_{\hat{\boldsymbol{\theta}}^{(j)}}(\mathbf{x})\Big) + \frac{L_j}{2}\log(N) + \log(I) \tag{2.99}$$

where:

- $\hat{\boldsymbol{\theta}}^{(j)}$ represents the ML estimate of model j for the vector of parameters $\boldsymbol{\theta}$.
- L_j is the number of parameters of $\hat{\boldsymbol{\theta}}^{(j)}$ in probabilistic model j.

- $\log(I)$ is a normalizing constant, being I the total number of probabilistic models being tested, that is, $j \in [1, I]$.

Figure 2.17 shows an example of the MDL criterion convergence. One of the advantages of the MDL criterion is that the second term defined in Equation 2.99 works as a penalty imposed for employing a large model size. So, as a model becomes more complex, the value of the first term decreases and that of the second term increases.

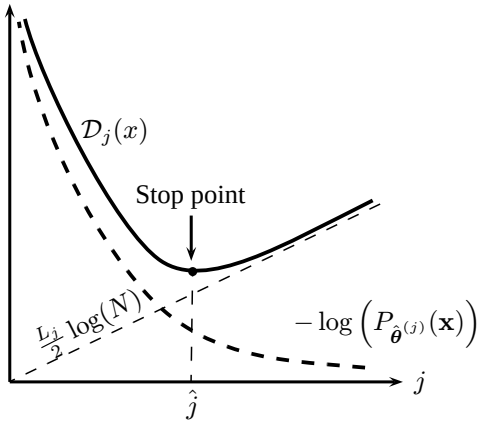


Figure 2.17: An example of the MDL criterion convergence where the horizontal axis is the set of models being created (i.e., the splitting process) and $\hat{j}$ is the optimal probabilistic model. The solid line is the description length criterion defined in Equation 2.99 whereas the dotted lines are the rest of the terms in that equation. Since the first term is identical to the one defined in the ML approach, the main advantage of the MDL criterion is that the stop point can be optimally selected using the second term of Equation 2.99.

In order to apply the MDL criterion to the decision tree-based clustering problems, and taking original Equation 2.99 as a reference, the following terms are redefined as follows (Shinoda and Watanabe, 1997, 2000).:

- The first term is identical to the negative of the log-likelihood described in Equation 2.95.

- L_i is now the number of parameters in model U_i. Assuming that covariances are diagonal, the number of parameters to be estimated for model U_i is $L_i = 2LM_i$ (being L the dimensionality of the mixture components and M_i the number of nodes in model U_i. Factor 2 is applied since dimensionality L_i is for means and variances. Finally, N, the number of data samples, is now defined as the sum of the state occupancy for all nodes defined as $\Upsilon = \sum_{m=1}^{M_i} \Theta_m$.

- Last terms is considered to be a constant for all models.

Taking into account Equation 2.99 and imposing previous assumptions, the description length of model U applied to the node splitting process is:

$$\begin{aligned}\mathcal{D}(U) &= -\mathcal{L}(U) + LM\log\Upsilon + C \\ &= \frac{1}{2}\sum_{m=1}^{M}\Theta_m\left(L + L\log(2\pi) + \log(|\mathbf{\Sigma}_{S_m}|)\right) + LM\log\Upsilon + C \end{aligned} \tag{2.100}$$

where C is the third term in Equation 2.99 considered as constant. In a comparison among models, the model with the shortest description length $\mathcal{D}$ may be considered the one having the best balance between likelihood and complexity.

Considering again that node S_m is split into two nodes, the description length for the resulting model U' of $M+1$ nodes is,

$$\begin{aligned}\mathcal{D}(U') &= \frac{1}{2}\Theta_{m_x}\left(L + L\log(2\pi) + \log(|\mathbf{\Sigma}_{S_{m_x}}|)\right) \\ &\quad + \frac{1}{2}\Theta_{m_y}\left(L + L\log(2\pi) + \log(|\mathbf{\Sigma}_{S_{m_y}}|)\right) \\ &\quad + \frac{1}{2}\sum_{\substack{i=1 \\ i\neq m}}^{M}\Theta_i\left(L + L\log(2\pi) + \log(|\mathbf{\Sigma}_{S_i}|)\right) \\ &\quad + L(M+1)\log\Upsilon + C \end{aligned} \tag{2.101}$$

Hence, the difference between description lengths before and after the splitting process becomes,

$$\begin{aligned}\delta_m(q) &= \mathcal{D}(U') - \mathcal{D}(U) = \\ &= \frac{1}{2}\left(\Theta_{m_x}\log(|\mathbf{\Sigma}_{S_{m_x}}|) + \Theta_{m_y}\log(|\mathbf{\Sigma}_{S_{m_y}}|) - \Theta_m\log(|\mathbf{\Sigma}_{S_m}|)\right) + L\log\Upsilon \end{aligned} \tag{2.102}$$

By using this difference $\delta_m(q)$ it is possible to construct a decision tree using a similar process to the one described for the ML approach in the previous Section. In order to reduce the computational cost during the tree construction (see Algorithm 3), the second term of Equation 2.99 is fixed to the initial case in which all nodes are tied together. Hence, the threshold used for the MDL criterion is defined as

$$R_{MDL} = LJ\log\sum_{j=1}^{J}\Theta_{m_j} \tag{2.103}$$

where J is the total number of mixtures in the system.

2.8.2.3 Decision tree construction algorithm

As stated in previous sections, both ML and MDL criterions are mean to build a tree using the same top-down optimisation approach. The building process is described in Algorithm 3 where the main difference relies on the stopping criterion. Thus, the main advantage of the MDL criterion is that

it does not need any external information to control the degree of clustering since this is performed using a penalty term related to the length of the model.

The process to build a tree is as follows. Initially all mixtures are placed in a single cluster at the root of the tree. Each leaf node is associated with a relative improvement (i.e., a likelihood difference defined in Equation 2.98 for ML or a description length difference defined in 2.102 for MDL).

Firstly, the best node to split is selected to have the maximum $\delta_m(q)$ according with the previous best question [17]. This value must be over the threshold defined in Equation 2.103 for MDL or externally computed for ML. Once the node to split is selected, a question is chosen to locally maximize the likelihood of the training data over all possible new models U' using Equation 2.97 for ML or Equation 2.101 for MDL. For every split, nodes with an increase in log likelihood below the threshold are discarded for further splitting. When there are no more nodes to be split, the process stops.

Eventually, all states in the same leaf node are then tied so they share the same HMM parameters (i.e., means and variances). The merging of those parameters is performed using the statistics of the leaf node contexts. The variance is calculated as in Equation 2.96 whereas the mean is obtained as

$$\boldsymbol{\mu}_{S_m} = \frac{\sum_{c \in C(S_m)} \gamma_c \boldsymbol{\mu}_c}{\sum_{c \in C(S_m)} \gamma_c} \tag{2.104}$$

2.8.3 Decision trees in Spanish

In order to develop the first objective of this thesis, this section proposes the Spanish HMM-based TTS system. The idea here is to adapt the decision tree-based clustering questions to represent the Spanish language. In consequence, a new labelling process is needed since the composite context for each phoneme must be representative of the new language (Gonzalvo et al., 2007a,b).

A good selection of a question set yields not just to a good decision tree building performance but also to a better synthesis quality, specially with respect to expressiveness. For these reasons, contextual factors have to accurately describe the linguistic context without introducing too much redundancy. As we have just described, the final set of questions used in the tree are selected according to the optimal stopping point according to the MDL criterion. Here, a complete list of attributes is described and experiment in Section 5.3.2 analyzes the use of these questions in the tree. The number of nodes of the resulting tree is shown in Table 2.2 for vocal-tract and in Table 2.3 for F0. These tables are extracted using the Spanish voice corpus described in Section 5.1 and a

[17] Note that for simplification, Algorithm 3 does not take into account the existence of a question for the first iteration.

Algorithm 3 Algorithm to create a decision tree using the ML or the MDL criterion. Q is the total number of questions.

if criterion is MDL **then**
  Compute threshold (R_{MDL}) using Equation 2.103.
else
  Get an optimal threshold (R_{ML}).
end if
Define initial model $U = \{S_0\}$
repeat
  if criterion is MDL **then**
    // Find best node to split ($S_{m'}$) in model U using Equation 2.102
    $S_{m'} = \underset{q \in Q, m \in [1,M]}{\arg\min} \; (\delta_m(q))$
    // Select best question q'
    $q' = \underset{q \in [1,Q]}{\arg\max}(\mathcal{D}(U'))$ using Equation 2.101
    if $\delta_{m'}(q') < R_{MDL}$ **then**
      Stop splitting this node.
    end if
  else
    // Find best node to split ($S_{m'}$) in model U using Equation 2.98
    $S_{m'} = \underset{q \in Q, m \in [1,M]}{\arg\min} \; (\delta_m(q))$
    // Select best question q'
    $q' = \underset{q \in [1,Q]}{\arg\max}(\mathcal{L}(U'))$ using Equation 2.97
    if $\delta_{m'}(q') < R_{ML}$ **then**
      Stop splitting this node.
    end if
  end if
until There are no more nodes to split
Tie states in each node

Female English voice (Tokuda et al., 2002b) [18]. Note that the number of clusters is significantly higher for F0 than for vocal-tract and this also happens for English. Clearly, the range of variations of F0 in a frame-by-frame basis is higher than for the vocal-tract and therefore, MDL criterion needs to use more questions in order to fulfill the stopping criterion.

HMM state	Spanish	English
1	117	283
2	125	232
3	105	239
4	125	256
5	130	278

Table 2.2: Number of nodes for each HMM state of the Spanish and English trees clustering vocal-tract parameters.

HMM state	Spanish	English
1	485	570
2	640	601
3	506	871
4	354	783
5	441	556

Table 2.3: Number of nodes for each HMM state of the Spanish and English trees clustering the F0 parameter.

The set of possible attributes used to build the questions is shown in Table 2.4. This set is designed focusing on expressiveness. The English HMM-based TTS system proposed by Nitech uses ToBi labels (Silverman et al., 1992) which have been widely studied and applied to many other systems. Unlike the English approach, we propose to control prosodic events for the Spanish approach using two linguistic contents (Garrido, 2001): Accentual (AG) and Intonational (IG) group. These parameters were successfully applied in a Spanish F0 estimator based on a Case Based Reasoning (CBR) machine learning approach (Iriondo et al., 2006) (see also Section A.1).

Apart from the IG and AG, questions are described using unit features and contextual factors. The former describes each phoneme linguistically and the latter enumerates what type of surrounding information is used for phonemes, words and utterances. In both cases, phonetic labelling and text

[18] This voice is part of the Artic database (Kominek and Black, 2004), in particular, SLT. Note that English database uses its own contextual factors as described by (Tokuda et al., 2002b). A greater number of nodes in the resulting English tree is due to the different corpus length (English, 56 minutes in comparison to 49 minutes for Spanish) and the fact that both systems use different number of contextual factors.

analysis has been performed using a rule based approach (SinLib, see Section E) [19].

Attributes		Description	
PrvPrvPh	PosCWordinCPhrasebc		
PrvPh	GPOSNWord		
CPh	nSyllNWord		
NPh	nSyllPPhrase		
NNPh	nWordPPhrase		
PrvStress	nSyllCPhrase		
nPhPrvGA	nWordCPhrase	Prv	Previous
PosAGinIG	PosCPhraseinUttfw	Ph	Phoneme
CSyllStress	PosCPhraseinUttbc	C	Current
nPhCGA	ToneCPhrase	N	Next
PosCSyllinCWordfw	nSyllNPhrase	n	Number of
PosCSyllinCWordbc	nWordNPhrase	Pos	Position
PosCSyllinCPhrasefw	nSyllUtt	fr	From
PosCSyllinCPhrasebc	nWordUtt	in	In
nStressedSyllBefCSyllinCPhrase	nPhraseUtt	Syll	Syllable
nStressedSyllAfterCSyllinCPhrase	PosPhinIG	fw	Forward
nSyllfrPrvStressedSylltoCSyll	PosAccent	Bc	Backward
nSyllfrCSylltoNStressedSyll	CPhBeginWord	GPOS	Part-of-Speech
NameCVowel	CPhEndWord	AG	Accentual Group
NSyllStress	CPhBeginAG	IG	Intonation Group
nPhNGA	CPhEndAG	Utt	Utterance
GPOSPWord	CPhBeginIG		
nSyllPWord	CPhEndIG		
GPOSCWord	CPhBeginSyll		
nSyllCWord	CPhEndSyll		
PosCWordinCPhrasefw			

Table 2.4: Group of 51 attributes and their description used to build the Spanish questions for decision tree-based clustering. These attributes can be either numeric or deterministic. They are self-descriptive using the labels from the "Description" column. Hence, for example, "PosC-WordinCPhrasefw" stands for "Position of Current Word in Current Phrase reading it forward".

Table 2.4 summarizes the information used to build the questions. The attributes used are described as follows:

1. **Phonemes**. The grapheme-to-phoneme conversion uses 36 units (see table 2.5) with one model for the silence because types of silences are POS-tagged. It is important to note that the system performs a co-articulatory transcription, thus rules are also applied between words (e.g., /**b**arko/ and /mi**B**arko/, translated as, "ship" and "my ship"). The following two types of units are considered:

   - **Vowels**. Two vowel models are using depending whether they are stressed or not (Lambert and Breen, 2004). In addition, the system differentiates various types of vowels (see Table 2.5): frontal, back, semi-vowel, half open, open, closed and half closed.
   - **Consonants**. A total of thirteen groups are used to classify the consonants depending on the typical linguistic information.

   Assuming the current phone is p_i, a previous (p_{i-2}, p_{i-1}) and a next (p_{i+1}, p_{i+2}) context is used.

[19] The first version of the Spanish labelling was performed using the open source Festival system (Black et al., 1999).

2. **AG** and **IG**. The information included is (see Section A.1 for specific details):
   - AG type. This group incorporates syllable influence and is related to speech rhythm. The type of AG depends on the position of the accented syllable in the word.
   - IG type. There are three types of sentences: interrogative, declarative and exclamative. Questions include the number of phonemes in current, previous and next AG; start/end flag and type of AG.
   - AGs and IGs start/end flags. Whether a phoneme is the start/end of an AG or an IG. In addition, types of IG.
3. **Syllable**. Basically, information included has to do with stress and position. Thus, the following contextual information is included: stress of current, previous and next syllables; forward and backward position of current syllable in current word and in current phrase; number of stressed syllables with respect to contextual syllables (assuming that current syllable is s_i, thus we have $s_{i-2}, s_{i-1}, s_{i+1}, s_{i+2}$); syllable's vowel and start/end flag.
4. **Word**. It takes into account Part-Of-Speech (POS) for current, previous and next word's Part-Of-Speech (POS); the number of syllables of current, next and previous words and position (forward and backward) of word in phrase and start/end flag.
5. **Phrase**. It involves information about the number of syllables and number of words in current, previous and next phrases; positions (forward and backward) of current phrase in the utterance.
6. **Utterance**. It includes the number of syllables, words and phrases in the utterance.

2.9 Conclusion

In this chapter HMM-based TTS system has been described in detail. Firstly, a brief historic outline has been introduced presenting the most important contributions to the HMM synthesis systems since the early 90s. A theoretical HMM definition has been given in order to define all the necessary background needed to describe the parameter generation algorithm, HMM training process, adaptation techniques and decision-tree based clustering.

From this chapter, it is important to highlight the following contributions. Firstly, the proposed F0 enhancing technique using an external F0 estimator which attempts to enhance the flat expressiveness of the conventional system (see Section 2.6.3). Secondly, details of the contextual clustering for Spanish HMM-based TTS systems have also been described in Section 2.8.3. The main difference of the Spanish system with respect to other languages is the use of AG and IG to improve expressiveness within the HMM framework.

In the following chapter, data parameterization is explained from the HMM point of view, detailing stream composite form.

Vowels	
Semi consonant	j,w
Half open	E,O
Open	a
Closed	j,w,i,u
Half closed	e,o
Frontal vowels	j,i,I,e,E,a,A
Back vowels	o,O,u,U,w

Consonants	
Dental	t,d
Velar	k,g,N,x,G
Bilabial	p,b,m,B
Alveolar	n,s,R,r,l
Palatal	J,L
Labiodental	M,f
Interdental	T,D
Plosive	p,b,t,d,k,g
Nasal	m,n,J,N,M
Fricative	B,f,tS,T,D,s,x,G
Lateral	l,L
Rhotic	R,r
Unvoiced	C,f,k,p,s,t,T,x

Table 2.5: Castilian Spanish consonants and vowels inventory (SAMPA (Llisterri and Mario, 1993)). Capital vowels refer to stressed units.

Chapter 3

Data parameterization and modelling for HMM-based speech synthesis

As it has been described in Chapter 2, HMM-based TTS system uses parameterized data in order to perform a statistical modelling. Speech is therefore represented by a set of coefficients which in a source-filter approach are vocal-tract (Section 3.3), F0 (Section 3.4) and mixed excitation (Section 3.5). In this Chapter, different approaches are described for each of these parameters. In order to improve the naturalness, we propose some changes to the mixed excitation in Section 3.5.3 along with other similar approaches.

3.1 Introduction

A sequence of observations which can represent any source of information (e.g., speech, image or written text) can be used to train HMMs within its statistical framework. Those coefficients are usually modelled using a multivariate Gaussian Mixture distribution. As described in Chapter 2, different streams are used to model each type (see Figure 3.1 where the three basic streams are depicted).

In order to model the vocal-tract information within an HMM, it is necessary to convert the spectrum envelope into a set of coefficients by using the same speech coding techniques used for speech recognition. In this case though, coefficients must provide a good speech quality during speech reconstruction. In the following sections, some of the most common parameter types are discussed with respect to their use in HMM speech synthesis.

- Linear Predictive Coding (LPC). It represents the spectral envelope of a speech signal in a compressed form using information of a linear predictive model and it is based on Auto

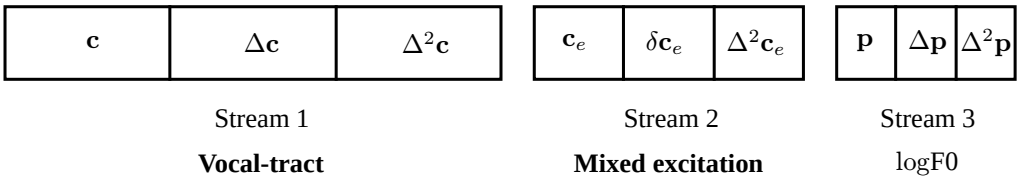


Figure 3.1: Feature vectors for each stream. Each of these blocks contains information of the static and dynamic features. Dimension for each stream depends on the approach taken discussed in this Chapter. Briefly, vocal-tract uses the highest number of coefficients (e.g., 25-th or 40-th order) whereas mixed excitation parameters are usually around 5-th order feature vector for a sampling rate of 16kHz. Fundamental frequency is modelled by one dimension.

Regressive (AR) model representation.

- Linear Spectral Pair (LSP). It is used to represent LPC for transmission over a channel. LSPs have several properties (e.g. smaller sensitivity to quantisation noise) that make them superior to direct quantisation of LPCs.

- Mel-cepstrum. It works in the mel-scale and is oriented to a perfect reconstruction using an approximation filter. Unlike MFCC that uses a filter bank to compute the coefficients, mel-cepstrum uses a warped phase to approximate the mel-scale.

Mixed excitation information has to do with a measure of "voicedness" which is represented in different ways depending on the type of approach. On the one hand, it is referred as voicing strengths for one of the multiband excitation in Section 3.5.3 or aperiodicity in Section 3.5.4. This "voicedness" information is often a continuous representation on the frequency domain. As for the spectrum, in order to model it within each state of the HMM, an average is performed in different frequency bands. In addition, in our proposal for multiband excitation in Section 3.5.3, a set of Fourier components are also used to reconstruct the excitation. In this case, extra streams are needed to model these information. Their dimension is the number of Fourier components.

Fundamental frequency described in Section 3.4 is stored in a one dimension stream in logarithmic domain in order to facilitate the correct modelling of small variations. Basically, two types of F0 modelling has been reported in the literature. On the one hand, F0 modelled by the so-called MultiSpace Distribution (MSD) where voiced and unvoiced frames are treated independently (see Section 3.4.1). On the other hand, unvoiced frames can also be modelled by a global Gaussian distribution (see Section 3.4.2). Note that unlike reported by (Zen et al., 2009), the F0 enhancement technique proposed in Section 2.6.3 is not considered an HMM F0 modelling approach here since the purpose of this approach is to enhance F0 once this has been generated from the HMMs.

3.2 Dynamic features

Static speech features are augmented by dynamic features which contain information about the variations of the original parameters. Dynamic features are a common technique and they have already been used in speech recognition applications in order to increase the accuracy of the system (Rathinavelu and Deng, 1995). TTS systems based on HMMs use a parameter generation algorithm which applies dynamic features as a constraint to obtain parameters able to produce a more natural speech. Thus, dynamic features are included in the model in the form of Δ and Δ^2 coefficients.

Dynamic features can be interpreted as the first and second derivatives of original data. Depending on the weights w the coefficients are calculated as,

$$\Delta c_t = \sum_{\tau=L_-^1}^{L_+^1} \omega^{(1)}(\tau) c_{t+\tau} \tag{3.1}$$

$$\Delta^2 c_t = \sum_{\tau=L_-^2}^{L_+^2} \omega^{(2)}(\tau) c_{t+\tau} \tag{3.2}$$

where L_s^i configures the window length of delta Δ ($i = 1$) and delta-delta Δ^2 ($i = 2$, second order coefficients or acceleration) from the left $s = -$ until the right $s = +$ side.

The proposed dynamic features in (HTS, a) use $w^{(1)} = [-0.5, 0, 0.5]$ as the window vector for Δ and $w^{(2)} = [1, -2, 1]$ for Δ^2, so they have an $L_-^1 = L_-^2 = -1$ and an $L_+^1 = L_+^2 = 1$. The dynamic coefficients are then $\Delta \mathbf{c}_t = 0.5\,(\mathbf{c}_{t+1} - \mathbf{c}_{t-1})$ and $\Delta^2 \mathbf{c}_t = \mathbf{c}_{t-1} - 2\mathbf{c}_t + \mathbf{c}_{t+1}$.

The effect of a reconstruction using dynamic coefficients is depicted in Figure 3.2. These coefficients smooth the speech envelope and improve the naturalness of the synthetic speech since they provide a more accurate modelling giving dynamism to the speech generation algorithm (see Section 2.5.1).

3.3 Vocal tract modelling

In the following Sections, vocal-tract modelling coefficients are described in detail relating their use in HMM-based TTS systems.

3.3.1 Linear Predictive Coding

The Linear Predictive Coding (LPC) is one of the most used techniques to linearly parameterize a speech signal. The main advantage of linear prediction is that it provides an automatic means for separating the source and filter components in speech. Therefore, it is important to introduce this kind of parameters due to its wide use in source-filter model applications and because of its

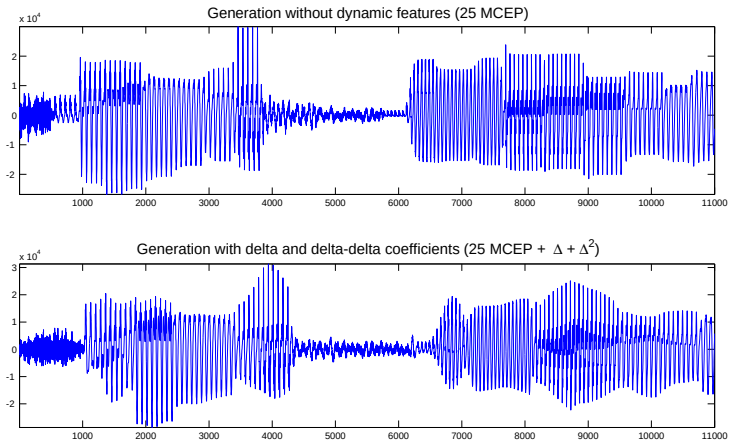


Figure 3.2: Example of the effect of dynamic features in the generation algorithm used in an HMM-based TTS system. "Se imagina un" in Spanish ("Imagine a" in English).

influence on the improvement of the speech naturalness using Mixed Excitation Linear Prediction (MELP) (McCree and Barnwell, 1995).

LPC approach calculates the coefficients of a p-th order linear system (FIR filter) that better predicts, in the mean square error sense, the current value of the real-valued time series $s[n]$ based on past samples (see Equation 3.3). This describes an AR model based on an all-pole filter. The best order of the LP coefficients (p) for speech signals can be obtained using Equation 3.4 [1],

$$\hat{s}[n] = \sum_{i=1}^{p} a_i s[n-i] \tag{3.3}$$

$$p = 2 + \frac{s_r}{1000} \tag{3.4}$$

where s_r stands for sample rate in Hz.

The LPC vocal-tract filter response is defined as,

$$H(z) = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A_p(z)} \tag{3.5}$$

During analysis, an ideal AR process filtering $s[n]$ using the FIR filter $1/H(z)$ would create a perfect white process, also called prediction error $e[n]$. $H(z)$ is, then, a whitening filter as it keeps the spectral envelope of the input signal. During synthesis, spectral behaviour of the process $s[n]$

[1] The origin of Equation 3.4 is related to the tube vocal-tract model (Taylor, 2009). Taking into account the required number of tubes to model a standard vocal-tract, it is assumed that a tube section is required per 1000 Hz sampling rate and each tube section gives rise to one pole. Finally, Equation 3.4 considers the source and radiation filters to be modelled by one pole each.

can be generated filtering the error with the filter $H(z)$.

As we can see, one of the main weaknesses of LPC is that all information relies upon the sharpness of the poles. As we have seen in Section 2.6, in the HMM-based TTS system framework, the statistical process smoothes the generated parameter sequence (and therefore, in this case, the poles) which results on a smoothed spectral envelope. Thus, the training process dramatically affects the formants which in fact reduces the general intelligibility of the phonemes and words. In general we can say that LPCs are very sensitive to quantization and manipulation and for those reasons, they are not suitable for statistical synthesis.

3.3.2 Line Spectral Pairs (LSP)

This representation (Sugamura and Itakura, 1986) is an equivalent parameter to LPC. It has excellent quantization and interpolation properties, which have been very useful in the research of training neural networks for speech synthesis as reported with an HMM-based TTS system by (Ling et al., 2006) and in low bit rate coding (Chu, 2003).

LSP-type parameters characteristics seem to be valuable for statistical parametric synthesis because statistical modelling is closely related to quantization and synthesis is closely related to interpolation (Zen et al., 2009). Although the various advantages over cepstral coefficients, it is well known that as long as the LSP coefficients are within $[0, \pi]$ and in ascending order, the resulting synthesis filter will be stable. However, it is difficult to guarantee whether LSPs generated from HMMs will satisfy these properties because state-output distributions are usually Gaussian distributions with diagonal covariance matrices.

LSP coding represents the frequency of the zeros of two polynomials $P(z)$ and $Q(z)$ which are related to the predictor polynomial $A_p(z)$ in Equation 3.5 by the following equations:

$$\begin{aligned} P(z^{-1}) &= A_p(z^{-1}) - z^{-(p+1)}A_p(z) = \\ &= 1 + (a_1 - a_p)z^{-1} + \ldots + (a_p - a_1)z^{-p} - z^{-(p+1)} \end{aligned} \tag{3.6}$$

$$\begin{aligned} Q(z^{-1}) &= A_p(z^{-1}) + z^{-(p+1)}A_p(z) = \\ &= 1 + (a_1 + a_p)z^{-1} + \ldots + (a_p + a_1)z^{-p} + z^{-(p+1)} \end{aligned} \tag{3.7}$$

where p is the order of polynomial $A(z)$. The zeros of $P(z)$ and $Q(z)$ lie on the unit circle in the z plane, and this reduction in the search space allows efficient root finding methods to be employed (the roots of $A(z)$ can also form a useful parameter set, however the root-finding process is computationally expensive). For the synthesis filter to be stable, the zeros of $P(z)$ alternate around the unit circle with the zeros of $Q(z)$.

3.3.3 Mel-cepstral modelling

Human sensitivity to the frequency scale is not linear. The mel-scale was a new frequency range that was defined to be more similar to the human sensitivity. Several methods have been proposed in order to obtain mel-cepstral coefficients. In particular, mel-cepstral analysis by (Fukada et al., 1992) was designed as a consistent method where during analysis, speech spectrum is modelled by L mel-cepstral coefficients minimizing the unbiased estimation of log spectrum using an adaptive procedure. Mel-Log Spectrum Approximation (MLSA) filter (Imai, 1983) is used to directly synthesize speech from the mel-cepstral coefficients. As in any source-filter model approach, an excitation is also needed.

In the mel-cepstral analysis (Fukada et al., 1992), the vocal tract transfer function $H(z)$ is modelled by an exponential in order to guarantee the stability. It consists of a parameter vector $\mathbf{c} = [c(0), c(1), \ldots, c(L)]^{\mathrm{T}}$ and it is defined as,

$$H(z) = \exp\left(\sum_{m=0}^{L} c(m)\tilde{z}^{-m}\right) = \exp\left(\mathbf{c}^{\mathrm{T}}\tilde{\mathbf{z}}\right) \tag{3.8}$$

where $\tilde{\mathbf{z}} = [1, \tilde{z}^{-1}, \ldots, \tilde{z}^{-L}]^{\mathrm{T}}$. The system $\tilde{z}^{-1}$ is defined as a first order all-pass function,

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \quad |\alpha| < 1 \tag{3.9}$$

with the following phase characteristic:

$$\tilde{\omega} = \tan^{-1} \frac{(1 - \alpha^2)\sin(\omega)}{(1 + \alpha^2)\cos(\omega - 2\alpha)} \tag{3.10}$$

The mel-scale can be approximated by the phase characteristic of this first order all-pass filter with an appropriate choice of α (frequency factor). See Section 3.6 and the work presented by (Masuko, 2002) for more information about the frequency factor and different sampling rates.

It has been shown that the mel-cepstral parameterization can be obtained by minimizing the Unbiased Estimation of Log Spectrum (UELS) (Imai and Furuichi, 1988) with respect to the mel-cepstral coefficients $\mathbf{c}$. Unlike LPC parameters that can only predict the poles of the transfer function, valleys can also be estimated using mel-cepstral coefficients based on the minimization of the spectral criterion. UELS assumes E as the criterion to be minimized in order to obtain an unbiased estimate in the sense of relative power of the spectrum $|H(e^{j\omega})|^2$,

$$E = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left(\exp\left(R(\omega)\right) - R(\omega) - 1\right) d\omega \tag{3.11}$$

where R is

$$R(\omega) = \log\left(I_N(\omega)\right) - \log\left|H(e^{j\omega})\right|^2 \tag{3.12}$$

and $I_N(\omega)$ is the modified periodogram of weakly stationary process $x(n)$ given by,

$$I_N(\omega) = \frac{\left|\sum_{n=0}^{N-1} w(n)x(n)e^{-j\omega n}\right|^2}{\sum_{n=0}^{N-1} w^2(n)} \tag{3.13}$$

where $w(n)$ is the window of length N.

By taking the gain factor K outside from $H(z)$, the minimization criterion E with respect to $\mathbf{c}$ results on the minimization of the filter coefficients $\mathbf{b}$ and K separately. The transfer function can be rewritten as,

$$H(z) = K \cdot D(z) = K \cdot \exp\left(\sum_{m=1}^{L} b(m)\Phi_m(z)\right) \tag{3.14}$$

where $K = \exp(b(0))$. The relation between $\mathbf{c}$ and the filter coefficients $\mathbf{b}$ is,

$$c(m) = \begin{cases} b(m) & m = L \\ b(m) + \alpha b(m+1) & 0 \leq m < L \end{cases} \tag{3.15}$$

and

$$\Phi_m(z) = \frac{(1-\alpha^2)z^{-1}}{1-\alpha z^{-1}} z^{-(m-1)} \tag{3.16}$$

Since $H(z)$ is a minimum phase system, it is possible to set the minimization of E with respect to $\mathbf{c}$ to be equivalent to that of ε with respect to $\mathbf{b}$, where ε is

$$\varepsilon = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{I_N(\omega)}{|D(e^{j\omega})|^2} d\omega \tag{3.17}$$

When minimizing E by setting $\partial E/\partial K = 0$, the following gain factor K is obtained

$$K = \sqrt{\varepsilon_{min}} \tag{3.18}$$

The minimization problem to compute $\mathbf{b}$ can be solved by the Newton-Raphson method (see the adaptive diagram in Figure 3.3). For i-th iteration, the coefficients are estimated as:

$$\mathbf{b}^{(i+1)} = \mathbf{b}^{(i)} - \mu \nabla\varepsilon \mid_{\mathbf{b}=\mathbf{b}^{(i)}} \tag{3.19}$$

where μ is the step size and the gradient of the criterion ε is a function of the outputs of $D^{-1}(z)$ and $\Phi_m(z)$ (Fukada et al., 1992).

To synthesize speech from the mel-cepstral coefficients, the exponential transfer function in Equation 3.8 must be performed efficiently. Although the transfer function is not a rational function, the Mel Logarithmic Spectrum Approximation (MLSA) filter can approximate it with sufficient accuracy (Masuko, 2002).

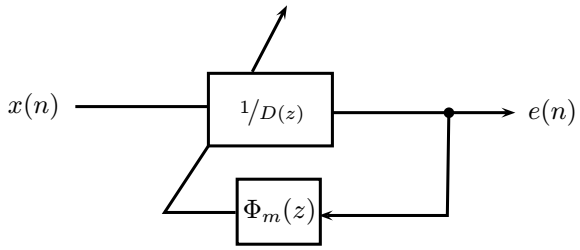


Figure 3.3: Block diagram of the adaptive mel-ceptral analysis.

3.3.3.1 Post-filter

Post-filter is a known technique used in many vocoders in order to bright the synthesized speech by emphasizing formants. For mel-cepstral coefficients (Yoshimura et al., 2001), let β be a coefficient expressing the intensity of the post-filtering. When $\beta = 0$ no post-filtering is applied, while if $\beta > 0$ spectra is emphasized. If we apply this parameter to the transfer function in Equation 3.14 in the b domain, it becomes,

$$D_\beta(z) = \exp\left(\beta \sum_{m=1}^{L} \tilde{b}(m)\Phi_m(z)\right) \tag{3.20}$$

where:

$$\tilde{b}(m) = \begin{cases} b(m) & 2 \leq m \leq L \\ -\alpha b(2) & m = 1 \end{cases} \tag{3.21}$$

Note that Equation 3.15 has been replaced by Equation 3.21 where $c(1) = 0$ in order to avoid emphasizing the overall trend of the spectra.

3.3.4 Mel-generalised parameterization

Although cepstral modelling can represent poles and zeros with equal weights, the mel-cepstral method from previous section overestimates the bandwidths of the formants with a small number of coefficients. To alleviate this problem, the generalized cepstrum is proposed by (Tokuda et al., 1994b).

The cepstrum (see Equation 3.22) is defined as the inverse Fourier transform ($\mathcal{F}^{-1}$) of the logarithmic magnitude of the Fourier transform ($\mathcal{F}$) of the signal.

$$c[n] = \mathcal{F}^{-1}\left\{\log\left|\mathcal{F}\left\{x[n]\right\}\right|\right\} \tag{3.22}$$

Unlike the conventional cepstrum, the Mel-Generalized Cepstrum (MGC) is defined as the inverse Fourier transform of the generalized logarithmic spectrum of the signal calculated on a warped

frequency scale $\tilde{w}$ of Equation 3.10.

$$c_\gamma[n] = \mathcal{F}^{-1}\left\{s_\gamma|\mathcal{F}\left\{x[n]\right\}|\right\} \tag{3.23}$$

where $s_\gamma(a)$ is the generalized logarithm function (Kobayashi and Imai, 1984),

$$s_\gamma(a) = \begin{cases} \frac{(a^\gamma - 1)}{\gamma} & 0 < |\gamma| \leq 1 \\ \log a & \gamma = 0 \end{cases} \qquad s_\gamma^{-1}(a) = \begin{cases} (1+\gamma a)^{\frac{1}{\gamma}} & 0 < |\gamma| \leq 1 \\ \exp a & \gamma = 0 \end{cases} \tag{3.24}$$

The MGC analysis (Tokuda et al., 1994b) is proposed as a unified approach where different spectrum representation can be obtained setting a pair (α, γ), where α controls the frequency warping and γ is the weight for pole/zero representation. This unified approach assumes that a speech spectrum $H(z)$ is modelled by the MGC coefficients $\mathbf{c}$ as,

$$\begin{aligned} H(z) &= s_\gamma^{-1}\left(\sum_{m=0}^{L} c(m)\tilde{z}^{-m}\right) \\ &= \begin{cases} \left(1+\gamma\sum_{m=0}^{L} c(m)\tilde{z}^{-m}\right)^{1/\gamma} & -1 \leq \gamma < 0 \\ \exp\left(\sum_{m=0}^{L} c(m)\tilde{z}^{-m}\right) & \gamma = 0 \end{cases} \end{aligned} \tag{3.25}$$

where $\tilde{z}$ is the all-pass filter defined in Equation 3.9. Different configurations based on the pair (α, γ) are presented in Table 3.1. Note that, from Equation 3.25 with $\gamma = 0$, mel-cepstral analysis from previous section is performed.

α	γ	**Method**
0	$-1 \leq \gamma \leq 0$	Generalized cepstral analysis
	-1	Linear prediction
	0	Unbiased estimation of log spectrum
$\lvert\alpha\rvert \leq 1$	$-1 \leq \gamma \leq 0$	Mel-generalized cepstral analysis
	-1	Warped linear prediction
	0	Mel-cepstral analysis

Table 3.1: Different forms of model spectrum as a function of the pair (α, γ).

With the unified approach using (α, γ) it is possible to create a wide range of model spectrums. An interesting approach is based on a system using MGC-LSP ($\alpha = 0.42, \gamma = -1/3$) (Zen et al., 2006). In this case, MGC-LSP parameters has better quantization/interpolation property than LSP or mel-cepstral coefficients.

3.3.5 A better spectrum estimate through STRAIGHT

Theoretically, a more accurate spectral envelope representation can be achieved increasing the number of parameters. However, a common problem of standard parameterization approaches described in previous sections, is to capture harmonics instead of the actual spectral envelope, specially when the order of the coefficients is very high. This problem is known as the F0 effect since the low frequency part of the spectrum envelope is affected by the F0 periodicity (see an example of this problem in Figure 3.4). In practise, this problem restricts the maximum order of the parameters that can be used limiting the resolution of the vocal-tract estimation. Thus, in order to be able to use higher orders, a better spectral envelope estimation is needed. For this purpose, a spectrum based on the STRAIGHT vocoding system (Kawahara et al., 2001) is used as described by (Zen et al., 2007a).

STRAIGHT is described as a very high quality speech analysis-modification-synthesis method implemented as a channel vocoder. It is based on the following main components: instantaneous-frequency-based F0 extraction and pitch-adaptive spectral smoothing to eliminate periodicity interference. In consequence, STRAIGHT produces a spectral envelope with reduced F0 effect and an aperiodicity measurement.

STRAIGHT was originally developed for reproducing a speech sound with a different fundamental frequency or even with a different vocal-tract length. For that purpose, the idea is to remove any periodicity interference in the spectrum in order to robustly reconstruct speech with modified F0 and vocal-tract parameters.

Within the STRAIGHT approach, F0 values are extracted with a fixed-point analysis. A F0-adaptive spectral analysis is combined with a surface reconstruction method in the time-frequency region to remove signal periodicity. It also extracts aperiodicity measurements on the frequency domain used by the synthesis stage (see Section 3.5.4 to see an excitation signal using this aperiodicity measurement).

Assume $S(\omega, t)$ is a three-dimensional space defined by the axes of time, frequency and amplitude (i.e., spectrogram). The goal is to remove any periodicity interference from this time-frequency representation based on the partial information given by an adaptive window analysis. This window has a Gaussian form in both time and frequency domains:

$$w(t) = \frac{1}{\tau_0(t_u)} \exp\left[-\pi \left(\frac{t}{\tau_0(t_u)}\right)^2\right] \tag{3.26}$$

$$W(\omega) = \frac{\tau_0(t_u)}{\sqrt{2\pi}} \exp\left[-\pi \left(\frac{\omega}{\omega_0(t_u)}\right)^2\right] \tag{3.27}$$

where $\tau_0(t_u) = {}^{2\pi}/_{\omega_0(t_u)}$ is the fundamental period that varies with time t_u referred to the whole utterance. In consequence, the analysis window size also adaptively follows the F0 change.

The problem of removing periodicity is considered to be a surface reconstruction process that

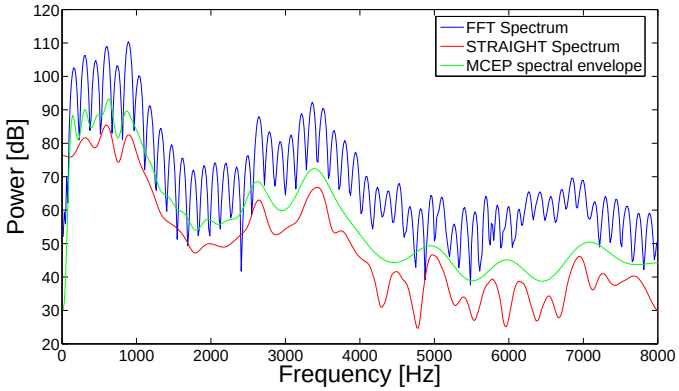


Figure 3.4: Example of the spectrum extracted with the FFT and compared with the spectrum envelope from STRAIGHT and mel-cepstral coefficients, respectively. Note that the spectrum approximation for the low frequency region is more accurate for STRAIGHT because it is not as affected by F0 harmonics as the mel-cepstral estimation.

can be solved by applying a smoothing function that provides an equivalent piecewise linear representation. In the approach proposed by (Kawahara, 1997), a convolution using a 2nd-order cardinal B-spline function is introduced. However, in a newer revision (Kawahara et al., 2001) it is proposed not to apply the spline smoothing function to the spectrogram but a pitch adaptive time window. Thus, to remove the periodic interference it is sufficient to perform the spline-based smoothing only in the frequency domain, once the temporal interference is effectively eliminated.

Let $w_p(t)$ be a pitch time window redefining Equation 3.26 and also based on a cardinal B-spline basis function and $w_c(t)$ a compensatory window that produces maxima where the original spectrogram has minimums:

$$\begin{aligned} w_p(t) &= \exp\left[-\pi\left(\frac{t}{\tau_0}\right)^2\right] \odot h\left(\frac{t}{\tau_0}\right) \\ w_c(t) &= w_p(t)\sin\left(\pi\frac{t}{\tau_0}\right) \end{aligned} \tag{3.28}$$

where $\odot$ refers to convolution and,

$$h(t) = \begin{cases} 1-|t| & |t|<1 \\ 0 & \text{otherwise} \end{cases}$$

Hence, a power spectrum with reduced phasic interference $P_r(\omega,t)$ is represented as a weighted squared sum of the power spectra $P_c(\omega,t)$ and $P_o(\omega,t)$ using the compensatory window (Equa-

tion 3.28) and the original time window (Equation 3.26), respectively,

$$P_r(\omega, t) = \sqrt{P_o^2(\omega, t) + \xi P_c^2(\omega, t)} \tag{3.29}$$

where ξ is selected to minimize the temporal variation of the resultant spectrogram (Kawahara et al., 2001).

It is important to note that the smoothing function successfully eliminates interferences caused by the signal's periodicity but also smoothes the underlying spectral structure. This causes a over-smoothing problem that is solved by recovering the original spectral values at each harmonic frequency applying a compensating operation which transforms the window to have only one non-zero value at each harmonic frequency.

The resulting spectrum envelope is a better representation of the input speech since it avoids the periodicity of the F0 in low frequency regions. With this spectrum envelope it is possible to increase the order of the conventional mel-cepstral coefficients which are extracted from the spectrum envelope with a recursive filter (Tokuda et al., 1994a). Increasing the order of coefficients yields to a better spectral representation which is very convenient for HMM synthesis.

3.4 Fundamental frequency (F0) modelling

Fundamental frequency (F0) is the lowest frequency in an harmonic series (i.e., the inverse of the time of the smallest repeating unit of the voiced speech signal). A general assumption is to consider F0 as a continuous contour in voiced regions and undefined in unvoiced. Depending on the method to handle unvoiced regions, we distinguish between continuous or discontinuous F0 modelling. Note that the F0 enhancing technique described in Section 2.6.3 is not considered to be an HMM F0 modelling technique.

3.4.1 Discontinuous F0 HMM

If the unvoiced regions of the F0 contour are considered as unknown, a discrete voiced/unvoiced (V/UV) indicator is required at each frame in order to model the F0 with HMMs. Several approaches deal with this type of approach. In the work presented by (Jensen et al., 1994), unvoiced symbols are explicitly modelled in the continuous HMMs by replacing each unvoiced symbol with "0" and adding an extra probability density function to each mixture for the unvoiced symbol. The approach presented by (Ross and Ostendorf, 1994) assumes that F0 values always exist but they cannot be observed in the unvoiced region and the training algorithm cannot be applied. In order to simultaneously model the discrete decision and the continuous F0 trajectory variables, a Multi-Space Distribution HMM (MSD-HMM) was introduced by (Tokuda et al., 1999). This approach is important for being widely used in the current design of the HTS system (HTS, a).

MSD-HMM can model sequences of observation vectors with a variable dimensionality including zero-dimensional observations, that is discrete symbols (Tokuda et al., 2002a). In an MSD-HMM, a sample space Ω is composed of G subspaces. Each subspace Ω_g has a weight and a continuous probability function with different dimensionality n_g. When the dimensionality of a certain space exists ($n_g > 0$), the MSD-HMM is the same as a continuous mixture whereas when $n_g = 0$ the MSD-HMM is the same as a discrete HMM.

In order to model the F0, the observation sequence of a F0 pattern is viewed as a mixed sequence of outputs from a one-dimensional space and a zero-dimensional space which correspond to voiced and unvoiced regions, respectively.

The state output distribution for an MSD-HMM is,

$$b_j(o_t) = \begin{cases} c_v \mathcal{N}(o_t|\mu_j, \sigma_j) & o_t \in \text{voiced region} \\ c_{uv} & o_t \in \text{unvoiced region} \end{cases} \tag{3.30}$$

where:

- o_t is the observation at time t,
- c_v and c_{uv} are the probabilities of voiced and unvoiced regions being $c_v + c_{uv} = 1$,
- μ_j and σ_j are the means and variances of the Gaussian distribution of the F0 in the voiced region in state j.

In order to accurately model F0 within the MSD-HMM framework, 3 MSD streams are used consisting of static F0 value at frame t and its Δ and Δ^2. Unvoiced regions are considered to be undefined values [2] and are represented by a distribution of zero length. During training, the weights are determined by the ratio of observed voiced or unvoiced regions. During synthesis, F0 for state j will be considered voiced if $c_v > \gamma$, where γ is the V/UV threshold [3].

MSD-HMM framework has some inherent limitations (Yu et al., 2009). Dynamic features in the boundary between voiced and unvoiced regions are not defined. As a consequence, during training, there will be a greater tendency to overfit the model and during synthesis, the reduced constraints on the temporal correlation may degrade the accuracy of the generated F0 trajectories.

Since $b_j(o_t)$ represents a continuous density in voiced regions and a discrete probability mass in unvoiced regions, each observation can only be either voiced or unvoiced, but not both at the same time. Consequently, during the forward-backward calculation for any F0 stream in training, the state posterior occupancy will always be wholly assigned to one of the two components depending on the voicing condition of the observation. This hard assignment limits the ability of the unvoiced component to learn from voiced data and vice versa, and it prevents any possibility of using a soft assignment to reduce the effect of F0 estimation errors.

[2] Usually, the actual implementation of an undefined value in the HTS system is $\log_e 0 \approx -1e^{10}$.
[3] Usually this value is usually fixed to 0.5.

A further problem is that the use of different F0 streams introduces redundant mixture weights (c_v and c_{uv}) for the dynamic features. Hence, the number of free parameters is increased during the state clustering which results in an underestimation of the cluster states. Furthermore, since the mixture weights for static and dynamic F0 streams are independent of each other, they may be very different in the final HMM after re-estimation.

3.4.2 Continuous F0 HMM

As we have seen in the previous section, the discontinuous F0 modelling requires the consideration of a discrete flag and consequently a discrete implementation of HMM for unvoiced regions. Moreover, the inconsistency of the transitions between voiced and unvoiced regions can degrade the F0 modelling resulting in an over-smoothed trajectory during synthesis.

In order to overcome these problems, a continuous approach defined as GTD (Globally Tied Distribution) is proposed by (Yu et al., 2009). The assumption of this approach is that F0 values exist in unvoiced regions but have markedly different statistical properties compared to their values in voiced regions. Due to the nature of unvoiced speech, mixtures corresponding to unvoiced F0 values are globally tied. The idea was first proposed by (Freij and Fallside, 1988) where each unvoiced symbol is replaced by a random vector generated from a probability density function with a large variance and then the random vectors explicitly modelled in the continuous HMMs.

Given the assumption that F0 is observable at every frame, dynamic features can be calculated straightforwardly. Consequently, static, Δ and Δ^2 can be modelled in a single stream. To model the statistical difference between voiced and unvoiced F0 values, a two component Gaussian mixture is used for the F0 stream. Hence, the globally tied distribution is used to model all unvoiced F0 values. The likelihood of observing the F0 feature o at state j is therefore given by,

$$b_j(o_t) = c_{uv}\mathcal{N}(o_t; \mu_{uv}, \sigma_{uv}) + c_v\mathcal{N}(o_t; \mu_j, \sigma_j) \tag{3.31}$$

where,

- c_v and c_{uv} are the probabilities of voiced and unvoiced regions so that $c_v + c_{uv} = 1$,
- μ_{uv} and σ_{uv} are the means and variances of the global shared Gaussian distribution of the F0 in the unvoiced region.
- μ_j and σ_j are the means and variances of the Gaussian distribution for state j of the F0 in the voiced regions.

In order to obtain F0 observations in unvoiced regions during the features extraction process, three techniques are analysed by (Yu et al., 2009):

1. 1-Best selection: the first F0 candidate output by the pitch tracker may be selected.

2. Interpolation-based selection: an interpolation is applied in the unvoiced regions and then the F0 candidate closest to the interpolated F0 track at each frame is selected.

3. Random-based selection. Pseudo-F0 observations could be used whereby log F0 values are sampled from a pre-defined Gaussian distribution with large variance.

During the synthesis stage, the required voicing classification is based on the component weights as in the MSD-HMM approach.

By making the unvoiced F0 existence assumption, the GTD approach has two advantages. First, there is only one single F0 stream so there are no redundant component weight parameters. Therefore, there will be no inconsistency in voicing classification which actually results on a better F0 contour generation in terms of expressiveness. Second, the continuous extraction of static and dynamic features solves the problem of dynamic features inconsistency of the MSD-HMM approach.

3.5 Excitation modelling

In a source-filter approach, the system requires filter coefficients and an excitation signal to be filtered. This excitation signal (or source signal) is a synthetic signal created in order to reproduce the original residual. The residual is the error that remains after codifying the speech by a set of coefficients (see Section 3.3). The purpose of the excitation signal is to approximate this residual during synthesis time. The quality of the excitation signal is measured in terms of how similar it is to the real residual signal. In general, it needs to incorporate information about periodic and aperiodic components.

Although excitations are not a specific topic of HMM-based TTS systems (in fact, excitation were originally designed for speech codifiers in order to improve the quality of transmitted speech at low bit rates (Chu, 2003)), the following section focus on different excitation approaches proposed for HMM synthesis. There are various systems to model the residual signal and to produce an excitation signal. Basically, the following types have been used in an HMM-based speech synthesis system:

- **Basic pulses excitation**. It is based on a simple model of speech, classifying parts as voiced and unvoiced frames.
- **Multi-band excitation**. In this approach, residual is modelled in B sub-bands. The amount of unvoiced information is properly weighted in each band.
- **Trainable excitation**. As any other wide-band signal, residual can be modelled using a ML technique.
- **Residual codebook**. The idea is to produce a more realistic excitation signal by residual manipulation.

All these types of excitation are described in the following sections in order to review current state-of-art. Basic excitation is the simplest of the techniques which is intended to be improved by using any type of mixed excitation (i.e., multiband, trainable or with residual codebooks). The purpose of any mixed excitation is twofold:

- Voiced part of basic excitation uses a flat spectrum. In this case, a mixed excitation will shape this spectrum in order improve the basic excitation's representation.
- Parameterize periodic and non-periodic components according to a measure of "voicedness" and mixture them together if necessary.

Experiments shown in Chapter 5 use multiband excitation approaches described in Section 3.5.2.

3.5.1 Basic excitation

This option is a straightforward design of the residual signal and has the advantage of being a low computationally cost method and to require no extra model information since it works only with the fundamental frequency period in samples (T_o). It uses the simplest excitation model using either a periodic impulse train or white noise (see Figure 3.5). This excitation is constructed using the following representation,

$$e[n] = \sum_{i=1}^{Z} \delta[n - p_i] \cdot a_i \tag{3.32}$$

where Z is the total number of pulses in an utterance, p_i is the position of pulse i (i.e., pitch marks) and a_i is the amplitude calculated so as each pitch period has power one,

$$a_i = \sqrt{p_{i+1} - p_i} = \sqrt{T_0} \tag{3.33}$$

The main disadvantage is that voice reconstructed with this excitation has a typical quality of

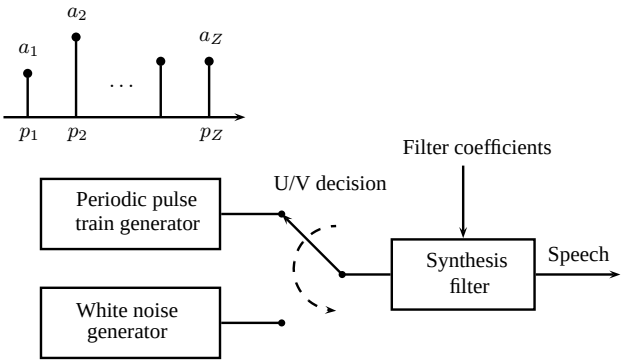


Figure 3.5: Basic excitation signal scheme based on periodic pulses and noise

"vocoded speech" since the model does not reflect the real information contained in the residual signal. In addition, the position of the deltas makes the system very sensitive to small variations in comparison with other speech production techniques such as sinusoidal modelling.

3.5.2 Multi-band excitation

The major effect of previous excitation is a buzzer-like quality perceived in the voiced parts of synthesized speech (Aoki et al., 2000). This undesirable degradation is perceived in synthesized voiced consonant. Buzzer-like quality of synthesized voiced consonants is attributed mainly to the insufficiency of the previous binary source signal model which exclusively switches between an impulse train and white noise. Since voiced consonants pronounced by human speakers do not necessarily show a perfect periodicity in all frequency bands, voiced consonants synthesized only by the impulse train may sound very different from the actual natural sounding.

A so-called Mixed Excitation (ME) can produce a more natural synthetic speech by employing an enhanced mixture of pulses and noise. A mixed excitation is based on modelling the multiband periodic and non-periodic information of the residual. Two methods have been proposed for HMM-based TTS systems:

- **Voicing strengths-based Mixed Excitation (VSME)** (Yoshimura et al., 2001; Gonzalvo et al., 2007b). It is based on Mixed Excitation Linear Prediction (MELP) (McCree and Barnwell, 1995), a method originally used for linear prediction coding. It models the excitation splitting the spectrum in a certain number of subbands in which periodic and non-periodic components are properly weighted. In the synthesis stage, subband weights are used to generate the excitation using the corresponding filter.

- **Aperiodicity-based Mixed Excitation (APME)** (Zen et al., 2007a). This technique generates a frequency continuous representation of the non-periodic component based on the STRAIGHT vocoding system (Kawahara et al., 2001). Unlike the VSME approach, APME reconstructs the excitation directly in the frequency-domain. Similarly, aperiodicity is averaged into a number of sub-bands in order to model it within an HMM.

It is a matter of fact that aperiodicity and voicing strengths attempt to model similar information. However, these parameters are extracted in different ways. On the one hand, voicing strengths are computed using a subband filtered input signal. On the other hand, aperiodicity is extracted in a smoothing process using F0 estimates.

3.5.3 Voicing strengths-based mixed excitation (VSME)

In this section, the proposed changes to the multiband mixed excitation are described. The original purpose of the multiband excitation was to overcome some of the limitations of the basic excitation

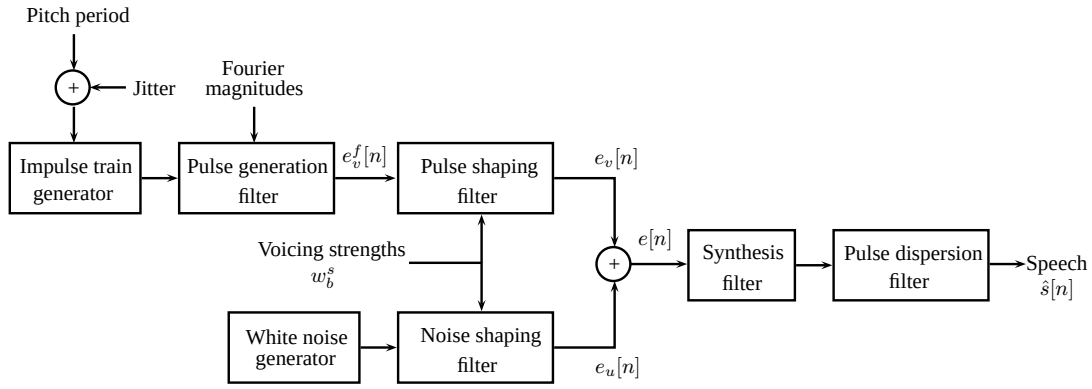


Figure 3.6: Reconstruction stage to create the multiband excitation signal ($e[n]$) using voiced $e_v[n]$ and unvoiced $e_u[n]$ parts. Reconstructed voiced excitation from complex amplitudes and phase is denoted as $e_v^f[n]$ and w_b^s are the voicing strengths for synthesis in band b.

in speech transmission applications. For synthesis purposes, some of the design constraints can be redefined. In particular, the number of parameters to store can be considerably increased. In consequence, the excitation signal can also be modelled more accurately.

As shown in Figure 3.6, the essential idea is the generation of an excitation signal that combines two components: a filtered periodic pulse with a filtered noise. This system is based on MELP, a technique originally developed for speech coding to overcome the problems of the pulse and noise excitation using LPC (McCree and Barnwell, 1995).

MELP approach used in this work includes the following modules (McCree and Barnwell, 1995):

- A randomly generated jitter period is used in order to perturb the value of the pitch period so as to generate a jittery impulse train. Unlike the pulses and noise approach, in this case the classification of each speech frame is not only voiced and unvoiced but also jittery voiced. The latter corresponds to the case when the excitation is aperiodic but not completely random during voicing transitions.

- Shape of the excitation pulse. Spectral shape of the original prediction error (i.e., residual signal) is captured by complex amplitudes. These are used to generate the impulse response of the pulse generation filter. Obviously, the prediction error is the result of parameterizing the input speech signal with the corresponding coefficient type, in this case, mel-cepstral (see Section 3.3.3).

- Periodic and noise excitations are filtered with the pulse and noise shaping filters, respectively. The output of the filters is added together to form the mixed excitation. Frequency response of the shaping filters is controlled by a set of parameters called voicing strengths, which measure the amount of "voicedness". Details are given in the following sections.

- Pulse dispersion filter (McCree and Barnwell, 1995). Once the speech is synthesized, it is fil-

tered by a pulse dispersion filter which is a 130-th order FIR derived from spectrally-flattened triangle pulse. The pulse dispersion filter can reduce some of the harsh quality of the synthesized speech.

The final excitation is generated using overlap-and-add in a pitch-period basis interpolating past and present frames. As described in Figure 3.6, firstly complex amplitudes are used to reconstruct the original residual spectral shape filtering pulses located at pitch periods. Then, depending on the voicing strengths, a pulse and a noise shaping filters are constructed. Once those filters are applied, the so-called mixed excitation is built adding the voiced and the unvoiced parts.

In order to use this type of excitation, the following parameters might be modelled in the HMM framework:

- B voicing strengths.
- K Fourier components (magnitude and phase).

3.5.3.1 Shaping filters

As we have seen in Figure 3.6, the VSME model uses two shaping filters. Filter's responses are controlled by a set of parameters called voicing strengths which are estimated from the original speech signal. By varying these strengths with time, a pair of time-varying multiband filters are obtained. These voicing strengths control the amount of pulse and noise in the excitation at various frequency bands.

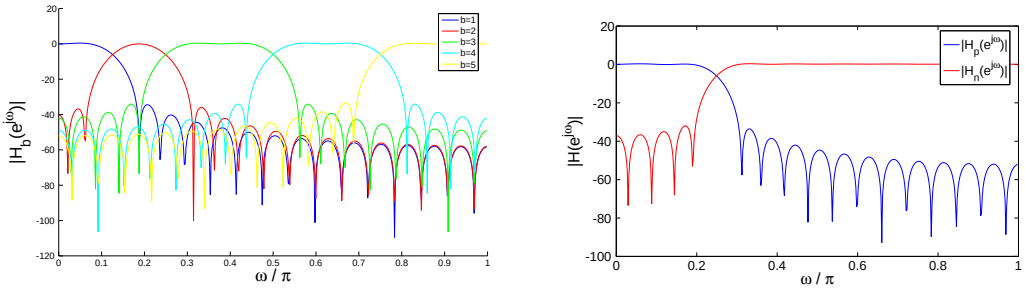
Each shaping filter is composed of $B = 5$ subbands controlled independently. For a sample rate of 16k Hz, passbands are defined as $[0, 1000]$, $[1000, 2000]$, $[2000, 4000]$, $[4000, 6000]$, $[6000, 8000]$ Hz (see the frequency response magnitudes in Figure 3.7(a)) (Yoshimura et al., 2001) [4]. Frequency response of the excitation shaping filter is defined by connecting these subband filters in parallel. Note that the gain of the parallel combination of two filters is always constant (see Figure 3.7(b) where an example of two complementary filters is shown). Each subband $b \in [1, 5]$ is controlled by a voicing strength factor (its filter gain), which is defined as $w_b^a \in [0, 1]$ for analysis and

$$w_b^s = \begin{cases} 1 & w_b^a > 0.5 \\ 0 & w_b^a \leq 0.5 \end{cases},$$

for synthesis. When $w_b^s = 0$, this band is synthesized as aperiodic and if $w_b^s = 1$, the band is considered as periodic.

Total frequency response of the pulse shaping filter is the sum of the subbands weighted by the

[4] For other sampling frequencies, refer to Section 3.6.



(a) Magnitude response of the synthesis shaping filters.

(b) Example of the synthesis shaping filters for pulse and noise ($w_1^s = w_2^s = 1, w_3^s = w_4^s = w_5^s = 0$).

Figure 3.7: Magnitude response of the synthesis shaping filters.

voicing strength:

$$h^{(p)}[n] = \sum_{b=1}^{5} w_b^s h_b[n] \tag{3.34}$$

where $h_b[n]$ is the filter frequency response for band b. The noise shaping filter has the following response:

$$h^{(n)}[n] = \sum_{b=1}^{5} (1 - w_b^s) h_b[n] \tag{3.35}$$

Shaping filters are implemented as FIR filters with 31 taps (Chu, 2003). FIR filters are utilized due to the following reasons:

- Linear phase. The group delay is constant with frequency thus non-linear phase distortion is avoided.
- Response of the total shaping filter can be easily obtained by summing operations.
- Impulse response of the shaping filters are interpolated during synthesis and by nature, FIR filters guarantee stability.

3.5.3.2 Voicing strengths

Voicing strengths are estimated in analysis time for each band utilizing a filter bank with five bands to separate the input speech signal ($s[n]$). Unlike the filters described for synthesis, analysis filters are sixth-order Butterworth (McCree et al., 1997). As we will see, these filters are only used for the correlation computation, thus the nonlinearity in phase response is not taken into account.

Voicing strengths are estimated by detecting periodicity in each corresponding subband using the normalized correlation coefficients computed around the pitch lag. For time delay t, this is defined

as,

$$c_b(t) = \frac{\sum_{n=0}^{N-1} s_b[n]s_b[n+t]}{\sqrt{\sum_{n=0}^{N-1} s_b[n]s_b[n] \sum_{n=0}^{N-1} s_b[n+t]s_b[n+t]}} \tag{3.36}$$

where $s_b[n]$ corresponds to the filtered input signal with bandpass filter b and N is the size of the pitch analysis window. For each band, voicing strengths are determined as follows.

- Low-band ($b = 1$). For this band, w_1^a is equal to the normalized autocorrelation value at pitch T_0 lag.

$$w_1^a = c_1(T_0)$$

- High-frequency bands $b \in [2, 5]$. The procedure is the following. Calculate two correlations:
  - Using the output of the corresponding passband filter ($s_b[n]$), calculate $r_1 = c_b(T_0)$.
  - Using the envelope of the bandpass signal ($\text{env}(s_b[n])$), calculate $r_2 = c_b^{(env)}(T_0)$. This envelope is obtained by full-wave rectification (absolute value in samples) followed by lowpass filtering which cancels the DC component.

  The first correlation works well with stationary signals but the correlation values can be too low in regions of varying pitch. This problem can be worse at higher frequencies. The second correlation makes envelopes to be clearly distinguishable especially at high frequencies.

  The maximum correlation factor is used as the overall voicing strength in the corresponding band:

$$w_b^a = \max(r_1, r_2)$$

As described by the MELP standard (McCree et al., 1997), the value of the voicing strength in each band is subjected to modifications according to other properties of the signal. Concretely, the peakiness of the residual signal ($e[n]$) is used as it refers to the presence of samples having relatively high magnitudes which is useful for voiced/unvoiced decision as well as in the characterization of transients in speech. The peakiness is calculated with the following equation:

$$p = \frac{\sqrt{\frac{1}{N} \sum_{n=-N/2}^{N/2} e^2[n]}}{\frac{1}{N} \sum_{n=-N/2}^{N/2} |e[n]|} \tag{3.37}$$

According to the value of peakiness, voicing strengths are modified as follows:

- if $p > 1.34$, then $w_1^a = 1$

- if $p > 1.6$, then $w_2^a = w_3^a = 1$

For the rest of subbands, voicing strengths are set to the estimated correlation coefficient.

In the standard description of MELP (McCree et al., 1997) there are a set of analysis windows defined for a sampling rate of 8k Hz. The basic frame length is $N = 180$ samples and positions vary according to the parameter being analyzed in order to facilitate interpolation (Chu, 2003). In this work, there is a single analysis window with $N = 360$ samples (sampling rate of 16k Hz).

Also, unlike the standard MELP system where the analysis stage was part of a real-time speech coder, mixed excitation model for HMM-based TTS systems uses an offline process during the voice building process. In that case, a parallel computation and refinement of the pitch along with the voicing strengths estimation is not required. For this reason, the analysis stage in this work utilizes a pre-estimated pitch based on the PMFA technique (Alías et al., 2006b). Voicing strengths are estimated a posterior.

3.5.3.3 Complex amplitudes

Assuming that voicing strengths can generate a mixed excitation with information for each sub-band, it is also possible to improve the model of this excitation by incorporating the complex amplitudes of the residual signal. The goal is to capture the shape of this signal and use this response during synthesis time in order to produce an excitation as close as possible to the original residual signal.

According to Figure 3.6, during synthesis, the pulse generation filter produces an enhanced pulse train using pitch periods and complex amplitudes. During analysis, a number of complex amplitudes of the spectrum peaks corresponding to the harmonics of the voiced speech frames are stored. Those peaks are extracted from the DFT of the residual signal.

The residual signal is obtained by inverse filtering. Modelling original signal $s[n]$ with vocal-tract filter coefficients $h[n]$ results in a prediction error $r[n]$, the so-called residual signal. In order to obtain this prediction error, the synthesis filter is inverted.

$$S(z) = H(z)R(z)$$
$$R(z) = \frac{S(z)}{H(z)}$$

Assuming mel-cepstral coefficients are used (see Section 3.3.3), magnitude response of the inverse filter is given by an inverted exponential transfer function (see Equation 3.14) [5].

$$H^{-1}(z) = K^{-1} \cdot D^{-1}(z) = K^{-1} \cdot \exp\left(-\sum_{m=1}^{L} b(m)\Phi_m(z)\right) \tag{3.38}$$

[5] This filter is efficiently implemented as described in Section 3.3.3. Basically, a rational function is used to approximate the exponential function.

In order to extract the magnitude peaks, the DFT of the voiced parts of the residual signal $r_v[n]$ is defined as

$$R_v[k] = \sum_{n=0}^{N_F-1} r_v[n] \exp\left(-j\frac{2\pi kn}{N_F}\right) \text{ for } k = [0, N_F - 1]$$

where N_F is the length of the DFT. The first K complex amplitudes are taken from the harmonics of $R_v[k]$ associated with the pitch frequency.

Due to the fact that the excitation signal is synthesized on a pitch-period basis and that we do not have all magnitudes, the reconstruction is performed by Equation 3.39. If T_0 is the pitch period in samples, the excitation reconstructed after the IDFT is

$$e_v^f[n] = \frac{1}{T_0} \sum_{k=0}^{T_0-1} E_v^f[k] \exp\left(j\frac{2\pi nk}{T_0}\right) \text{ for } n = [0, T_0 - 1] \tag{3.39}$$

where $E_v^f[k]$ is a symmetric sequence of length T_0 used for the generation of a real-valued sequence after the IDFT:

$$E_v^f[k] = \left[0, \check{R}_v[1], \check{R}_v[2], \ldots, \check{R}_v[K], 0, \ldots, 0, \underbrace{\check{R}_v^*[K]}_{(T_0-K)-th}, \check{R}_v^*[K-1], \ldots, \check{R}_v^*[1]\right]^{\mathrm{T}}$$

where the first element is set to zero in order to guarantee no DC component, $(\cdot)^*$ stands for complex conjugate and $\check{R}_v[k]$ are the residual complex amplitudes generated from the HMM.

In the standard definition of MELP (McCree et al., 1997), the number of magnitudes is chosen in order to get an optimal balance between quality and transmission rate ($K = 10$). However, in this work, the main constraint is quality rather than transmission efficiency and thereby the number of magnitudes selected to be trained in the HMMs is $K = 30$.

Nevertheless, note that during analysis and synthesis, the maximum number of magnitudes for each frame is limited by the fundamental frequency period T_0. During synthesis, the voiced excitation signal ($e_v^f[n]$) within a certain pitch period will be reconstructed from only $\tilde{K}, \tilde{K} < T_0$. During analysis, when the number of magnitudes that can be stored are less than expected ($\tilde{K} < K$), the rest of the magnitudes for that frame ($K - \tilde{K}$) are set to zero. Number of magnitudes $K = 30$ has been empirically fixed assuming a voice with a maximum pitch of 250 Hz so it can reconstruct an harmonic bandwidth of 250Hz $\cdot$ 30 = 7500 Hz.

3.5.4 Aperiodicity-based mixed excitation (APME)

This type of excitation uses a frequency domain representation of the aperiodicity to build a mixed excitation (Zen et al., 2007a; Yamagishi et al., 2007). Aperiodicity is defined as a measure of the energy on inharmonic frequencies normalized by the total energy (Kawahara et al., 2001). Similarly to the voicing strengths approach in Section 3.5.3.2, aperiodicity model also suppresses the buzziness

timbre of the basic excitation model described in Section 3.5.1. Nevertheless, aperiodicity offers a better overall quality than VSME as it gives a better representation of the "unvoicedness" of the residual signal (see experiment in Section 5.3.3) although it is computationally more expensive.

In order to model aperiodicity within the HMM framework, this is averaged into B subbands. Assuming that $P_{ap}(\omega)$ is the aperiodicity in the frequency domain and $P_{ap}[k]$ is the DFT representation, aperiodicity for band $b \in [1, B]$ is:

$$P_{ap}^{(b)} = \frac{1}{k_e^{(b)} - k_s^{(b)}} \sum_{k=k_s^{(b)}}^{k_e^{(b)}} P_{ap}[k] \tag{3.40}$$

where $k_s^{(b)}$ and $k_e^{(b)}$ are the frequency bins associated with $\omega_s^{(b)}$ and $\omega_e^{(b)}$, that is, the start and end frequencies for band b. As the VSME approach, the number of subbands modelled by the HMM is usually fixed to $B = 5$ for a sampling rate of 16k Hz (see Section 3.6 for other sampling frequencies).

During analysis, aperiodicity is extracted as follows. Let $s[n]$ be the input signal and $S_s(\omega)$ be the smoothed power spectrum obtained by applying a cepstral liftering $L(\omega)$ to suppress components having quefrencies greater than F0. Then, assume $\left|S_u(\omega)^2\right|$ and $\left|S_l(\omega)^2\right|$ represent the upper and the lower spectral envelopes, respectively. The upper envelope is calculated by connecting spectral peaks and the lower envelope is calculated by connecting spectral valleys. The aperiodicity measure is calculated as the lower spectral envelope normalized by the upper spectral envelope which in logarithmic domain yields to,

$$P_{ap}(\omega) = \mathcal{I}_e \left\{ w_{ERB}(\omega) \cdot \mathcal{T}_e \left\{ S_l(\omega) - S_u(\omega) \right\} \right\} \tag{3.41}$$

where $w_{ERB}(\omega)$ is a simplified auditory filter shape for smoothing the resulting power spectrum (Kawahara et al., 2001) and $\mathcal{I}_e\{\cdot\} \equiv \mathcal{T}_e^{-1}\{\cdot\}$ refers to ERB scale transformations (Moore and Glasberg, 1996).

A mixed excitation is generated as pitch-synchronous overlap-and-added pulses. These pulses are reconstructed in the frequency domain using the aperiodicity. So, assuming that at each pitch mark p_i the impulse response is $v_{p_i}[n]$, aperiodicity is used for its spectral shaping as follows::

$$v_{p_i}[n] = \underbrace{\mathcal{F}^{-1}\left\{ \Phi(\omega) \cdot P_{ap}^{-1}(\omega) \cdot \mathcal{F}\{a_i \delta[n - p_i]\} \right\}}_{\text{Periodic component}} + \underbrace{\mathcal{F}^{-1}\left\{ \Phi(\omega) \cdot P_{ap}(\omega) \right\}}_{\text{Aperiodic component}} \tag{3.42}$$

where $\Phi(\omega)$ is an all-pass filter that allows control of the excitation by phase manipulation (Kawahara, 1997), $P_{ap}^{-1}(\omega)$ is the inverse aperiodicity of Equation 3.41 and $\mathcal{F}\{a_i \delta[n - p_i]\}$ is the Fourier transformation of the pulse. The flat spectrum of the delta is shaped with the inverse of the aperiodicity as by definition, aperiodicity is defined as the normalized energy on the inharmonics frequencies.

3.5.5 Trainable excitation

This excitation approach was firstly proposed by (Maia et al., 2007). This could be considered as a particular case of the multiband excitation described in Section 3.5.2 since it is closely related. However, as depicted in Figure 3.8, this method is based on the principle of analysis-by-synthesis speech coders and consists in the optimization of state-dependent (Maia et al., 2008) filter coefficients through the minimization of the difference between synthetic excitation and residual [6]. Mixed excitation is produced by inputting a pulse train and white noise into two state-dependent filters. These filters are derived so as to maximize the likelihood of residual sequences over the corresponding states through an iterative process. Apart from filter determination, amplitudes and positions of the pulse train are also optimized in the sense of residual likelihood maximization during the referred closed-loop training.

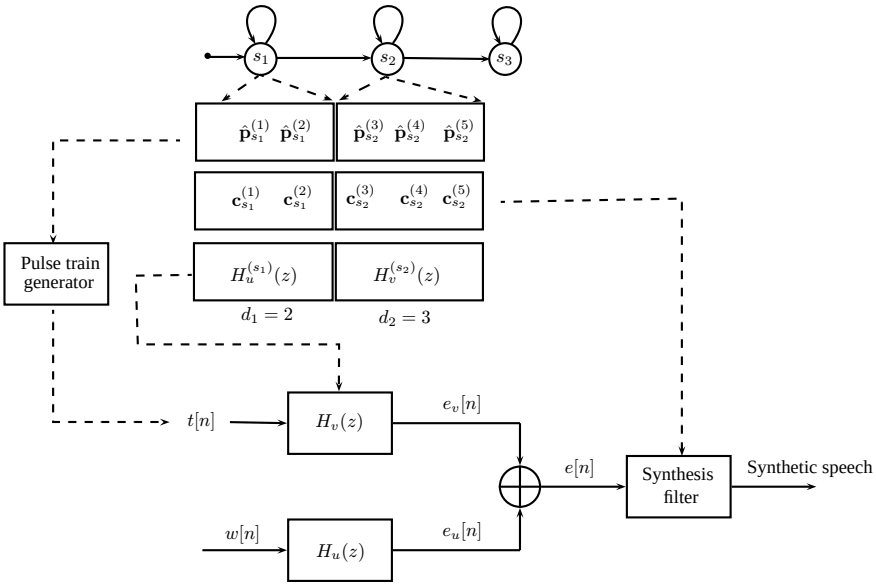


Figure 3.8: Trainable excitation scheme for synthesis.

The idea of this approach is to achieve an excitation signal whose waveform can be as close as possible to the residual signal. Therefore, the objective is to design a set of state dependent filters $H_v^{(s)}(z)$ and $H_u^{(s)}(z)$ and optimize the positions and amplitudes of $t[n]$. These filters are associated with each HMM state s (see Figure 3.8 for an example of state dependent filters associated with each HMM state). Since the minimization of the error between excitation and residual sequences is to determine the state dependent voiced and unvoiced filters, a likelihood of the residual signal is proposed by (Maia et al., 2007). Basically, the voiced filter is found by maximizing this likelihood whereas the unvoiced filter is set by autoregressive spectral estimation.

[6] This is obtained by inverse filtering the original signal as described in Section 3.5.3.2.

During synthesis time and according to Figure 3.8, the input pulse train ($t[n]$) and white noise sequence ($w[n]$) are filtered through $H_v(z)$ and $H_u(z)$, respectively, and summed together to result in the excitation signal $e[n]$. Transfer functions are given by,

$$H_v(z) = \sum_{l=-p_v/2}^{p_v/2} h_s(l) z^{-l} \tag{3.43}$$

$$H_u(z) = \frac{G}{1 - \sum_{l=1}^{p_u} g_s(l) z^{-l}} \tag{3.44}$$

where p_v and p_u are the respective filter orders. On the one hand, the purpose of $H_v(z)$ is to conform the spectral shape of the input pulse train $t[n]$ into $e_v[n]$ which intends to be similar to the voiced part of the residual sequence. On the other hand, $H_u(z)$ spectrally weights the noise ($w[n]$) into the unvoiced excitation $e_u[n]$ which is finally added to the voiced excitation $e_v[n]$.

3.5.6 Residual codebook

The idea presented by (Drugman et al., 2009) proposes the use of a codebook of pitch-synchronous residual frames in order to construct a more realistic excitation signal.

During the analysis stage, a limited codebook of typical residual representations is built from the training speech samples. Residual is obtained by the same filtering process described in the VSME approach (see Section 3.5.2). In this case, though, pitch-synchronous residual frames are resampled and normalized (RN) to obtain N_c samples in order to make it suitable for codebook modelling. These N_c coefficients are modelled in the HMM framework. This version of residual frames corresponds to a low-frequency signature of the underlying residual.

During synthesis time, apart from the typical vocal-tract and F0 coefficients, HMMs are used to generate the RN coefficients. These are used to retrieve a quantized representation of the target residual frames from the codebook. The excitation signal is constructed by overlap-and-adding pitch-sized residual frames and concatenating them together.

3.6 Using different sampling rates

The quality of a signal highly depends on the sampling rate. In the case of HMM-based TTS systems this is crucial as it increases the accuracy of the spectrum envelope and the brightness of the synthesized speech. In addition, the use of higher sampling rates clearly increases the number of parameters needed to model such wider spectrum. As a matter of fact, different sampling rates does not only change the vocal-tract but also the rest of parameters used by the HMM-based synthesis model.

In the current section, a proposal for adapting the use of different sampling rates is described

under the assumption that mel-cepstral coefficients (Section 3.3.3) and a multiband excitation (see Section 3.5.2) are used. In this type of excitation, "voicedness" information is parameterized in B subbands and this is what it is discussed in this section. For the rest of excitation types, a more complex analysis should be made. In addition, although this section could apply to either a VSME or an APME approach, only the latter has been tested in experiment of Section 5.3.3 since this one offers more naturalness than the VSME approach.

Basically, besides converting window lengths and frame steps, analysis and synthesis is performed as described in previous sections and the basic procedure is not modified. What does change is the vocal-tract and mixed excitation parameters.

Firstly, let's review the analysis of the mel-cepstral coefficients described in Section 3.3.3. Unlike the conventional MFCC, mel-cepstral coefficients are designed to minimize the energy of the periodogram using an iterative estimation of the coefficients for each frame. In fact, this approach eases the use of different sampling rates as it does not use frequency bands but a frequency warping function. Therefore, the frequency warping factor (α in Equation 3.9) is adapted for different sampling rates. This factor gives a control of the auditory frequency scale approximation at several sampling frequencies. In this work, new frequency warping factors have been computed in order to minimize the error of the approximation to auditory frequency scale in a similar way the original warping factors were proposed by (Fukada et al., 1992) using the phase response for mel-cepstral given by Equation 3.10,

Figure 3.9 shows the approximation result (frequency warping function) for each sampling frequency and the desired auditory mel-scale. As we can see, the higher the sampling rate, the worse the approximation is at low frequencies. In fact, 48k Hz is a very bad approximation of the mel-scale and this will affect the quality of the synthesized speech (see Experiment in Section 5.2.2). Table 3.2 shows the relation between the optimal α and its sampling rate. This optimal value of α has been found by performing an exhaustive exploration of all possible $\alpha \in [0.1, 0.9]$ and finding the minimum error between the mel-scale curve and each approximation using Equation 3.10 [7].

As described in Section 3.3.5, STRAIGHT is used to extract a better spectral envelope and therefore the order of the conventional mel-cepstral coefficients could be increased without suffering the so-called F0 effect. As we will see in the experiments (see Chapter 5), usually a 39-th order mel-cepstral analysis is used to model the STRAIGHT spectrum envelope for a sample rate of 16k Hz. Nevertheless, for higher samples rates, the number of mel-cepstral coefficients has not been increased due to two reasons:

- STRAIGHT envelope is more accurate than the conventional mel-cepstrum approximation

[7] Transformations between mel-scale (f_m) and linear (f_l) scales are obtained using the following Equations:

$$f_m = 1127 \log\left(1 + \frac{f_l}{700}\right) \text{[mel]}$$

$$f_l = 700\left(\exp\left(\frac{f_m}{1127}\right) - 1\right) \text{[Hz]}$$

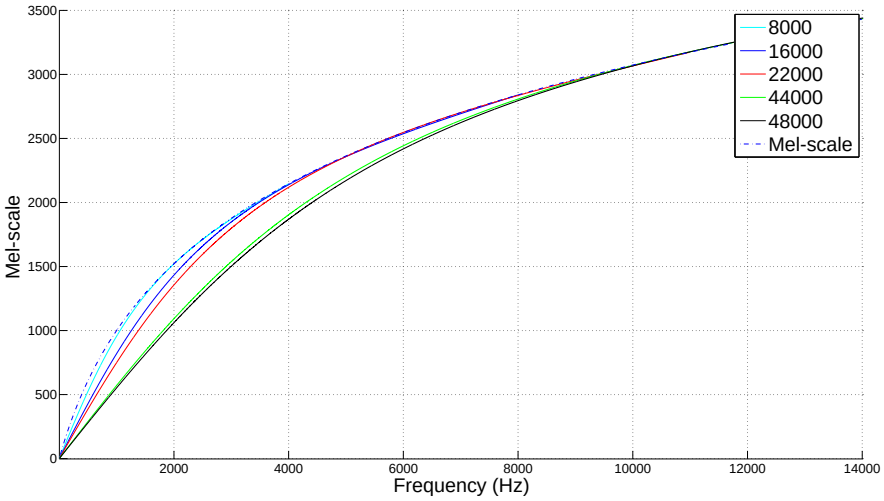


Figure 3.9: Frequency warping approximation to Mel-frequency scale for each sampling frequency (shown in the legend box).

Sampling rate [Hz]	α	**Root mean square error [Hz]**
8k	0.34	35.35
16k	0.42	77.75
22k	0.46	97.96
44k	0.53	164.35
48k	0.53	171.10

Table 3.2: Frequency warping values for different sampling rates.

and the high number of coefficients used with STRAIGHT guarantees a good modelling of this spectrum envelope.

- As the number of coefficients modelled by HMMs increases, the training of the models becomes less robust and therefore the estimation becomes less reliable.

Mixed excitation parameters also need to be adapted to different sampling rates. The following proposal is for the excitation parameters based on frequency bands (in other words, either voicing strengths or aperiodicity are used based on a VSME or an APME approach, respectively). Evidently, the higher the sampling rate, the more frequencies are under analysis so the number of modelled bands must increase as well. Since a finite number of bands can be used and in higher bands speech is perceptually less important (i.e., these bands are more noisy although it is still important to model them for high quality models because is where part of the speech brightness comes from), N bands have been added every 2000 Hz up to half the sampling rate. In particular, this yields to 2 extra bands for a sample rate of 22kHz and 4 extra bands for a sample rate of 44kHz with respect to a sample rate of 16kHz (see Table 3.3 for details).

Sampling rate [Hz]	**Bands [Hz]**
16k	0-1000, 1000-2000, 2000-4000, 4000-6000, 6000-8000
22k	Bands{16k}, 8000-10000, 10000-11025
44k	Bands{22k}, 11025-13000, 13000-16000, 16000-18000, 18000-22050

Table 3.3: Mixed excitation bands for different sampling rates.

3.7 Conclusions

In this chapter, data parameterization for HMM-based TTS systems has been described. Basically, there are three types of parameters to be modelled in a source-filter model approach:

1. **Vocal-tract**. This information is extracted from the spectral envelope to model the speaker identity. Unlike ASR systems where these coefficients must have good generalization characteristics to increase the recognition accuracy, HMM synthesis needs parameters to perform natural speech reconstruction. However, due to the statistical process, they must also have good quantization properties.

   In this work, mel-cepstral coefficients are used (see experiments in Chapter 5) extracted from STRAIGHT in order to increase the accuracy of the spectral envelope's representation.

2. **Fundamental frequency**. F0 is modelled in logarithmic domain using a single estimation for each frame. Two modelling techniques for HMM-based TTS systems have been proposed

in the literature. The main difference is the way that voiced and unvoiced frames are modelled (i.e., independently or simultaneously).

This work uniquely uses the MSD approach, that is, independently modelling unvoiced frames. Once HMM-based F0 parameters are generated, the proposed F0 enhancing technique based on the external CBR-based F0 estimator can be applied (see Section 2.6.3).

3. **Mixed excitation**. Mixed excitation is designed in order to increase the quality of the basic excitation model (i.e., pulses and noise). There are four types of excitations briefly summarized in Table 3.4. Changes to the multiband excitation have been proposed and described in Section 3.5.3 (defined as VSME) where voicing strengths and complex amplitudes are statistically modelled. Nevertheless, experiments have been conducted in this work with both VSME and APME approaches.

Excitation	**Type**	**HMM coefficients**
Pulses and noise	Basic	F0
VSME	Multiband	F0, B voicing strengths (one for each subband) and K_s complex magnitudes (stored as magnitude and phase)
APME	Multiband	F0, averaged aperiodicity (B, one for each subband)
State-dependent filters	Trainable	F0, voiced and unvoiced filter's responses (p_v and p_u, order of the filters)
Pitch-synchronous residual	Residual Codebook	F0, resample and normalized residual coefficients (N_c)

Table 3.4: Summary of possible excitation models described in Section 3.5.

Chapter 4

Hybrid speech synthesis systems

In this chapter, a recent type of speech synthesis technique is described. As we have introduced in Section 1.2, two main trends are used nowadays for speech synthesis: statistical based on HMMs and concatenative systems. In principle, each one of them are focused on different types of applications depending on the target domain, quality and performance of the platform. However, combining both approaches could produce a more reliable synthesis where advantages of both systems would be emphasized. Hence, in this chapter, a hybrid system is defined as *a system where HMM and concatenative based synthesis approaches work in a single framework*. Throughout this chapter, state-of-art of hybrid systems is reviewed classifying different approaches into concatenation-driven and HMM-driven hybrids. Firstly, a concatenation-driven system is described in Section 4.2.1. Then Section 4.3 proposes a novel hybrid approach which uses HMM as the speech production technique in order to maintain the overall spectral smoothness and, also, to take advantage of the possibility of using techniques such as adaptation.

4.1 Introduction

Current state-of-the-art in TTS systems often produce intelligible and very natural speech. The most popular approaches are the concatenative and the statistical synthesis (Black et al., 2007) (see Section 1.2 and Chapter 2).

Concatenative speech synthesis system is based on the selection and concatenation of recorded units. It is the most widely used approach because it can produce high-quality speech. However, its drawback is that the quality can degrade if, for some reason (e.g., data sparsity), an incorrect joint is produced. Therefore, Co-TTS systems are preferred in terms of naturalness and expressiveness and are specially used in limited domain applications where concatenation discontinuities are under control and expressiveness is one of the main goals.

HMM-based TTS systems, on the other hand, are based on generating parameters from a set of trained models. This type of system generates a smoother synthesis which overcomes some of the problems of the concatenative approach. However, the resulting speech quality is lower due to the vocoder effect. Nevertheless, statistical approach is a very attractive solution because it offers a more flexible system (e.g., model manipulation techniques such as speaker adaptation). Moreover, its high level of intelligibility and spectral transition stability make this kind of system very convenient for open domain applications.

Although both approaches seem to use completely different technologies, the possibility of creating a single framework combining both systems will allow to build a more robust system with a new range of possibilities. The three main differences between concatenative and statistical approaches are model topology, acoustic representation and building process (Taylor, 2006). These differences are based on the fact that usually, Co-TTS systems select and concatenate natural segments (e.g. diphones (Hunt and Black, 1996)) although they can also use encoding techniques such as Linear Predictive Coding (LPC) (Hunt et al., 1989) or the Harmonic plus Noise Model (HNM) (Stylianou, 2001). Unlike Co-TTS systems, statistical synthesis usually reconstructs each unit from a sequence of parameters generated from an HMM (see Section 2.5.1). The type of unit used to build the synthetic speech constraints the model topology. In particular, Co-TTS systems use natural segments and HMM statistically models speech in a sequence of states and Gaussian distributions. Obviously, depending on the type of model, a different building process is used. On the one hand, Co-TTS systems are usually built as a database of natural units whereas one of the main characteristics of HMM-based TTS systems is to be automatically trainable.

One of the first proposed hybrid systems was the IBM TTS (Donovan and Woodland, 1999). In this approach, the system uses a set of decision tree-based clustered states to automatically generate a leaf level segmentation of a large database. During synthesis, the phone sequence to be synthesized is converted to an acoustic leaf sequence by descending the HMM decision trees. To determine the segment sequence to concatenate, a dynamic programming search is performed over all the waveform segments aligned to each leaf during the training stage. As we will see in the following sections, subsequent hybrid systems will share part of this structure, hence its importance.

Recently, hybrid systems are receiving a lot of attention and, in fact, efforts are being focused on considering this new approach as an alternative to the Co-TTS and HMM-based TTS systems. As explained, a hybrid system attempts to combine the benefits of concatenative and statistical synthesis. These hybrid systems can be classified in:

1. **Concatenation-driven hybrid systems**. This type of hybrid system uses HMMs as a support to the concatenative process. Basically, synthesis is performed by concatenating natural units using some sort of target units pre-selection by means of a decision-tree based clustering (as explained in Section 2.8). Other approaches described in Section 4.2 can be more sophisticated (e.g., mixing units).

2. **HMM-driven hybrid systems**. Unlike the previous category, this type of system uses an

HMM parameter generation algorithm and the concatenative synthesis is used to enhance the quality of the resulting sequence of parameters. It is based on the idea that a natural unit is the best candidate for a synthesis unit except in the concatenation points, where it is likely to be a quality degradation due to the concatenation process (e.g., audible artifacts). Intensity of the degradation could be estimated by the target and joint costs.

So far, existing hybrid systems in the literature belong to the first type, that is, concatenation-driven hybrid system. The main reason is that this type of hybrid is more likely to guarantee the naturalness of the synthetic speech. Basically, the main advantage of this hybrid over conventional Co-TTS systems is the reduction of undesirable effects due to data sparsity and the decrease of corpus footprint. In addition, it also allows an improvement of the search algorithm's performance. However, in spite of these advantages and the fact that concatenation problems can be alleviated, bad joins can still persist. Moreover, those approaches also lose the advantages of a pure statistical approach such as spectral transition stability or the use of speaker adaptation techniques. For these reasons, a novel hybrid approach based on the second type of hybrids (i.e., HMM-driven) is presented in this work and described in Section 4.3. This proposed system is an HMM-driven hybrid designed with the following purposes:

- Enhance the baseline naturalness of the HMM system in order to make it closer to the quality of the natural units.
- Maintain a low memory footprint.
- Guarantee the possibility of applying HMM techniques such as parameter adaptation (e.g., emotions shown in experiment of Section 5.4.1).

4.2 Concatenation-driven hybrid system

As it has been described, these systems are characterized by using a concatenative approach as the speech production system. As a consequence, natural units are used to produce the synthetic signal whereas HMMs are relegated to a second place. Several types of concatenation-driven approaches have been described in the literature (Zen, 2008; Zen et al., 2009):

- **Target prediction**. Parameters generated from the HMMs are used as a reference to drive the unit selection process. Depending on how HMMs are used, the following approaches have been presented:
  - (Kawai et al., 2004) presented a Co-TTS system using HMMs to constraint the prosodic parameters of the target units. In particular, HMM generates F0, durations and power of phone-based units which are then used by the unit selection algorithm.

- (Hirai et al., 2007) proposed to use HMMs to build an acoustic target vector to select 5 ms segments. Similarly, (Rouibia and Rosec, 2005) used the HMM framework as an acoustic target which is then explicitly used in a cost function during the selection process.
- A composite target and join cost system is defined by (Ling et al., 2007) using HMMs as part of the unit selection algorithm. This system is described in Section 4.2.1 since a simplified version of it is a fundamental part of the proposed hybrid system in Section 4.3.
- A Minimum Selection Error (MSE) criterion was proposed by (Ling and Wang, 2008) in order to select the optimal phone-size unit sequence from the speech database by maximizing the combined likelihood of a group of HMMs. The idea is to estimate the HMM parameters in order to minimize the number of different units between the selected and natural phone sequences for the training sentences. This optimization is performed using a generalized probabilistic descent algorithm. In other words, they introduce an objective criterion into model training that is able to evaluate the overall performance of a unit selection system on the training set. The simplest criterion is to evaluate the synthesized speech by counting how may phones in the selected unit sequence are different from the natural sequence when synthesizing a sentence in the training database. Then model weights and parameters are estimated to minimise such unit selection error.

- **Unit Smoothing**. HMMs are used to alleviate the quality degradation that is due to the use of traditional spectral techniques for smoothing the joins between units (Dutoit, 1994).

  1. Probabilistic smoothing. As proposed by (Plumpe et al., 1998), HMMs can be used to smooth the spectrum according to what was observed at the junctions of real speech during training while retaining a filter calculated from an actual speech utterance.
  2. Unit fusion. As described by (Wouters and Macon, 2000), the proposed idea is to fuse units by using an extension of a cost function. Synthesis is performed by selecting two distinct types of speech units: concatenation and fusion units. The latter characterizes the spectral dynamics at the joint points between concatenated units. The signal is regenerated using a sinusoidal all-pole speech representation. The fusion process uses a weight function for each frame that reaches "1" at each concatenation point (which corresponds to the center of a fusion unit) and it reaches "0" at the boundaries of the fusion unit (which is a point within the concatenation unit).

- **Unit Mixing**. Approaches within this category alternatively use natural units and sequences generated by HMMs. In this context, mixing refers to the process of concatenating both types of signals without actually blending one with each other. Two approaches have been proposed:

  1. A multiform segment algorithm (Pollet and Breen, 2008) determines the optimal sequence of segments (i.e., natural or HMM-generated units) by minimising the degradation of speech.

2. A hybrid voice conversion (Okubo et al., 2006) attempted to employ a conversion method in order to combine unit selection with spectrum generated by speaker-adapted HMMs. In case the required phoneme context is missing (or the concatenation is likely to produce a discontinuity), a statistical adapted vocal-tract sequence is concatenated.

- **Unifying approaches**. In order to merge both synthesis approaches, a frame-based sequence probability algorithm was presented by (Taylor, 2006). The idea behind it is to develop a purely statistical join probability (similar to the join cost used in the Co-TTS). The proposed join model seeks to provide a genuine probability that one section of speech will follow another. In doing this, all state models are joined in an ergodic fashion. Unlike the normal way of calculating the acoustic join cost, this approach studies natural occurring sequences and uses those as a model for what should constitute a good join. The probabilistic formulation is based on calculating the joint probability that a sequence of frames may occur.

4.2.1 An HMM-based unit selection and waveform concatenation

In this section we introduce a concatenation-based hybrid approach that is based on target prediction as proposed by (Ling et al., 2007). We describe this approach in detail here because it is part of the novel hybrid system that we propose in Section 4.3.

According to the previous classification, the HMM module in this system guides the selection of phone-sized candidate units. In fact, it uses the HMM emission probabilities trained using the ML criterion as target and joint costs. Hence, the system is described as an ML-based unit selection TTS system. The optimal phone sequence is expected to be selected from the speech database to maximize the combined likelihood of the acoustic model, phone duration model and concatenation model.

For a phoneme within an utterance ($n \in [1, N]$), the context-dependent acoustic model, phone duration model and concatenation model determined by clustered HMMs and decision trees are λ_n^m, λ_n^d and λ_n^c, respectively. For a candidate unit u_n the associated acoustic feature vector consists of static and dynamic features (see Section 3.2) of length d_n:

$$\begin{aligned} \mathbf{u} &= \{u_1, u_2, \ldots, u_N\} \\ \mathbf{o}(u_n) = \mathbf{o}_n &= \{\mathbf{o}_{n,1}, \mathbf{o}_{n,2}, \ldots, \mathbf{o}_{n,d_n}\} \end{aligned}$$

where $\mathbf{u}$ is the complete sequence of candidate units for an utterance and $\mathbf{o}_n$ are the acoustic feature vectors which consist of static and dynamic features for each frame of the natural speech. For a whole utterance, the corresponding target models are:

$$\lambda^x = \{\lambda_1^x, \lambda_2^x, \ldots, \lambda_N^x\} \text{ where } x \in \{m, d, c\}$$

The goal is to find an optimal $\mathbf{u}^*$ that maximizes the following sum of log likelihoods:

$$\mathbf{u}^* = \arg\max_{\mathbf{u}} \left[\mathcal{L}_m(\mathbf{u}, \lambda^m) + \mathcal{L}_d(\mathbf{u}, \lambda^d) + \mathcal{L}_c(\mathbf{u}, \lambda^c)\right] \tag{4.1}$$

where $\mathcal{L}_x$ measures the log likelihood of the candidate unit sequence $\mathbf{u}$ for $x = \{m, d, c\}$ as follows:

$$\mathcal{L}_m(\mathbf{u}, \lambda^m) = \sum_{n=1}^{N} \log P\left(\mathbf{o}_n | \mathbf{Q}_n, \lambda_n^m\right) \tag{4.2}$$

$$\mathcal{L}_d(\mathbf{u}, \lambda^d) = \sum_{n=1}^{N} \log P\left(d_n | \lambda_n^d\right) \tag{4.3}$$

$$\mathcal{L}_c(\mathbf{u}, \lambda^c) = \sum_{n=2}^{N} \log P\left(\mathbf{o}_{n,1} - \mathbf{o}_{n-1,d_{n-1}} | \lambda_n^c\right) \tag{4.4}$$

where $\mathbf{Q}_n$ is the fixed-state sequence.

In order to facilitate the search process, the traditional target cost and join cost equation (see Equation 1.1) was proposed by (Ling et al., 2007) [1]:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \left(\sum_{n=1}^{N} C_t(u_n) + \sum_{n=2}^{N} C_c(u_{n-1}, u_n) \right) \tag{4.5}$$

where:

$$C_t(u_n) = w_m \frac{d_n^p}{d_n} \sum_{j=2}^{d_n - 1} \mathrm{M_d}\left(\mathbf{o}_{n,j}, \boldsymbol{\mu}_{n,j}^m, \mathbf{U}_{n,j}^m\right) + w_d \mathrm{M_d}\left(d_n, \mu_n^d, \sigma_n^{d2}\right) \tag{4.6}$$

$$C_c(u_{n-1}, u_n) = w_c \frac{d_n^p}{d_n} \mathrm{M_d}\left(\mathbf{o}_{n,1} - \mathbf{o}_{n-1,d_{n-1}}, \boldsymbol{\mu}_n^c, \mathbf{U}_n^c\right) \tag{4.7}$$

being:

- $\mathrm{M_d}(\mathbf{c}, \boldsymbol{\mu}, \mathbf{U})$ is the Mahalanobis distance between vector $\mathbf{c}$ and the corresponding distribution parameters with mean $\boldsymbol{\mu}$ and variance $\mathbf{U}$ defined as,

$$\mathrm{M_d}(\mathbf{c}, \boldsymbol{\mu}, \mathbf{U}) = (\mathbf{c} - \boldsymbol{\mu})^{\mathrm{T}} \mathbf{U}^{-1} (\mathbf{c} - \boldsymbol{\mu})$$

- d_n and d_n^p are the candidate and predicted phone durations, respectively. The latter is estimated by the duration model λ_n^d. These durations are used for target cost normalization.
- $\boldsymbol{\mu}_{n,j}^m$ and $\mathbf{U}_{n,j}^m$ are the mean vector and covariance matrix for the observation Gaussian probability density function for frame j and candidate unit u_n.
- w_x for $x = \{m, d, c\}$ are cost weights for the acoustic, duration and concatenation models, respectively.

[1] Note that unlike Equation 1.1, in this case there is no feature vector $\mathbf{f}$ because this information is already present in the HMMs.

- $\boldsymbol{\mu}_n^c$ and $\mathbf{U}_n^c$ are the mean vector and covariance matrix of the concatenation model, respectively.
- μ_n^d and σ_n^{d2} are the mean and the variance of the duration model, respectively.

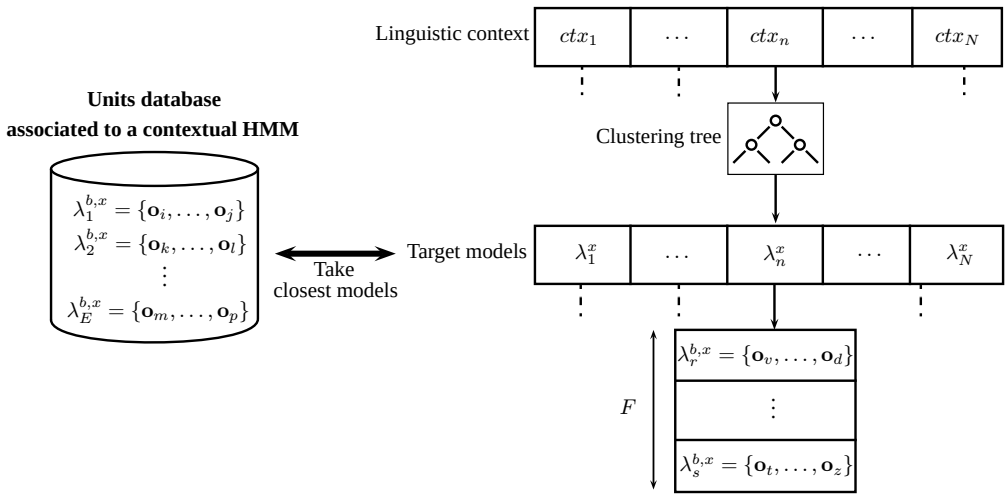


Figure 4.1: Synthesis process of the HMM-based unit selection TTS system. A context ctx_n for each unit is obtained from the input text. A target model λ_n^x is obtained by clustering this context. Then, F models are picked up from the database and the associated observations are used to compute the target and join costs.

A dynamic programming search can be used in order to find the optimal sequence of units in Equations 4.5, 4.6 and 4.7. The process (depicted in Figure 4.1) is as follows:

- As depicted in Figure 4.2, the database can be defined in two different ways. On the left side, attention is focused on the U units. Each unit ($u_l = \{\lambda_i^{b,x}, \mathbf{o}_l\}$) refers to its observation vector $\mathbf{o}_l$ and to a contextual model ($\lambda_i^{b,x}$). Mapping between units and a corresponding HMM is given during alignment in the phoneme segmentation stage. On the right side, the same database is expressed in terms of the E contextual models. In this case, each of these models ($\lambda_i^{b,x}$) represents several observation vectors.
- Given an utterance, a target model is associated with its contextual phoneme ($ctx_n \rightarrow \lambda_n^x$). This is obtained by clustering the linguistic contexts with a decision tree (see Section 2.8).
- Using the target model λ_n^x for unit u_n, the F closest models are picked up from the database ($\{\lambda_r^{b,x}, \ldots, \lambda_s^{b,x}\}$). All observations associated with all the models ($\{\mathbf{o}_v, \ldots, \mathbf{o}_z\}$) are then used to calculate the target and join costs using Equations 4.6 and 4.7. The purpose is to have several observations where to compute the target and join costs.

  As it can be seen, for each unit u_n there are F models. Doing this for all the models in the database can result in a very computational expensive process and thus $F < E$. The closest F models are selected by a Kullback-Leibler (KL) divergence based unit pre-selection algorithm as

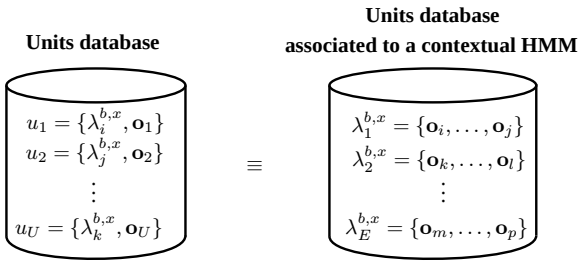


Figure 4.2: Two representation of the same database of units. On the left, the database is defined for each unit. There are U units in total and unit u_i contains its observation vector $\mathbf{o}_i$ and an associated contextual model $\lambda_k^{b,x}$. On the right, the same database is described with respect to E contextual models. In this case, it is clear that each contextual model is a one-to-many mapping, so for each model there are a set of observation vectors.

proposed by (Ling et al., 2007). This divergence (see Section C.2) would be computed between the HMM of the target unit (λ_n^x) and the HMM of each candidate unit ($\{\lambda_r^{b,x}, \ldots, \lambda_s^{b,x}\}$) to select the F-best units with minimum KL divergence before the final minimization process of the total cost.

Because the state observations of all context-dependent HMMs are clustered using decision trees, the KL divergence presented in Equation C.4 can be calculated offline (i.e., before synthesis) as a symmetric matrix (Equation 4.8) for every two leaf nodes in the decision tree of each state. Therefore the unit pre-selection step can be realized efficiently. Assuming that state j has J leaf nodes, its corresponding KL divergence matrix is:

$$\mathbf{D}^{(j)} = \begin{bmatrix} 0 & \mathrm{D}\left(\lambda_1^{(j)}, \lambda_2^{(j)}\right) & \mathrm{D}\left(\lambda_1^{(j)}, \lambda_3^{(j)}\right) & \ldots & \mathrm{D}\left(\lambda_1^{(j)}, \lambda_J^{(j)}\right) \\ \mathrm{D}\left(\lambda_2^{(j)}, \lambda_1^{(j)}\right) & 0 & \mathrm{D}\left(\lambda_2^{(j)}, \lambda_3^{(j)}\right) & \ldots & \mathrm{D}\left(\lambda_2^{(j)}, \lambda_J^{(j)}\right) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathrm{D}\left(\lambda_J^{(j)}, \lambda_1^{(j)}\right) & \mathrm{D}\left(\lambda_J^{(j)}, \lambda_2^{(j)}\right) & \mathrm{D}\left(\lambda_J^{(j)}, \lambda_3^{(j)}\right) & \ldots & 0 \end{bmatrix} \tag{4.8}$$

where $\lambda_k^{(j)}$ is the model associated with leaf node k and state j. This matrix is symmetric if a symmetric KL divergence is assumed (see Section C.1).

- Dynamic programming search can be applied to solve Equation 4.5 (Ling et al., 2007).

4.3 HMM-driven hybrid system

In the following section, a new hybrid system is proposed in order to develop the last of the objectives of this thesis. As it will be seen, this type of hybrid proposes an enhancing algorithm and a weighting function in order to blend HMM parameters with natural units.

4.3.1 Introduction

In this work, a proposed hybrid alternative based on the second category of hybrid systems (i.e., hybrid HMM-driven) is proposed taking into account the following constraints (Gonzalvo et al., 2009a):

1. The synthesis process should avoid mixing units as this can result in quality degradation due to their different spectral nature. Unlike other approaches described in Section 4.2, the idea now is to blend segments rather than concatenating them. In particular, segments are natural units and sequences produced by HMMs.

2. The concatenative module should be used to improve the quality in stable regions. The closest approach found in the literature refers to unit fusion (Wouters and Macon, 2000). In our approach we do not just intend to smooth the concatenation point but also to enhance the quality of the whole utterance. However, taking into account the unit fusion approach, it is interesting to incorporate a way to emphasize stable parts of natural units. This is achieved by introducing a weight function $w(f)$ which controls the contribution of the concatenation segments (i.e., the intensity of the blending) for each frame f (see Section 4.3.6).

This HMM-driven hybrid speech synthesis system sacrifices part of the quality of the natural units. However, it creates a more flexible structure able to improve the quality of pure state-of-the-art HMM systems while keeping their main advantages (e.g., spectral stability and adaptation techniques). By the definition of these constraints, the resulting system cannot be described as a unit selection smoother because synthetic speech can just become a vocoded re-synthesized version of the concatenative system. In other words, if we set the weighting function to minimum enhancement (i.e., no blending of natural and HMM sequences), then, only the HMM parameter sequences are generated, and in consequence, the original HMM synthesis can be retrieved. Otherwise, by setting the weight function to maximum enhancement, it is not possible to recover the original natural units but a vocoded version of them.

4.3.2 System description

The following hybrid system enhances the naturalness of the HMM synthesis using natural segments obtained form a Co-TTS system. This is performed by updating the HMM's parameters used during the synthesis process of a sentence. This approach is similar to adapting the HMMs for a sentence in order to obtain the closest HMM that represents the natural reference best. As a consequence, HMMs are reset for each sentence in order to guarantee that statistical models are not overfit to previous observations. Moreover, a weight function is used to control the intensity of the enhancement and to smooth possible concatenation errors.

Similar to other TTS systems, the proposed approach consists of training and synthesis stages shown in Figure 4.3. As it can seen, the system is composed of two types of phone-based modules:

1. HMM-specific component (in blue in the figure). It generates sequences of parameters with the conventional parameter generation algorithm described in Section 2.5.1. In order to provide the system with the feature of efficient voice adaptability this system uses speaker-independent training (target speaker data is used for adapting the multi-speaker voice). The standard procedure used in this context is described in Sections 2.3 and 2.7.3 and the adaptation technique used in this system is in Section 2.7.1.

2. Concatenative module (in green in the figure). This part performs a selection of natural units and it is based on a simplified version of the concatenation-driven hybrid system described in Section 4.2.1. Further details are described in Section 4.3.4.

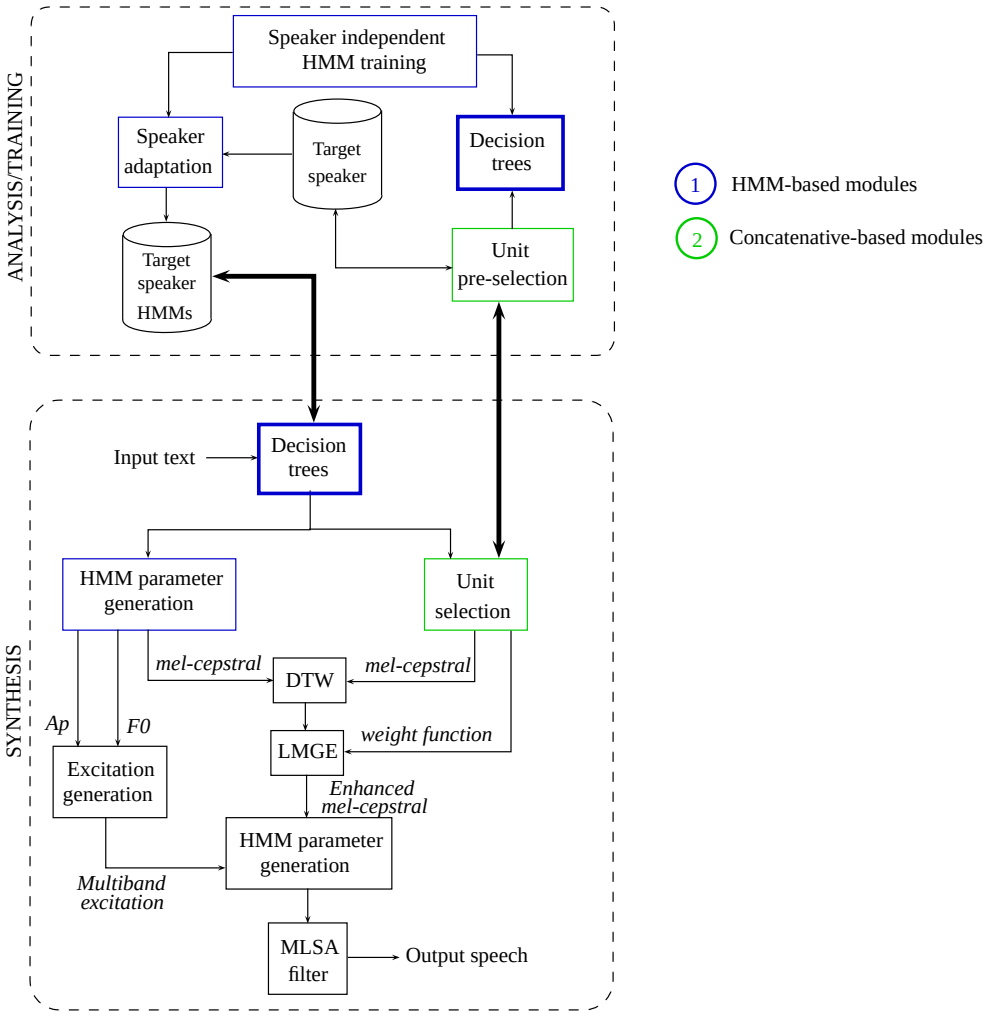


Figure 4.3: An overview of the HMM-driven hybrid TTS system.

According to these two types of modules, the training stage consists of two steps. Firstly, once a

speaker-independent HMM voice has been trained (see Section 4.3.3), these models are adapted to the target speaker. Using the speaker independent trees, the second step builds the concatenative representation. This essentially constructs an HMM-based unit selection system which uses decision trees to pre-select groups of units. Note that as both modules use the same decision tree-based clustering, low spectral distortion is expected.

During synthesis time, the goal of the concatenative module is to produce the best sequence of natural units that will be used to enhance the vocal-tract parameters from the HMM. First, the text to be synthesized is converted to its linguistical context. Decision trees are then used to select the corresponding HMM units. Durations are estimated from the HMMs and are used to generate the parameters of vocal tract, F0 and excitation. Natural units are then selected by the concatenative module based on the pre-selection of units made during the training stage (see Section 4.3.4 for more details).

The blending process starts by generating a weight function taking into account the concatenation boundaries. In order to guarantee an optimal synchronization between vocal tract sequences of the two modules, mel-cepstral information extracted from the concatenative segments and the one generated by the HMM component are aligned using a Dynamic Time Warping (DTW) algorithm (Turetsky and Ellis, 2003). Once the concatenative information have been properly aligned to the HMM sequence, means and variances of the current HMM units are updated using the Local Minimum Generation Error (LMGE) criterion (described in more detail in Section 4.3.5) based on the aligned mel-cepstral sequence of natural units and the weight function (see more details in Section 4.3.6).

Finally, synthetic speech is generated from the enhanced HMM parameter sequence which generate a new sequence of mel-cepstral coefficients. Ultimately, MLSA filter reconstructs the speech signal by filtering the aperiodicity multiband excitation (this excitation is described in Section 3.5.4). Next sections describe the above steps in more detail.

4.3.3 Speaker-independent HMM training

Consider the following situation: we want to build a high-quality TTS system with only a few samples of the target speaker. In that case, and considering the two existing synthesis trends (i.e., Co-TTS-based and HMM-based TTS systems), there are three possibilities (see Section 1.2.1 for a general discussion of synthesis approaches):

- Build an HMM system. Although the amount of data necessary to reach a reasonable quality is less than the data needed to build a stable Co-TTS, in some cases it could still be not good enough.
- Build an adapted HMM system. The adaptation process is attractive because it does not require a huge amount of data and reliable quality can be obtained even with a few adaptation utterances. By using a speaker-independent training and adapting, the resulting system has

enough data to train robust models (as described in the experiment of Section 5.3.4, a total of 8 speakers are used). This system still has the conventional advantages and disadvantages we have been discussing throughout this thesis (e.g., vocoded quality, over-smoothing or spectral stability).

- Build a Co-TTS system. The size of the target data is likely to be insufficient to select stable unit sequences. Therefore, this is not a good option unless the TTS is for a limited domain application.

- Build a Co-TTS system with a voice conversion module (Erro, 2008). This approach could alleviate some of the problems of the previous speaker-dependent Co-TTS system reaching a better quality than the HMM-based adapted system. However, the main drawback for that approach is the stability of the Co-TTS system and the necessity of a high-quality conversion. In that respect, HMM speaker adaptation has been demonstrated to be a very efficient and robust method to obtain a very reliable transformation to a target speaker (Yamagishi et al., 2009).

In such a situation, the proposed hybrid system introduces a set of interesting features. First, it is possible to apply speaker adaptation techniques and second, the quality of the system will be at least similar or higher than that for the conventional HMM system since a speaker-independent training is used (see experiment in Section 5.4.2). For these reasons, speaker adaptation is applied in this hybrid system. Note that the size of the target speaker corpus is a key aspect of the design of this hybrid system. Firstly, the smaller the corpus is, the more concatenation can arise from the concatenative module. In addition, the performance of the adaptation can also degrade the synthetic quality. In that respect, the weighting function described in Section 4.3.6 can help to alleviate the bad joint problem. In case the hybrid system has not enough data to produce a reliable concatenation sequence, the weight can be set to favour the HMM system.

In our system, an average voice model is trained using a set of different speakers in order to obtain a robust voice with only some samples of the target speaker. Model training for HMM-based synthesis constructs a set of context-dependent HMM models where the vocal tract, pitch and excitation parameters are simultaneously modelled.

A standard mel-cepstral HMM system (Zen et al., 2007a) is employed as the the core of the HMM module using the following start-of-the-art quality improvements. First, the high-quality speech vocoder STRAIGHT is applied to analyze the spectral envelope using 39-th order mel-cepstral coefficients (see Section 3.3.5). Also, aperiodicity (Kawahara et al., 2001) is used for the multiband excitation model to reduce the buziness of the vocoder (see Section 3.5.4 for more details). The aperiodicity-based mixed excitation uses 5 subbands (0-1, 1-2, 2-4, 4-6 and 6-8 kHz) as the sample rate is 16k Hz (see Section 3.6 for other sample rate configurations).

In order to alleviate the over-smoothing effect, GV (Toda and Tokuda, 2007) is applied (see Section 2.6.1). In this particular system, explicit state duration probability density function (HSMM)

(Zen et al., 2007b) has been employed (see also Section 2.4.3). Pitch information (represented in logarithmic domain) is modelled by an MSD (see F0 modelling in Section 3.4). Decision trees are constructed using MDL criterion as described in Section 2.8.2.2.

Adaptation uses the CMLLR technique where mean and covariance matrices are obtained by simultaneously transforming all the parameters. In addition a MAP transformation (Yamagishi and Kobayashi, 2007) is also used. Both techniques have been detailed in Sections 2.7.1 and 2.7.2, respectively.

4.3.4 Concatenative system

The aim of the concatenative module is to provide the best natural units to the hybrid system in order to improve the HMM-generated parameters. The method employed here is based on a simplified version of the HMM-based unit selection system described in Section 4.2.1. The simplification is twofold: it only uses acoustic models (λ_n^m) for the target cost computation; this cost is only computed in a unit-basis rather than in a frame-basis, therefore the unit selection algorithm uses a conventional Manhattan distance-based approach.

According to Figure 4.3 (green blocks) this module consists of:

- **A pre-selection of natural units**. A cross-distance matrix is computed between all leaf nodes of the decision trees as in Equation 4.8 using the KL divergence. By doing this, the search space of the unit selection algorithm is reduced. However, unlike the Co-TTS hybrid in Section 4.2.1, here the system does not use a distance matrix $\mathbf{D}_i$ for each state i but a unique matrix $\mathbf{D}$ where all mixtures from all states are grouped together. This simplification reduces the computational cost and lets the system compute the target cost for each unit rather than in a frame-basis.
- **A unit selection algorithm**. A classical unit selection search is carried out to get the best acoustic joins. The search is carried out over the set of pre-selected units using the Manhattan distance, a set of conventional linguistic features $\mathbf{F}$ and Equation 1.1 (Taylor, 2009).

4.3.5 Local Minimum Generation Error

Minimum Generation Error (MGE) criterion was first introduced by (Wu and Wang, 2006) and has been described in Section 2.6.2. As it has been seen, the idea is to minimize the error of the HMM-generated parameters with respect to the original training data. This is achieved by post-processing of the previously trained models using the standard ML criterion and a distance definition. The consequence is that the mean and the variance of each mixture are enhanced becoming a better representation of the natural data.

Rather than applying MGE to a set of multiple observations, we propose an alternative application of MGE, defined as Local Minimum Generation Error (LMGE) criterion. This approach is

essentially an MGE applied to a single target utterance during the synthesis process, rather than to multiple sentences during the training stage. By using LMGE during synthesis time with an optimal single sequence of natural units generated by the concatenative module, it is possible to update the HMMs used by the current sentence and adapt their parameters to the stable part of the natural segment. In order to guarantee that concatenation errors are smoothed, a weight function $w(f)$ is introduced into the LMGE algorithm for each frame f (see Section 4.3.6).

Firstly, it is necessary to measure the distortion between the concatenative-based parameterized speech signal ($\mathbf{C}$) and the HMM-based generated parameter ($\hat{\mathbf{C}}$) vectors. The similarity metric we adopt is the squared Euclidean distance:

$$D_c\left(\hat{\mathbf{C}}, \mathbf{C}\right) = \left\|\hat{\mathbf{C}} - \mathbf{C}\right\|^2 \tag{4.9}$$

LMGE is based on the simplified version of MGE because the process is conducted during synthesis time. As described in Section 2.6.2 and Appendix B, MGE is a very computationally expensive algorithm because it has to perform the inversion of matrix $\mathbf{R}$ defined in the HMM parameter algorithm (see Section 2.5.1). The simplified version of MGE (described in Sections 2.6.2.2 and B.3) does not need to invert this matrix and therefore it is more suitable for synthesis purposes.

According to the simplification of Equation B.11, Equations 2.71 and 2.72 can be obtained by using all the training samples of the concatenated-based sequence at the same time and the updating rule of Equation 2.70. In this case, unlike the original reduced MGE, a single utterance is used to update the HMM models. In consequence, the LMGE sets $N = 1$ in Equations B.13 and B.16. In addition, we expand the original definition to update the mean and the variance of the corresponding HMM including the weight function $w(f)$ as follows:

$$\mu_{ij}^{(new)} = \mu_{ij}^{(old)} - \frac{1}{N_i}\sum_{f=1}^{F} \bar{w}(t)\varphi_i(t)D_f(j) \tag{4.10}$$

$$v_{ij}^{(new)} = v_{ij}^{(old)} - \frac{\sigma_{ij}^{2(old)}}{N_i}\sum_{F=1}^{F} \bar{w}(t)\varphi_i(t)D_f(j)\left(\mu_{ij}^{(old)} - o_t(j)\right) \tag{4.11}$$

where:

- $f \in [1, F]$ is the frame being analyzed.
- $\varphi_i(t) = \begin{cases} 1 & \text{if distribution } i \text{ is visited at frame } t \\ 0 & \text{otherwise} \end{cases}$
- j is the order of the coefficient of the multivariate Gaussian distribution.
- $D_f(j) = (\hat{o}_f(j) - o_f(j))$ is the actual error between the generated and the natural observations, respectively.

- N_i is the total number of samples in model i. Note that, usually, some mixtures are updated using samples from various frames (e.g., the duration of a mixture is usually longer than one frame). In this case, N_i works as a normalization factor (see Equation B.17).

- $\mu_{ij}^{(old)}$ and $v_{ij}^{(old)}$ are the original mean and inverse variance prior to update.

- $\bar{w}(t) = 1 - w(t)$ is the inverted weight for frame t. It controls the intensity of the blending process. The purpose of this function is to weight different regions intensifying the LMGE adaptation in stable periods (regions between concatenation points) (see Section 4.3.6).

Note that the use of LMGE does not convert the HMM generated sequence into natural units because the updating process is performed on the mean and variance of the HMM only.

4.3.6 Weight function for region updates

In order to control the relevance of the HMM updating in different regions of the signal, a special weight function has been introduced into the updating Equations 4.10 and 4.11. In general, the weight function could be used for controlling the updating of segments, phonemes, words or even full utterances by selecting each time segment accordingly. In this thesis, however, we introduce a weight function that allows to use a different adaptation intensity for each frame, that is, $w(f) \in [s_L, s_U]$ for frame f under the following conditions:

$$0 \leq s_L < s_U \leq 1$$

$$s_M = \frac{s_L + s_U}{2}$$

The purpose of this function is to weight different regions intensifying the LMGE adaptation in stable periods (regions between concatenation points). Figure 4.4 shows an example of the weight function for a phoneme and its right context. Note that a phoneme is divided into two different parts: stable part and concatenation points. Around concatenation points, the function tends to the maximum value (s_U), so HMM enhancement is relaxed. In the limit, if the maximum value is $S_U = 1$, then the corresponding HMM model is not adapted in this frame ($\mu_{ij}^{(new)} = \mu_{ij}^{(old)}$, $v_{ij}^{(new)} = v_{ij}^{(old)}$) as it is preferable to maintain pure statistically derived parameters. Otherwise, when the weight function takes the minimum value (S_L) around stable regions, LMGE is strongly applied as concatenative measured parameters are considered here as an excellent representation of the speech signal.

As the weight is defined per-frame, transitions between them are smoothed using either a sigmoid representation or a linear interpolation in case the phoneme is too short, and in consequence, the sigmoid does not fulfill the condition to obtain Equation 4.15. Assuming that the weight function is defined for each phone independently, that is, $f \in [0, d_p]$, then the weight function $w(f) \in [s_L, s_U]$

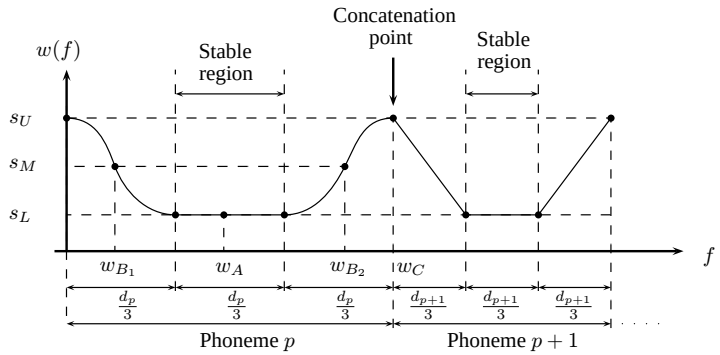


Figure 4.4: Per-frame weight function for phoneme p and phoneme $p+1$ with durations d_p and d_{p+1} frames, respectively. Phoneme p uses the sigmoid-based smoothing function (Equation 4.12) whereas phoneme $p+1$ uses the linear interpolation-based smoothing function (Equation 4.16) as d_{p+1} is too short. During the transition frame between phonemes (e.g., $f = w_C$) the weight tends to be high and the update of the models is not very intense. On the stable regions, (e.g., $f = w_A$) s_L will be small and therefore the model will be updated.

for a phoneme p is defined as

$$w(f) = \begin{cases} \frac{a}{b+\exp\left(f-w_{B_1}\right)} + s_L & f \in \left(0, \frac{d_p}{3}\right) \\ s_L & f \in \left[\frac{d_p}{3}, 2\frac{d_p}{3}\right] \\ \frac{a}{b+\exp\left(-(f-w_{B_2})\right)} + s_L & f \in \left(2\frac{d_p}{3}, d_p\right) \\ s_U & f = \{0, d_p\} \end{cases} \tag{4.12}$$

where d_p is the duration of phoneme p and w_{B_1} and w_{B_2} are the centre of each sigmoid defined as,

$$w_{B_1} = \tfrac{1}{6} d_p \quad f \in \left(0, \tfrac{d_p}{3}\right) \tag{4.13}$$

$$w_{B_2} = \tfrac{5}{6} d_p \quad f \in \left(\tfrac{2d_p}{3}, d_p\right) \tag{4.14}$$

In order to define the sigmoid values between s_L and s_U, a and b must be:

$$\begin{aligned} b &= \frac{s_M - s_L}{s_U - s_M} \\ a &= b \cdot (s_U - s_L) \end{aligned} \tag{4.15}$$

Note that Equation 4.15 is based on the following constraint:

$$w\left(f \rightarrow w_{B_2} + \frac{d_p}{6}\right) \approx \frac{a}{\underbrace{b + \exp\left(-\frac{d_p}{6}\right)}_{\approx 0}} + s_L = s_U$$

The exponential in the above equation converges to zero for ${}^{d_p}/_6 \geq 6$, that is when $d_p \geq 36$ frames. For a frame shift of 5 msec, the minimum duration of a phoneme to produce a valid sigmoid in the concatenation point must be $d_p \geq 180$ msec. When this constraint is not fulfilled, the smoothing-based function is approximated by a linear interpolation and, in consequence, it is defined as,

$$w(f) = \begin{cases} \frac{3(s_L - s_U)}{d_p} f + s_U & f \in \left[0, \frac{d_p}{3}\right] \\ s_L & f \in \left(\frac{d_p}{3}, \frac{2d_p}{3}\right) \\ \frac{3(s_U - s_L)}{d_p}\left(f - \frac{2d_p}{3}\right) + s_L & f \in \left[\frac{2d_p}{3}, d_p\right] \end{cases} \tag{4.16}$$

Figure 4.4 shows an example of the weight function with a phoneme and its right context. On the one hand, phoneme p uses the sigmoid-based smoothing function. In this case, w_A is a point in the stable concatenation region, w_B is the centre of the sigmoid or a smoothed transition frame and w_C is a concatenation point. On the other hand, duration of phoneme $p+1$ is too short so the linear interpolation-based smoothing function is applied.

4.4 Conclusions

In this chapter we have described the state-of-the-art of hybrid systems. We have proposed to extend the current hybrid systems classification to: concatenation-driven and HMM-driven hybrid systems. On the one hand, the former type implies concatenating natural units whereas HMMs basically guide the unit selection process (see Section 4.2). On the other hand, HMM-driven hybrid systems is a proposed new category of hybrid systems (described in Section 4.3). This type of approach uses an HMM system with a concatenative module which helps to increase the naturalness of the generated parameters. The purpose of an HMM-driven hybrid system is to, firstly, take advantage of the HMM system characteristics (e.g., spectral transition smoothness or adaptation) and, then, to improve the naturalness by taking the stable part of natural units.

Within the HMM-driven system type, we have proposed a hybrid system that uses the LMGE algorithm in order to blend HMM parameter sequences with natural units selected with a concatenative module. That concatenative module has being designed as a simplified version of a concatenation-driven hybrid system described Section 4.2.1. The blending process is controlled by a weighting function in order to tune the intensity of the enhancing process and, in addition, to help smoothing the joins.

Chapter 5

Experiments

Experiments have been conducted with the purpose of evaluating the performance of different HMM-based TTS systems. Two types of experiments have been performed. On the one hand, subjective measures use listener's opinions to validate naturalness and expressiveness of synthetic speech. On the other hand, different objective experiments will be also proposed in order to give a mathematical point of view of the subjective opinion.

Three types of corpora has been used: Castilian Spanish, English and emotional Castilian Spanish. Depending on the purpose of the experiment, a different corpus is used. First of all, experiments using Spanish are focused on showing the quality of an HMM-based TTS system using this language. Similarly, improvements on expressiveness, naturalness and emotion adaptation are also validated with the emotional Spanish corpus. Secondly, real scenario applications have been analyzed using English since more data is available for the study. In addition, that is also the language used for validating the proposed hybrid approach.

Although by the end of this thesis an HMM training toolkit has been developed and tested, all experiments in this section have been trained with HTS (HTS, a). For synthesis, the HTS engine tool (HTS, b) was used. The reduced version of STRAIGHT for spectrum and aperiodicity was used (Kawahara, 1999).

This section is organized as follows. Firstly, corpora is described in Section 5.1 and then, four types of experiments are categorized as follows:

1. **Experimental tests** (see experiments in Section 5.2). Firstly, phonemes and diphones are compared and secondly, different sampling rates are evaluated.

2. **Proposed work and baseline improvements** (experiments in Section 5.3). Firstly, the enhancement of the F0 prediction for the Spanish HMM-based TTS system is evaluated in Section 5.3.1. Secondly, the performance of the contextual factors for Spanish are shown in

Section 5.3.2. Mixed excitations are subjectively evaluated in Section 5.3.3. Finally, the hybrid system is analyzed in Section 5.3.4.

3. **Applications** (Section 5.4). Emotion and speaker adaptation experiments are described in Sections 5.4.1 and 5.4.2, respectively.

4. **Overall TTS performance** (see Section 5.5). The overall quality of the Spanish and the English systems are shown in Sections 5.5.1 and 5.5.2, respectively.

Examples can be listened here: http://www.salle.url.edu/~gonzalvo/hmm

5.1 Corpora details

5.1.1 Castilian Spanish corpus

The Spanish female voice (standing PTR) was created from a corpus developed along with LAICOM (Iriondo et al., 2006). Speech was recorded by a professional speaker in neutral emotion and segmented and revised by speech processing researchers. The total length of the corpus is 49 minutes.

The corpus contains 8.3% of exclamative (EXC) sentences, 70.7% declarative (DEC) sentences and 21% interrogative (INT) sentences. All objective experiments utilizing this corpus use 25% of the sentences for testing purposes and 75% for training (in other words, 620 phrases of a total of 833 are used for training). Training sentences were selected accordingly in order to maintain corpus categories' percentages resulting on 8.1% EXC, 70.8% DEC and 21.9% INT. Consequently, from the HMM system point of view, contextual units to be trained represent around 20000 units.

5.1.2 English corpus

English voices were designed both for Co-TTS and HMM-based TTS systems. Therefore, as described in Table 5.1 the average size of a voice is larger than the Spanish corpus. Two English variants have been used, British (UK) and American (US).

The voice of three speakers was recorded. Firstly, DIG is a male speaker with American accent. He is interpreting an action agent and in consequence, his speaking style is slightly more expressive (specifically more aggressive) than a conventional neutral speaker. On the contrary, RJS is a professional British male speaker employing a very neutral and stable voice tone. Both speaking rates are slower and utterances are designed to be longer in comparison with the rest of the voices. This guarantees a continuous constant tone. Finally, HFS is a female voice with British accent female recorded in neutral emotion for an open domain adventure game.

Name	Accent	Gender	Duration [hours]	Contextual units	Domain
DIG	US	Male	5.40	55245	Action game
RJS	UK	Male	5.72	83528	Generic newspaper
HFS	US	Female	3.94	58068	Adventure game

Table 5.1: English corpus detail information.

5.1.3 Emotional Castilian Spanish corpus

The emotional speech database used in this work was developed to learn the acoustic models of emotional speech and to be used as a database for a synthesizer. It was used in an objective evaluation conducted by means of automatic emotion identification techniques using statistical features obtained from the prosodic parameters of speech (Iriondo et al., 2007b). For the recording of the corpus, a

	Happy	Sad	Neutral	Other
Happy	81.0%	0.1%	1.9%	17%
Sad	0.0%	98.8%	0.5%	0.7%
Neutral	2.3%	1.3%	90.4%	6%

Table 5.2: Average confusion matrix for the subjective test. The first columns refers to the original label and the first row reflects the listener response.

female professional speaker was chosen among others as she could successfully convey the suitable expressive style to each text category (simulated/acted speech). For the design of texts semantically related to different expressive styles, an existing textual database of advertisements extracted from newspapers and magazines was used. Based on audio-visual publicity, three categories of the textual corpus were chosen and the most suitable emotion/style were assigned to them: New technologies (neutral) [1], education (happy) and trips (sad-melancholic). It is important to mention that the speaker had previously received training in the vocal patterns (segmental and suprasegmental) of each style. The use of texts from an advertising category aims to help the speaker to maintain the desired style through the whole recording session. Therefore, the intended style was not performed according to the speaker's criteria for each sentence, but all the utterances of the same style were consecutively recorded in the same session following the previously learned pattern. The speaker was able to keep the required expressiveness even with texts whose semantic content were not coherent with the style.

For each style, 96 utterances were chosen and the test set was divided in several subsets. A forced answer test was designed with the question *What emotional state do you recognize from the voice of the speaker in this phrase?*. The possible answers were the 3 styles of the corpus plus one more option for *other*. Table 5.2 shows the percentage of identification by style and test, being the

[1] Neutral emotion is in fact the Spanish voice described in Section 5.1.1.

sad style the best rated, followed by neutral and finally happy one.

5.2 Experimental tests

Experiments in this section have been used to analyze common issues related with TTS systems. Specifically, Section 5.2.1 describes an experiment to compare phonemes and diphones and Section 5.2.2 shows HMM-based TTS systems with different sample rates.

5.2.1 Evaluating the synthesis unit

Speech synthesis systems can use two types of basic units: phones and diphones. A diphone is an adjacent pair of phones. On the one hand, a phone-based unit selection system has much less units than a diphone system because the latter has to consider all possible combinations among phones. On the other hand, the "center" of a phonetic realization is the most stable region, whereas the transition from one "segment" to another contains the most interesting phenomena, and thus the hardest to model (Lenzo and Black, 2000). The diphone cuts the units at the points of relative stability, rather than at the volatile phone-phone transition, where so-called coarticulatory effects appear.

The following test compares the quality of an HMM using phonemes and diphones. Both systems were trained using a non-explicit duration model (Section 2.4) and a discontinuous F0 modelling (see Section 3.4). Vocal-tract is modelled by 39-order STRAIGHT mel-cepstral coefficients (see Section 3.3.5). They use the mixed excitation based on aperiodicity described in Section 3.5.4. Contextual factors for these systems are irrelevant since both systems use the same contextual factors and the only difference is the basic unit (phoneme and diphone). In addition, HMM parameters are enhanced using GV (see Section 2.6.1).

The corpus under study is the English DIG. It was decided to analyze this language due to the following reasons:

- This English corpus is larger than the Spanish voice and in consequence more joins are likely to occur.
- The main interest of studying the best synthesis unit size is related to the HMM-driven hybrid approach described in Chapter 4 and Section 4.3. This hybrid approach is developed in English as it requires a speaker independent voice which uses a vast the amount of data. This data was available in English at the time of this research.

This experiment is an AB test conducted with 6 listeners and 20 random sentences. Unlike previous subjective tests, listeners were presented 5 options (see Figure 5.1) to answer the following question *Which sentence do you prefer in terms of quality?*: system A is much better, system A

is slightly better, no preference for any system, system B is slightly better and system B is much better.

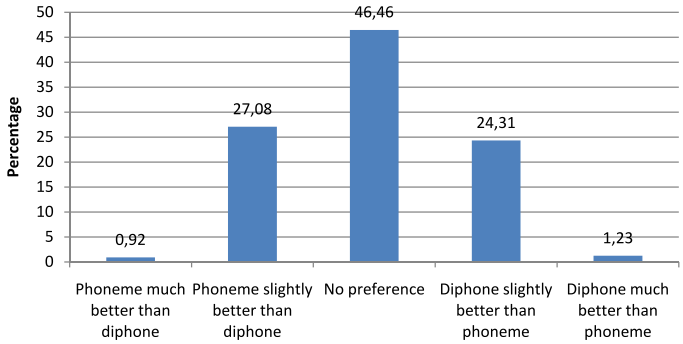


Figure 5.1: Preference test for phones and diphones.

Figure 5.1 shows the performance for both unit types. Since HMM-based TTS systems produce a smooth parameter trajectory there are no concatenation errors or glitches as in the Co-TTS systems. Due to this reason, AB test results show no preference for neither of both types. Consequently, either phonemes or diphones can be used in an HMM-based TTS system without quality degradation. Nevertheless, for a conventional parameter generation system, a phone-based system is more attractive since the footprint of the voice can be reduced. On the other hand, as it is described in Section 4, the use of diphones can be interesting for certain types of hybrid systems. With this experiment, good performance of diphones is validated.

5.2.2 Evaluating different sampling rates

As described in Section 3.6 the conventional working sampling rate is 16 kHz. Nevertheless, it is obvious that the higher the sampling rate, the better quality is likely to be produced and HMM-based TTS systems should not be an exception.

In the following subjective experiments, three different sampling rates are compared: 16 kHz, 22 kHz and 44 kHz. These three frequencies has been selected for being standards used in audio and speech. For each sampling rate a different HMM-based TTS system has been built. All systems utilize a non-explicit duration model (Section 2.4) a discontinuous F0 modelling (see Section 3.4)) without mixed F0 contour. Mixed excitation is designed to use aperiodicity as described in Section 3.5.4 and HMM parameters are enhanced using GV (Section 2.6.1). The only difference between each system is:

- Vocal-tract modelled by a 39-order STRAIGHT mel-cepstral coefficients (see Section 3.3.5) using the warping frequency factor according to Table 3.2.
- Aperiodicity uses a different multiband configuration (see Table 3.3).

This subjective test was conducted with the DIG English voice and 10 randomly selected sentences. A five steps (1-5) MOS (quality evaluation: bad, poor, fair, good and excellent) was conducted on 6 listeners. Results are shown in Figure 5.2.

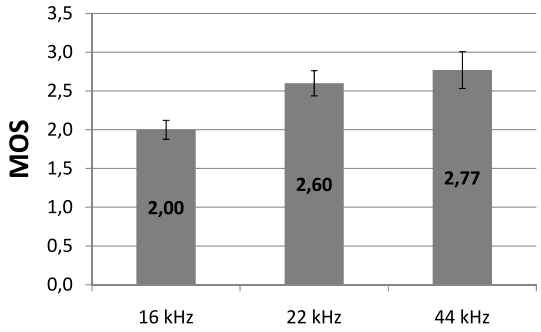


Figure 5.2: A five steps (1-5) Mean Opinion Score (MOS) comparing different sampling rates for the English DIG voice.

The result shown in Figure 5.2 indicates that the higher the sampling rate, the higher the quality of the HMM system. However, although 44 kHz is brighter than the rest of the sampling rates, its improvement is not statistically significant with respect to 22 kHz. This is caused because synthetic speech using 44 kHz resulted in a buzzier quality. This might be due to the fact that aperiodicity parameters would need to use a better approach than simply averaging 9 bands.

In addition, the higher the sampling rate, the more parameters needs to be modelled. Ultimately, this yields to larger voice's footprints. In consequence, 22 kHz seems to be the best trade-off between quality and performance.

5.3 Proposed work and baseline improvements

The following experiments show the results for all the proposed work presented in this thesis. First, the F0 enhancing technique is evaluated in Section 5.3.1 with subjective and objective experiments. Second, linguistic features for Spanish are analyzed in Section 5.3.2. Then, Section 5.3.3 shows a comparison of different mixed excitation approaches. Finally, results for the proposed hybrid system are shown in Section 5.3.4.

5.3.1 Evaluating the F0 enhancement for Spanish

The goal of this experiment is to evaluate the approach proposed in Section 2.6.3. The problem of HMM synthesis is the over-smoothing of the generated parameters which affects both vocal-tract and F0 sequences. In order to improve the expressiveness of the system, an external F0 estimator based on Case Based Reasoning (CBR) (see Section A.1) is introduced into the synthesis stage. The

resulting F0 is a mixed contour combining F0 and CBR estimations. Consequently, three systems are under evaluation here depending whether prosody is estimated from HMM, from the CBR system or from the mixed approach described in Section 2.6.3.

The purpose of the first objective evaluation is to assess the performance of the Spanish HMM prosody (F0 and durations) with respect to the external Case Based Reasoning (CBR) estimator and natural speech, respectively. By measuring the relative error between the estimated F0 and durations, it is possible to estimate the actual expressiveness problem in the Spanish HMM system. Error is measured using a Root mean Squared Error (RMSE) estimation. RMSE for F0 is obtained in a frame-basis whereas duration error is calculated by phoneme.

In the second part of the experiment, a subjective test determines the improvement of the mixed F0 contour approach with respect to the conventional HMM system.

HMM-based TTS system is trained using a non-explicit duration model (Section 2.4) and a discontinuous F0 modelling (see Section 3.4). Vocal tract uses 24-order mel-cepstral coefficients (see Section 3.3.3). The system uses the mixed excitation based on voicing strengths described in Section 3.5.3.2. Decision trees are built using the set of contextual factors described in Section 2.8.3. By contrast, the CBR system uses the features from Table A.1. Postfiltering described in Section 3.3.3.1 is used to enhancement speech and no GV is applied here.

5.3.1.1 Objective test

This objective experiment compares natural F0 and durations with the HMM-based TTS system and the CBR approach. The test is built in Spanish using the PTR voice. Evaluation of these parameters is a key step for various reasons (Keller and Keller, 2003). Primarily because both F0 and duration accuracy estimations are crucial in a source-filter model approach. Secondly, because these parameters describe the level of expressiveness. In order to mix F0 contours, weights have been empirically fixed to $\alpha_\sigma = 0.6$ and $\alpha_\mu = 0.3$ (recall that α_μ controls the amount of F0 offset used and α_σ weights the variance of the contour). The effect of this configuration is to benefit the CBR contour whereas the mean of the F0 is just slightly modified. As a consequence, distortions due to elevated F0 values are avoided.

This objective measurement evaluates the RMSE (estimated vs. real) of the F0 contour (see Figure 5.3) and the mean duration of each phoneme (see Figure 5.4).

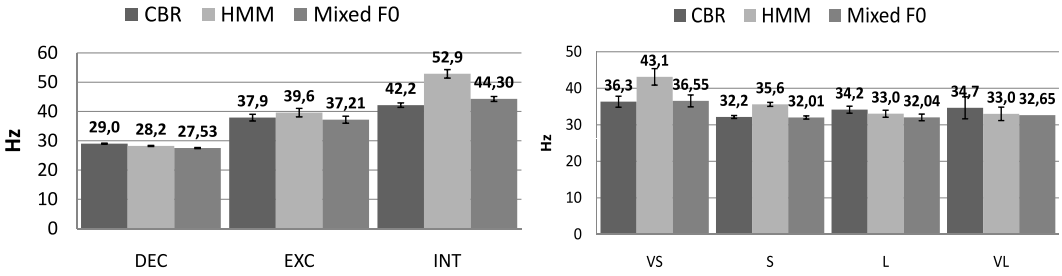
Results are presented for various phrase types and lengths (number of phonemes). Phrase lengths classification is referenced to the corpus average length. Each sentence is classified as very short (VS), short (S), long (L) and very long (VL) according to its length (see Table 5.3).

On the one hand, Figure 5.3(a) shows that CBR and HMM systems have a similar performance for F0 except for the case of interrogative sentences. In general, short ones are worse (see Figure 5.3(b)). On the other hand, Figure 5.4 depicts the RMSE for durations. Both HMM and CBR systems have similar duration estimations although in this case, interrogative sentences are worse with an HMM

Sentence classification	**Length** l
Very short	$l \leq \mu - 2\sigma$
Short	$\mu - 2\sigma < l \leq \mu - \sigma$
Large	$\mu - \sigma \leq l < \mu + 2\sigma$
Very large	$l \geq \mu + 2\sigma$

Table 5.3: Classification of an utterance with respect to the average length of a sentence in the corpus where the mean length is μ and the standard deviation σ.

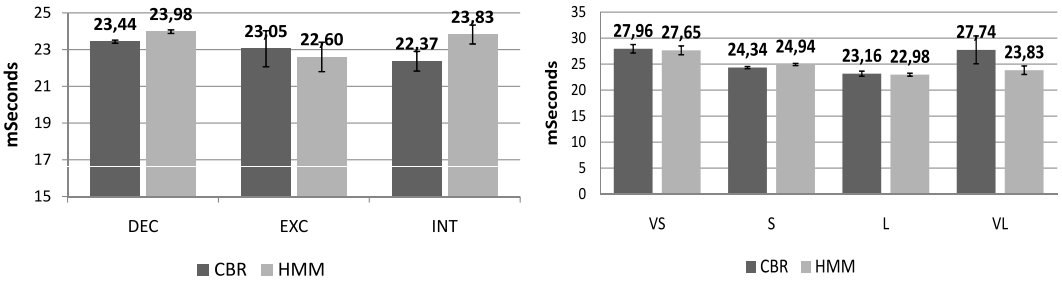
whereas VL sentences have a larger error if CBR is used. Moreover, the error for declarative sentences is also the highest for HMM contours. Nevertheless, the maximum error difference for duration is not higher than 2 msec. For this reason, only F0 contour has been enhanced. The consequence of these results shows that HMM generally estimates worse F0 contours for interrogative sentences. Obviously, this kind of sentence is the most expressive one.



(a) Error vs. phrase type.

(b) Error vs. phrase length.

Figure 5.3: RMSE for F0 contour.



(a) Error vs. phrase type.

(b) Error vs. phrase length.

Figure 5.4: RMSE for duration.

Once the prosody errors has been located, let us turn into some real examples of the effect of the mixed F0 contour. A F0 contour for a short interrogative sentence is shown in Figure 5.5 where both HMM and CBR estimate a similar curve for the start of the sentence (i.e., up to frame 150).

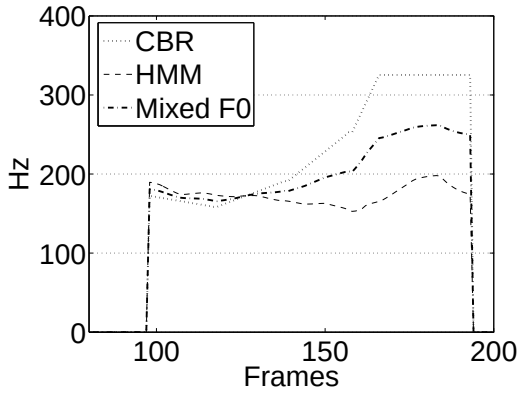


Figure 5.5: Example of F0 estimation for HMM-TTS "Y ahora?" translated as "And now?"

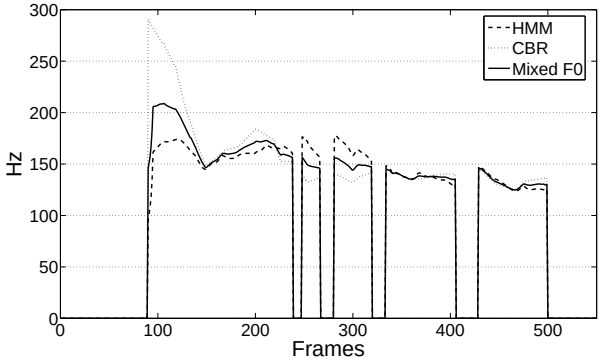


Figure 5.6: Example of F0 estimation for sentence "Una herramienta para privilegiados." translated as "A tool for privileged people."

However, towards the end, CBR becomes a better approach. This might be due to the fact that it reproduces fast changes better. Mixed F0 contour introduces the necessary variations when the expressiveness must be improved thanks to the CBR system, hence the final F0 contour yields to a better overall intonation. Another example is depicted in Figure 5.6. In this case, this is a long enunciative sentence. As we can see, contours are very similar for CBR and HMM. However, the first frames contain a peak error that the mixed F0 approach is able to enhance thanks to the CBR estimation.

5.3.1.2 Subjective test

The goal of this subjective experiment (see Figure 5.7) was to evaluate the preference of the F0 estimation approach. For this purpose, 7 listeners were presented with 20 interrogative and exclamative sentences. The experiment was designed as an AB test where two versions of every sentence were presented (one produced with the HMM estimate and another with the mixed F0 technique). Declarative sentences were not taken into account in this evaluation since objective measurements indicated that HMM and CBR estimates are very similar.

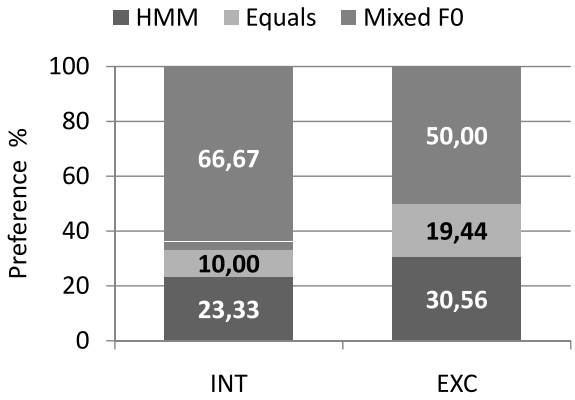


Figure 5.7: Preference test for prosody and phrase type.

The preference of the users was biased towards the mixed system either for interrogative or exclamative sentences. For the latter, the preference of the mixed system decreases and the non preference option is more important than for the interrogative case.

In this experiments, we have shown that the mixed F0 estimation is more expressive than the conventional HMM prediction. By using Equation 2.73 and the mean and variance weights, the quality of the mixed F0 system is maintained. The main reason to justify the overall expressiveness improvement relies on the fact that the external CBR system estimates the F0 contour in syllable blocks whereas the HMM systems generates F0 predictions in a frame-by-frame basis.

5.3.2 Evaluating the linguistic features for Spanish

In order to adapt the conventional HMM-based TTS system to the Spanish language, linguistic features must be changed. Section 2.8.3 describes the Spanish attributes used to build the decision trees. During the first stage of this thesis, linguistic features were analyzed using Festival (Black et al., 1999) and exactly the same contextual factors attributes employed by the English system (Tokuda et al., 2002b). With the purpose of improving intelligibility and expressiveness of Castilian Spanish, the SinLib system was designed (see Section E.3). The main contributions and changes to the linguistic features are:

- The use of Intonational Group (IG) and Accentual Group (AG).
- The SinLib grapheme-to-phoneme conversion is a rule-based system specifically designed for Spanish. Therefore it describes the language much better.

Two experiments are presented here. On the one hand, a subjective test compares the SinLib and the Festival approach in order to assess the quality of the proposed linguistic features. On the other hand, the resulting decision trees are discussed showing percentages of some of the key linguistic features.

5.3.2.1 Subjective test

This experiment performs a subjective test to evaluate the preference of the listeners towards the Festival or the Sinlib system. Therefore, the test compares two systems in terms of linguistic performance. Both systems were trained using a non-explicit duration model (Section 2.4) and a discontinuous F0 modelling (see Section 3.4)). Both systems use 24-order mel-cepstral coefficients (see Section 3.3.3) and a mixed excitation based on voicing strengths described in Section 3.5.3.2. The only difference between them is that decision trees are built with different linguistic information. In addition, postfiltering described in Section 3.3.3.1 was used to enhancement speech and no GV is applied here.

Twenty random sentences of the PTR voice were presented to 15 listeners and they could choose between 3 options (one is a non preference option) answering the question: *Which sentence do you prefer in terms of quality?*.

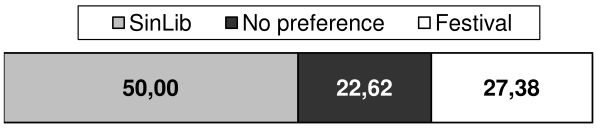


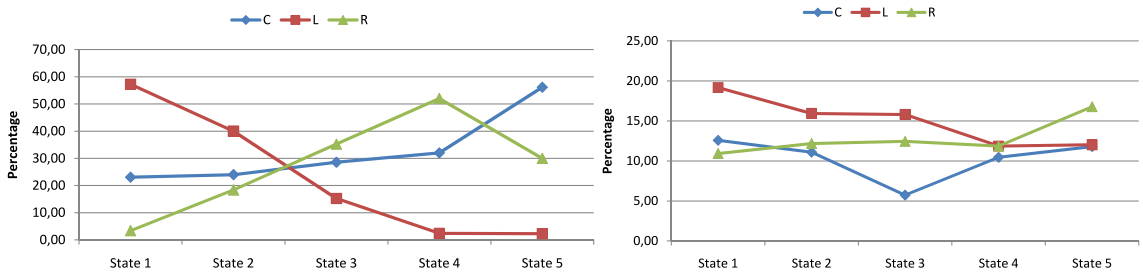
Figure 5.8: Preference test for linguistic improvements.

Figure 5.8 shows the preference results for both systems. Clearly, the designed SinLib system is preferred because this system implies a better representation of each unit (contextual factors are

more descriptive for Spanish) and a vast amount of errors in the grapheme to phoneme conversion were fixed with respect to the Festival system.

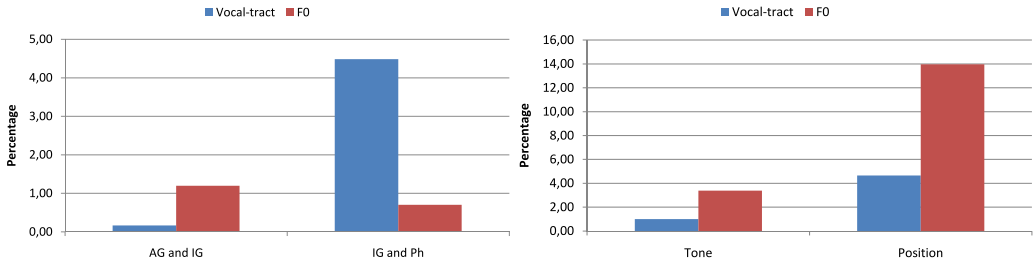
5.3.2.2 Discussion

As we have described in Section 2.8, decision tree-based clustering selects the best set of questions based on the MDL criterion (see Section 2.8.2.2). In particular, questions are built using the Spanish contextual factors shown in Table 2.4 and described in Section 2.8.3. As we introduced in these sections, vocal-tract and F0 trees are likely to use different questions since these parameters are modelled by different factors (e.g., F0 is related to the whole utterance). Two different tests are



(a) Vocal tract. (b) F0.

Figure 5.9: Percentage of questions and HMM state. Percentages are obtained from the Spanish voice for vocal-tract and F0. The type of phoneme is compared for current (C), left (L) and right (R) context.



(a) Questions related to AG, IG and phonemes. (b) Questions related Tone (type of IG) and positioning.

Figure 5.10: Percentage of questions and state for vocal-tract and F0. Questions related to AG, IG and phonemes.

presented. Firstly, the use of questions related to the type and position of phonemes and secondly, a discussion about AG and IG modelling.

Figures 5.9 shows the percentage of questions related to the type of phoneme (e.g., Is the current phoneme a vowel "a"?) including left and right context (i.e., current (C), left (L) and right (R) phonemes). As you can see, the average use of phoneme type questions is fairly constant for F0

(around 15% in Figure 5.9(b)) whereas vocal-tract (see Figure 5.9(a)) is state-dependent. As a consequence, first state of the HMM is clustered taking into account more questions of the left context and the last state uses more questions of the right context. This is because, as we expected, vocal-tract is strongly dependent on the phoneme identity and, in addition, the left context is obviously influenced by the first state of the HMM whereas the right context is more related with the last state.

Moreover, Figure 5.10(a) shows two measures in order to discuss the intonation modelling in the Spanish voice. On the one hand, the percentage of questions using IG and AG (e.g., Position of AG in IG?) and questions using IG and phonemes (e.g., Is current phoneme the end of the IG?). Unsurprisingly, vocal-tract is modelled using more questions related to phonemes whereas F0 is clustered with questions related to AG and IG. This is because the F0 contour is better modelled by information related to groups of syllables (AG) and vocal-tract by phonemes.

Finally, Figure 5.10(b) shows the percentage of questions using the type of utterance tone (i.e., type of IG) and an average percentage of all the questions related to positioning (e.g., Position of current phoneme in the word? or, position of previous syllable in the word?). As you can see, questions related to tone and general positioning are more used to cluster F0 than vocal-tract. This is absolutely normal for the same reason discussed earlier. F0 contains more information of the utterance as a whole where vocal-tract is focused on clustering phoneme spectrum similarities.

We can conclude from that discussion that AG and IG are working as expected. Vocal-tract is clustering phoneme information because HMM observations are related to spectrum whereas F0 is clustering positions, tones and syllables in IG since F0 contour contains a generic representation of the utterance.

5.3.3 Evaluating mixed excitations

The following subjective test is intended to compare different excitations algorithms. As described in Section 3.5, there exists three different excitations: pulse and noise (conventional excitation described in Section 3.5.1), voicing strengths and Fourier magnitudes (multiband mixed excitation in Section 3.5.3.2, standing ME in this experiment) and aperiodicity (AP) excitation from Section 3.5.4.

An HMM system was trained for every excitation using a non-explicit duration model (Section 2.4) and a discontinuous F0 modelling (see Section 3.4)). With the purpose of comparing excitations a 24-order mel-cepstral coefficients (see Section 3.3.3) is used. The Spanish PTR voice was used for this purpose and decision trees were built using the linguistic contextual factors described in Section 2.8.3. The only difference between them is the type of excitation. Moreover, no mixed F0 contours are used in this experiments and only postfiltering (Section 3.3.3.1) is used to enhance speech.

Different excitations were compared using a subjective AB test. There are two experiments. In the first test (Figure 5.11), conventional and voicing strengths excitations are evaluated. In the

second test (Figure 5.12), aperiodicity and voicing strength excitations are compared. Rather than using a single experiment, it was decided to perform two different evaluations in order to assess the performance of each mixed excitation separately. If both mixed excitations were compared simultaneously, results could not show a realistic preference between them. Both subjective experiments are evaluated with 20 random sentences and 15 listeners.

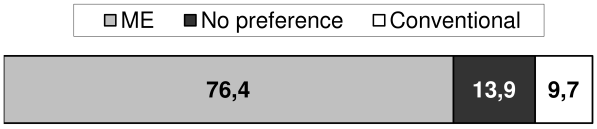


Figure 5.11: Preference test for the pulse excitation and the multiband voicing strength mixed excitation systems.

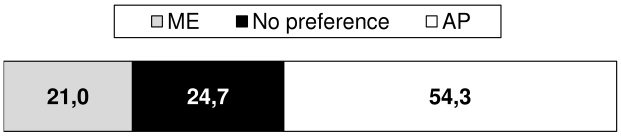


Figure 5.12: Preference test for the multiband voicing strength mixed excitation (ME) systems and the aperiodicity excitation (AP).

Figure 5.11 shows the preference of the multiband excitation in comparison with the conventional one. The effect of the ME (i.e. speech reconstruction buzzy is significantly reduced) is more important than the linguistic improvement shown in Experiment 5.3.2.

Results of the second test (see Figure 5.12) show that aperiodicity excitation is preferred to the multiband voicing strengths excitation. In this case, aperiodicity is a finer representation of the "inverse voicedness" than voicing strengths are of the "voicedness". Note that unlike the voicing strength excitation, aperiodicity is constructed by filtering in the frequency domain rather than in the time domain.

5.3.4 Evaluating the HMM-driven hybrid system

The purpose of this experiment is to assess the quality of the hybrid system proposed in Section 4.3. This system combines an average voice training (Section 2.7.3), a target speaker adaptation (see Section 2.7) and a hybrid Co-TTS system described in Sections 4.2.1 and 4.3.4.

Speaker independent training for this hybrid is further detailed in Section 4.3.3 and it includes 8 speakers and a total of 21.2 hours of speech (see Table 5.5). Target adaptation data consists of a 50 minutes recording of the DIG voice or in other words, 1000 open-domain utterances automatically segmented.

The HMM system uses an explicit duration model (Section 2.4.3) and a discontinuous F0 modelling (see Section 3.4)). Vocal-tract is modelled using a 39-order STRAIGHT mel-cepstral coef-

ficients (see Section 3.3.5). Mixed excitation is designed using aperiodicity as described in Section 3.5.4. No mixed F0 contour is used. Decision trees are built using English contextual factors of Experiment 5.5.2. In addition, HMM parameters are enhanced using GV.

The parameters of the weight function (see Section 4.3.6) have been empirically fixed to the following values: $s_U = 0.2$, $s_M = 0.1$ and $s_L = 0.05$. This configuration benefits the use of the concatenative system while smoothing the concatenation points. Note that for smaller amounts of target speaker data s_U may need to be increased since the Co-TTS system will produce worse joins.

Hereafter, we refer to three systems: the standard statistical HMM system used as a "backbone" of the hybrid system, the concatenative synthesis system employed to select the natural units to improve the quality of the latter and, finally, the proposed hybrid system that combines the previous two approaches.

Two tests are presented in this experiment in order to compare the hybrid system against the conventional HMM-based TTS system using parameter generation. First, a subjective test evaluates the preference of the hybrid system and, second, an objective measure let us evaluate the distortion of the hybrid vocal-tract sequences.

5.3.4.1 Subjective test

An AB test was performed in order to evaluate the performance of the proposed hybrid system. In this test, 10 listeners were presented with 24 utterances randomly chosen. The results, shown in Table 5.4, indicate that the proposed approach increases the average quality of synthesized speech. The hybrid system is preferred by 61.68% of the listeners in comparison to the 23.77% preferring the conventional HMM system.

Preference	Percentage
HMM much better than hybrid	7.19
HMM slightly better than hybrid	16.68
No preference	14.43
Hybrid slightly better than HMM	46.67
Hybrid much better than HMM	15.01

Table 5.4: AB test for the HMM and hybrid systems.

5.3.4.2 Objective tests

In order to study the effect of the weight function, an analysis of distortion is presented in Figure 5.13. Natural sequences of vocal-tract are compared with the HMM and the hybrid systems using the distortion between mel-cepstral coefficients using the following equation:

$$d_{12} = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{M}\left(c_{1,t}(k) - c_{2,t}(k)\right)^2$$

where c_1 is the natural reference, c_2 is the system under analysis (i.e., hybrid or conventional HMM system), M is the length of the coefficients and T is the total number of frames.

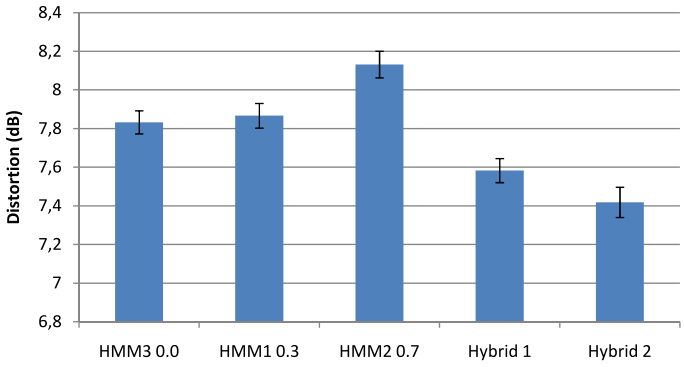


Figure 5.13: Distortion analysis to compare the effect of the weight function. Two systems are shown: conventional HMM with different GV weights ($\alpha_1 = 0$, $\alpha_2 = 0.3$ and $\alpha_3 = 0.7$) and two hybrid approaches with the following weight boundaries: ($s_U = 0.9$, $s_M = 0.8$ and $s_L = 0.7$) and ($s_U = 0.2$, $s_M = 0.1$ and $s_L = 0.05$). Significance $p = 0.05$ is also shown.

In the figure, "HMM α_x" refers to the HMM system with a global variance weight α_x while "Hybrid s_u" is the hybrid system with different upper weights for the sigma (see Equation 4.12). On the one hand, we can see how the effect of the global variance in order to alleviate over-smoothing increases the distortion of HMM system with respect to the natural speech. On the other hand, two hybrid approaches are also compared using two configurations in order to analyze the effect of the weight function. The first uses a relatively high weight ($w(f) \in [0.7, 0.9]$, the LMGE algorithm has no much effect) and the second uses a lower weight ($w(f) \in [0.05, 0.2]$) so the updating process is more intense. As you can see, both have a lower distortion than the HMM system. Moreover, as the weight function uses lower values, the distortion also decreases.

The main advantage of the hybrid system with respect to the conventional GV approach is that the mean and the variance of the models are updated according to the natural units generated by the concatenative module. Although brightness is improved by using GV, updating the variance of the generated parameter sequence is, in fact, distorting the signal since is a process with no natural reference. This distortion can be audible if the GV weight is very high.

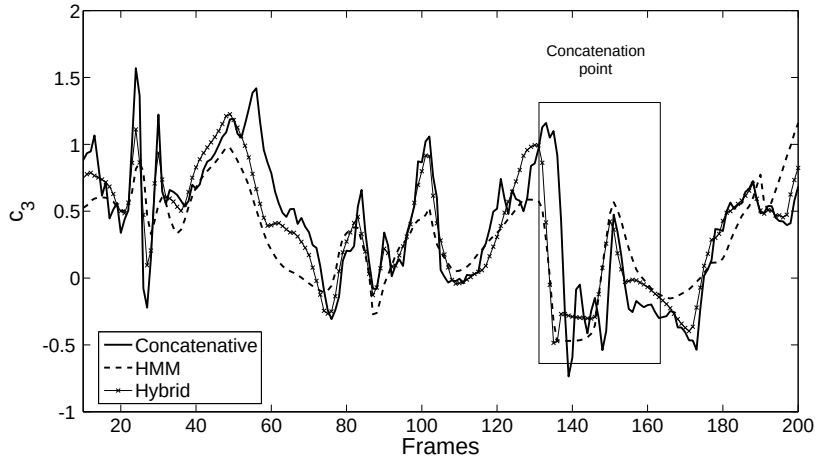


Figure 5.14: Mel-cepstrum sequences for the 3rd coefficient. The same phoneme duration was used for the three systems.

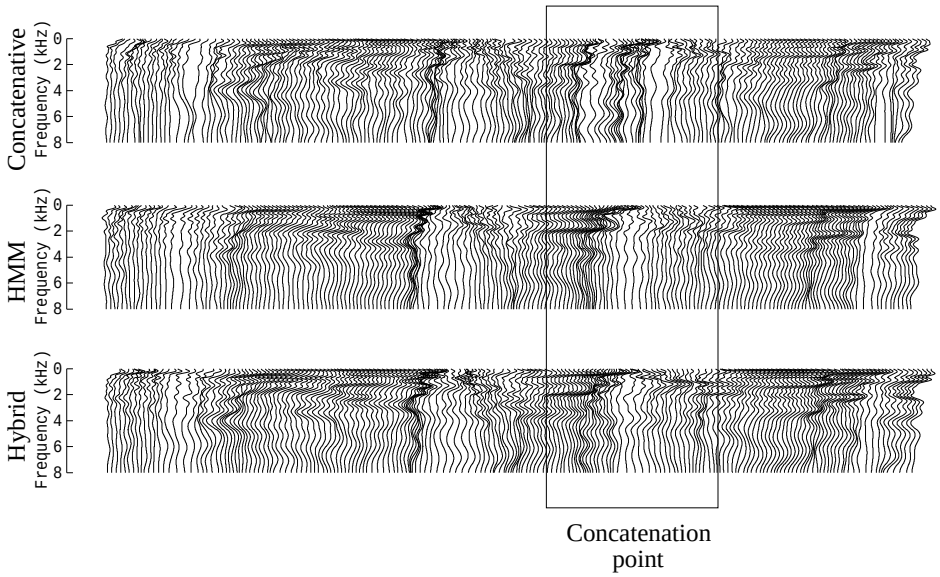


Figure 5.15: An example of generated spectrum sequences. Text "*took you*"

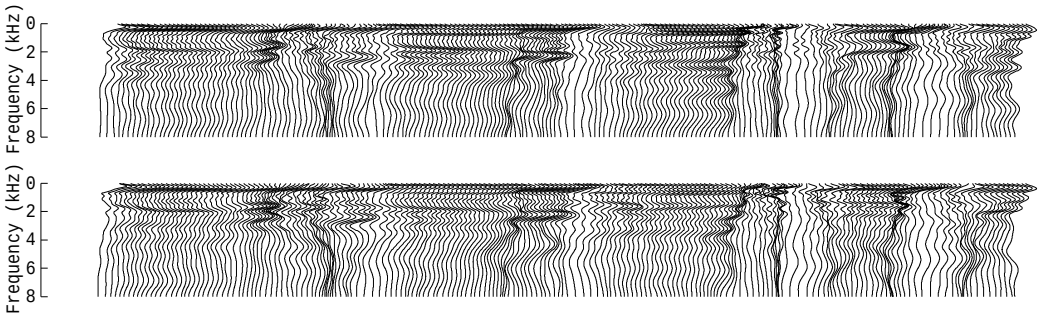


Figure 5.16: Conventional HMM and hybrid examples of generated sequences for the text "*the city is*".

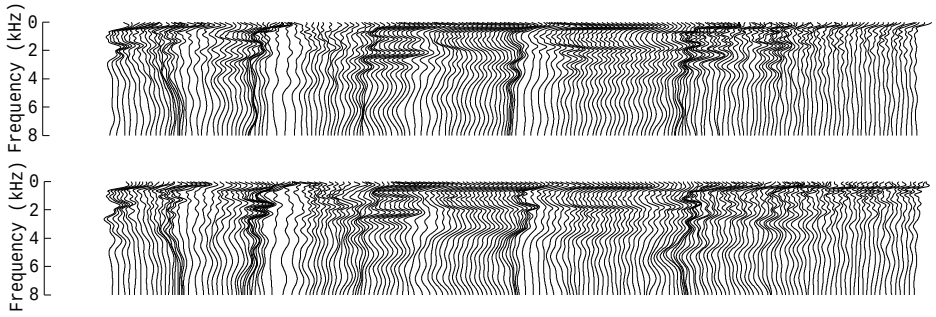


Figure 5.17: Conventional and HMM examples of generated sequences for the text "*always*".

5.3.4.3 Some examples

Figures 5.14, 5.15, 5.16 and 5.17 show examples of parameters trajectories generated by the three systems. Figure 5.14 shows a sequence of the 3rd mel-cepstral coefficient extracted from the three systems. It can be observed that the mel-cepstral sequence generated by the proposed hybrid system is closer to the sequence produced by the concatenative system, except in the region of the concatenation point, where the weight decreases and the sequences overlap with the HMM trajectory.

Sample spectra generated by the three systems are shown in Figures 5.15, 5.16 and 5.17. The proposed hybrid system makes spectral peaks much sharper than those generated by the HMM, except in the concatenation point where the spectrum is made more similar to the "default" HMM spectrum, resulting in a smoother joint.

5.4 Applications

Two applications are described in this section as described in the objectives of this thesis. Firstly, an emotional HMM-based TTS system is described where adaptation techniques are used to transform a neutral emotion into happy and sad. Secondly, a speaker adaptation system where an average voice (or speaker independent voice) is converted into a target speaker identity.

5.4.1 Evaluating emotion adaptation

One of the main advantages of HMM-based TTS systems is model adaptation (Section 2.7). Using that technique with a limited amount of target data is possible to transform HMM parameters creating new voices, speaking styles and speaker identities without the need of large amounts of data or performing the full building process described in Section 2.3.

In this experiment, an HMM synthesis system for various emotion styles is evaluated. Unlike the rest of the experiments, HMM is trained using an explicit duration model and Hidden semi-Markov Model (HSMM) (Section 2.4.3). In order to properly adapt emotions, duration must be adapted as well since it plays a very important role (as it is shown later in this experiment, a *sad* emotion uses a slower speaking rate than other emotion styles). By using an explicit duration model a robust adaptation can be used to transform vocal-tract, mixed excitation, F0 and duration parameters as described in Section 2.7. In addition, a discontinuous F0 modelling (see Section 3.4) is used. Vocal-tract is modelled using a 39-order STRAIGHT mel-cepstral coefficients (see Section 3.3.5). Mixed excitation is designed using aperiodicity as described in Section 3.5.4. No mixed F0 contour is used and HMM parameters are enhanced using GV (Section 2.6.1). Decision trees are built using Spanish contextual factors described in Section 2.8.3.

This experiment presents a set of subjective and objective experiments using three emotional speaking styles: neutral, happy and sad. Evaluations and systems use the emotional corpus described

in Section 5.1.3. As described in Section 5.4.1.1 there are two emotion modelling approaches: a corpus-based modelling and an adaptation-based approach. This experiment compares both of them.

5.4.1.1 Emotion modelling

Unlike Co-TTS systems, HMM-based TTS systems can model emotion styles by using a corpus-based approach or more interestingly, using adaptation techniques described in Section 2.7 and particularly detailed for emotion modelling by (Yamagishi et al., 2004; Tachibana et al., 2005, 2006).

There are basically two major approaches for corpus-based modelling (Yamagishi et al., 2005):

- **A style-independent modelling**. In this technique each style is trained independently so there are different decision trees and models for each emotion. In addition, a new root node is added to the decision trees in order to decide which emotion style is to be used. In this method is straightforward to add new a emotion.
- **A mixed-style modelling**. A full acoustic model is created containing all emotion styles. For that purpose, a new contextual factor is added to the full context labels described in Section 2.8.3. This factor is the style label. In this case, since all emotions are modelled in a single acoustic model is not easy to add a new emotion. However, since HMMs are trained using more samples from all styles, similar parameters are used among emotions and therefore the final model is more accurate.

  Moreover as shown by (Yamagishi et al., 2004), naturalness of synthesized speech generated from style dependent and mixed models is almost the same but the number of output distributions of the mixed style model is clearly smaller, so it is a more attractive approach when an efficient solution is required.

As a consequence in this experiments two systems are compared: a mixed style modelling and an adaptation approach to convert a neutral style into the desired emotion using different amounts of data.

5.4.1.2 Adaptation of style models

There are several possible adaptation techniques (see (Yamagishi and Kobayashi, 2007) and Section 2.7). Basically a transformation can be based on linear regression (where the best approach uses Constrained Maximum Likelihood Linear Regression (CMLLR)) or Maximum a Posteriori (MAP). The best performance is obtained when both approaches are used simultaneously taking advantage of their properties.

The physical manifestation of the various emotional expressive styles is complex and affects both vocal tract and prosody and therefore emotion adaptation must not be limited to only vocal-tract but also to prosody. In that respect, in this case an HSMM CMLLR (see Section 2.7.1)

approach is taken where vocal-tract, mixed excitation, F0 and durations are transformed to the target emotion style. Note that mean and variance are obtained by transforming the distribution parameters simultaneously.

5.4.1.3 Subjective evaluation

The purpose of the following subjective experiment was to evaluate the naturalness of different modelling approaches described in previous Section. For that respect, a five steps (1-5) MOS corresponding to the following quality evaluation: bad, poor, fair, good and excellent.

Two subjective evaluations over 20 test utterances were conducted to evaluate the performance of the systems (see Figures 5.18 and 5.19). Happy and sad styles were used as target styles and adapted from the neutral style with 10, 30 and 200 random utterances, respectively. The *Full* identifier in these figures stands for the mixed style modelling using a full corpus built with the happy, sad and neutral styles. The first subjective test evaluates the naturalness [2] and the second the intensity of the synthesized speech to transmit a certain emotion [3].

The first notable result is related to the naturalness of the synthesized speech shown in Figure 5.18. Neutral emotion reaches a 3.8 of MOS in comparison to the happy style which has the lowest MOS and the sad style with the best naturalness. Naturalness of the happy style is affected by high F0 values that distort the quality of the signal whereas the sad style has a better quality close to the natural speech for the style mixed modelling. It is a matter of fact that HMM-based synthesis oversmoothes the generated parameters (see Section 2.6) and therefore this effect is not problematic in the sad style score since it has the lowest variation of the F0 compared to neutral or happy emotions.

Secondly, Figure 5.19 shows a measure of how well the synthesizer actually reproduced each emotion style. The users were asked to score in a MOS test the intensity of the emotion being reproduced (i.e., happy and sad). We believe it is fair to consider that any emotional style is properly reproduced when its score goes over 3. The results show that the happy style is more affected by the number of adaptation utterances than the sad style. In fact, the happy style is not perceived until the system is fairly well adapted with almost 30 utterances. In contrast, the sad style is perceived well with just 10 utterances. This is partly produced by the duration adaptation made using HSMM and CMLLR since sad style is very dependent of the speaking rate.

From these results we conclude that it is particularly useful to use emotion style adaptation with HMMs. We have shown how even small amount of data can reproduce a fair emotion style specially when this style is highly dependent on durations (e.g., sad). On the other hand, perception of styles which depend more on vocal-tract and F0 variations (e.g., happy but there could also be included angry or rough styles) need more adaptation data. In addition, naturalness for these latter styles can also be a problem due to the over-smoothing and F0 which ultimately could be solved by a

[2] Answering the question: *How natural is the following sentence to you?*
[3] Answering the question: *How well the following emotion is transmitted?*

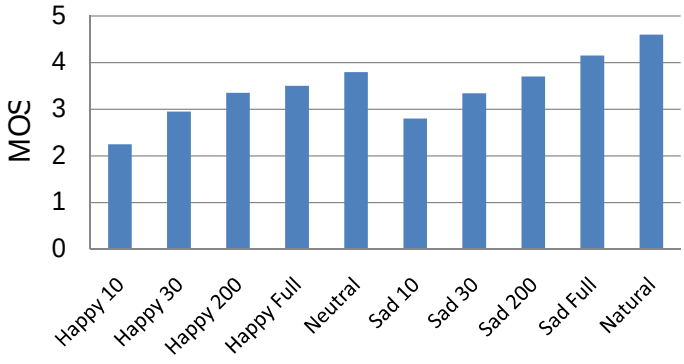


Figure 5.18: MOS for naturalness.

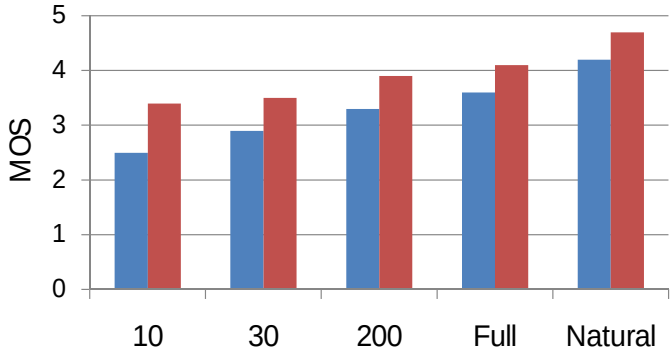


Figure 5.19: MOS for emotion style intensity (sad in red and happy in blue).

mixed F0 contour as described in Section 2.6.3.

5.4.1.4 Objective evaluation

The objective experiment is conducted to reinforce the effect of the emotion style production from a mathematical point of view. In some works, a distortion of the mel-cepstrum is calculated to determine the acoustic distance between the neutral and the target emotional speech (Kawanami et al., 2003), whereas others show a root-mean-squares error of the F0 (Yamagishi. et al., 2006). As the adaptation of emotional expressive styles affects both the vocal tract and the prosody, an evaluation to implicitly measure its performance would be desirable. VoQ parameters comply with these constraints and were shown to be an adequate emotion discrimination method by (Monzo et al., 2007). VoQ parameters are described as a set of measurements to weigh aesthetic features of the speech (e.g. harsh or trembling). The following VoQ parameters were directly calculated over the vowels of 200 synthesized test utterances adapted with 200 target sentences of each style:

- Jitter (J) and Shimmer (S). They describe frequency and amplitude modulation noise, re-

spectively. These parameters are calculated using the new model presented in (Monzo et al., 2008) in order to minimize the effect of the prosody (see jitter in Equation 5.1 and shimmer in Equation 5.2 where i is the analyzed frame, $F0$ is the fundamental frequency, N the size of the analysis window and ϕ is is the peak-to-peak amplitude measured in each pitch period).

$$J_i = \frac{1}{N}\sum_{j=1}^{N-1}\left(F0_i(j+1) - F0_i(j)\right)^2 \tag{5.1}$$

$$S_i = \frac{1}{N}\sum_{j=1}^{N-1}\left(\phi_i(j+1) - \phi_i(j)\right)^2 \tag{5.2}$$

- Glottal-to-Noise Excitation Ratio (GNE) (Michaelis et al., 1997). This parameter indicates whether a given voice signal originates from vibrations of the vocal folds or from turbulent noise generated in the vocal tract, and is thus related to breathiness. GNE is similar to Harmonics-to-Noise Ratio (HNR) and Normalized Noise Energy (NNE).

- Spectral Flatness Measure (SFM). It is computed as the ratio of the geometric to the arithmetic mean of the spectral energy distribution. In Equation 5.3, E_k is the amplitude in bin number k and K is the total number of amplitudes in frequency band j.

$$SFM_j = 10\log\left(\frac{\sqrt[K]{\prod_{k\in\text{Band } j} E_k}}{\frac{1}{K}\sum_{k\in\text{Band } j} E_k}\right) \tag{5.3}$$

- Hammarberg Index (HammI), defined as the ratio between the maximum energy in the $[0, 2000]$ Hz ($E_{0-2000Hz}$) and $[2000, 5000]$ Hz ($E_{2000Hz-5000Hz}$) frequency bands (see Equation 5.4).

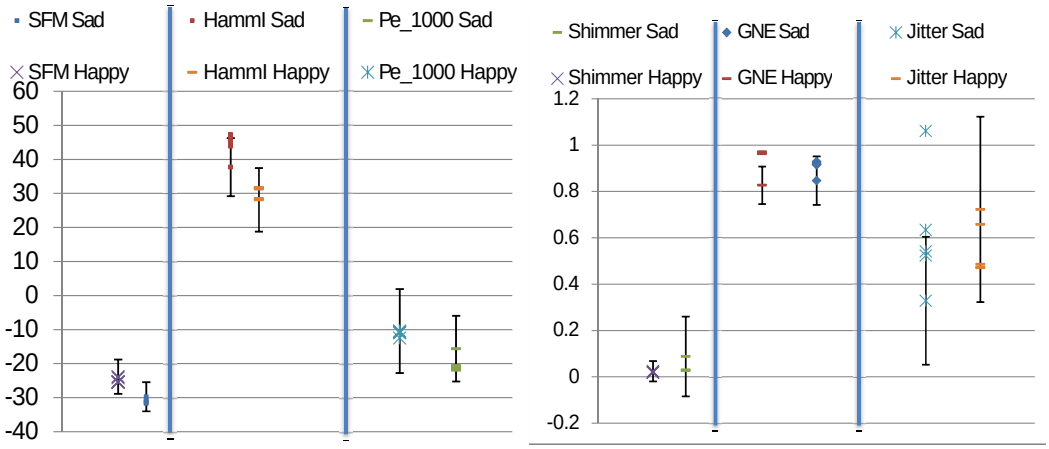
$$HammI = 10\log\left(\frac{\max(E_{0-2000})}{\max(E_{2000-5000})}\right) \tag{5.4}$$

- Drop-off of spectral energy above 1000Hz (Drop_1000), a linear approximation of spectral tilt above 1000 Hz, which is calculated using a least squares method (Abdi et al., 2003).

- Pe_1000: Relative energy in the frequency band $[1000, {}^{s_r}/_2]$ normalized to the energy in the low frequency band $[0, 1000]$ Hz being s_r the sampling rate.

$$Pe_{1000} = 10\log\left(\frac{E_{1000-{}^{s_r}/_2}}{E_{0-1000}}\right) \tag{5.5}$$

Rather than discriminating synthesized emotions, the purpose of these measures is to show the limitation of the emotional adapted HMM synthesis approach. Figures 5.20(a) and 5.20(b) show



(a) VoQ parameters: SFM, HammI and Pe_1000.

(b) VoQ parameters: Shimmer, GNE and jitter.

Figure 5.20: Main VoQ parameters comparing happy and sad styles. The area within lines represents the standard deviation of natural speech.

descriptive statistics of the VoQ parameters for the happy and sad styles. Each column represents a parameter and the standard deviation of the natural speech for that parameter.

Most of the VoQ parameters are inside the standard deviation boundaries. Nevertheless, note that the HammI parameter for the sad style is biased towards upper part meaning that the synthetic speech is more low-pitched than natural recordings. In addition, jitter results indicate that synthesized speech suffers from a trembling effect. These two conditions combined produce an undesirable effect which increases as the adaptation uses more data. In that respect, listeners concerned about certain degree of roughness which could be explained by these VoQ parameters.

Also, it is important to highlight the GNE parameter for the happy style which indicates a distortion. GNE is considered to be an additive noise which is clearly affecting the subjective measure in such a manner its MOS for naturalness is below other styles.

As both effects are mainly caused by the vocal-tract, a possible solution to alleviate it would be to use a speaker independent voice or a style independent style modelling using more training data so HMM parameters were more robustly estimated.

5.4.2 Evaluating speaker adaptation

Speaker adaptation is a similar task to the one described in Experiment 5.4.1. In this case, rather than adapting to an emotion style, adaptation is applied in order to get a different speaker identity. For that purpose, a speaker independent (SI) model is built first and then adapted to a target speaker. Apart from producing a new voice with a small amount of data, the present model is particularly useful in order to get robust target speaker's models because a SI training contains more data than a conventional training.

The purpose of this experiment is to give a subjective measure of the quality of a voice produced using adaptation from an average speaker independent training with respect to a conventional build. The former is defined as a SI system and the latter is a speaker dependent (SD) system. For that effect, the DIG English voice is used as target data. The SI voice is built using 8 speakers with different English accents (US and UK) and different gender but similar age rank. Table 5.5 summarizes the details of each speaker.

	BCL	**BDL**	**BGB**	**CLB**	**DIG**	**JMK**	**RMS**	**SLT**
Size (hours)	5.74	0.84	5.124	1.069	5.397	0.904	1.102	0.943
Mean utt. dur. (sec)	5.51	2.701	2.364	3.39	3.319	2.876	3.50	2.99

Table 5.5: Corpora for speaking independent training. For each speaker, its total corpus length and mean utterance duration is specified.

As described in Section 2.7.3, the average voice in this experiment was created using a conventional speaker dependent-based training. In other words, the training is performed as described in Section 2.3 but instead of having just 1 speaker, there are N. Other training approaches such as Speaker Adaptive Training (SAT) (Gales, 1997) have not been used.

As in the emotion experiment, an explicit duration model (Section 2.4.3) is used so as to improve the adaptation reliability. Also, a discontinuous F0 modelling (see Section 3.4) is used. Vocal-tract is modelled using a 39-order STRAIGHT mel-cepstral coefficients (see Section 3.3.5). Mixed excitation is designed using aperiodicity as described in Section 3.5.4. No mixed F0 contour is used and decision trees are built using a simplified version of the English contextual factors described in Experiment 5.5.2. Moreover, adaptation is based on two stages, first applying CMLLR and a then MAP (see Section 2.7).

Figure 5.21 shows a comparison of a speaker dependent (SD) and a speaker independent (SI) by means of an AB preference test [4]. The SI system was adapted to the target speaker with either 200 or 1000 utterances. In this experiment, 6 listeners were presented 15 random sentences from the DIG voice.

Results seem to indicate that the SI system is not notoriously more preferred than the SD system. The fact that makes SI system preferable in front of the conventional SD is the richness and robustness of the synthesis. A common problem with HMM systems is the small number of training examples. In the case of a SI system, every HMM model has a very consistent model trained with a larger number of examples making each model a much better representation of the unit that it is representing.

From the feedback of the listeners, most of the samples were very similar but in all cases, the intelligibility of one of the samples was better than the other. The intelligibility is directly related to the robustness of the model and the number of training examples.

[4] Listeners were asked *Which of the following synthetic sentences do you prefer?*

Figure 5.21: Preference percentage of the DIG voice for speaker independent and speaker dependent system adapted with 200 and 1000 utterances.

This test was performed for 200 and 1000 utterances of the target speaker respectively. Unsurprisingly, the more utterances of the target speaker, the better the quality is assessed.

5.5 TTS systems performance

Experiments in this section present the overall performance of Spanish and English TTS systems.

5.5.1 Evaluating the overall quality of the Spanish HMM-based TTS system

The primary purpose of this experiment is to evaluate the overall quality of the Spanish HMM-based TTS system. To do this, a comparison assesses three different systems:

- OLD-HMM: the HMM-based TTS system using the basic pulse plus noise excitation (see Section 3.5.1).
- ME-HMM: exactly the same OLD-HMM system but using the voicing strengths mixed excitation (see Section 3.5.3.2).
- E-PSOLA: this is an extended concatenative system based on unit selection. The main peculiarity is the use of a hybrid method based on TD-PSOLA (Moulines and Charpentier, 1990) and the harmonic plus noise model to jointly modify pitch and time-scale (Iriondo et al., 2003).

Both HMM systems were trained using a non-explicit duration model (Section 2.4) and a discontinuous F0 modelling (see Section 3.4)). Vocal-tract is modelled using 24-order STRAIGHT mel-cepstral coefficients (see Section 3.3.3). No mixed F0 contour is used and over-smoothing is alleviated by using postfiltering (Setion 3.3.3.1).

The Spanish PTR voice was used for this experiment and decision trees were built using the linguistic contextual factors described in Section 2.8.3. Both HMM and concatenative systems used

the same number of training sentences. As described at the beginning of this chapter (Section 5.1.1), the training part of the corpus represents 75% of the total number of sentences. Note that the HMM-based TTS system models a high variability F0 contour (μ_{F0}=167 Hz, σ_{F0}=41 Hz) and the concatenative E-PSOLA system uses natural prosody extracted from corpus.

This experiment presents two subjective tests where 25 listeners (most of them students of a technical degree) were asked to evaluate 20 utterances randomly selected. The first test (see results in Figure 5.22) compares HMM and concatenative systems in terms of acceptability, intelligibility and naturalness. Different studies refer to acceptability as a composite measure (Schweitzer et al., 2004). It is clear that in subjective user evaluations, at least intelligibility and naturalness play an important role. Subjective acceptability is not necessarily a simple consequence of intelligibility and, therefore, acceptability tries to establish a distinction between the aesthetic and the functional aspects of synthetic speech. For each of the three concepts evaluated, a five steps (1-5) Mean Opinion Score (MOS) is used corresponding to the following quality evaluation: bad, poor, fair, good and excellent. Listeners were asked the following instructions: *Evaluate the general acceptability of this sentence*; *evaluate the general intelligibility of this sentence* and *evaluate the general naturalness of this sentence*. The second one (Figure 5.23) shows the stability of the acceptability test in a bar graph.

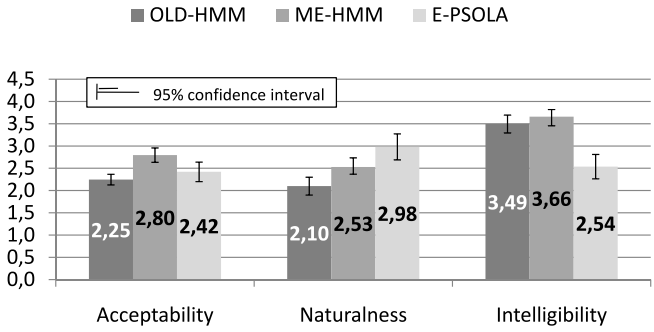


Figure 5.22: Acceptability, intelligibility and naturalness MOS tests for ME-HMM, OLD-HMM and Co-TTS systems. 95% confidence interval are included for the statistical significance.

The results for the first test are discussed as follows:

1. **Acceptability**. Figure 5.22 shows that acceptability reaches the highest score (MOS of 2.8) in the ME-HMM system. This abstract concept is a composite measure that gives an idea of the overall quality of a system. In this case, ME-HMM is clearly better than an HMM using a basic excitation. Because the PTR voice is not a large corpus, bad joins are likely to occur. Therefore, HMM can maintain a constant quality for most of the sentences unlike the concatenative system. As a consequence, ME-HMM produces in fact a more acceptable synthesis for most of the sentences.

2. **Naturalness**. This measurement deals with quality and intonation. It indicates how much

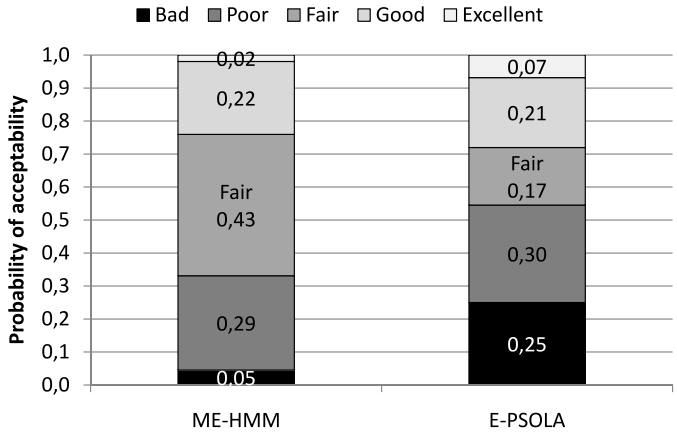


Figure 5.23: Stability comparison based on the acceptability MOS results.

a synthesizer sounds like a human voice. On the one hand, the main problem of the HMM-based TTS system is to produce a flat synthesis. Moreover, although ME-HMM uses a mixed excitation, the best example of a concatenative system still produces a better synthesis than the best statistical reconstruction. On the other hand, concatenative synthesis sounds more like a human but naturalness is affected by quality discontinuities. In any case, ME-HMM improves the quality with respect to the conventional HMM due to the use of mixed excitation and contextual factors (see Section 2.8).

3. **Intelligibility**. This is a measurement of the maximum number of intelligible words in a phrase. While E-PSOLA produces strong discontinuities that affect the comprehension of some sentences, HMM systems solve it by means of a smoother synthesis. This test also measures the effect of the linguistic description (see Section 2.8.3) with respect to the previous Festival implementation (see Experiment 5.3.2).

A second analysis of the results obtained with the subjective test calculates the probability of each of the possible answer of the MOS tests for acceptability which, in fact, is also measuring the stability of the HMM system. Note that the Co-TTS system can produce more high-quality sentences although the probability of producing a bad synthesis is also higher than that of the ME-HMM system. Stability of the ME-HMM system is then guaranteed thanks to a high probability "fair" zone.

From these results, we can conclude that an HMM-based TTS system presents the most stable quality and although is less natural than the Co-TTS system, it avoids quality discontinuities. This stability is one of the main advantages of this kind of systems which can be used in open domain applications even with small corpus such as the PTR voice.

5.5.2 Evaluating an English HMM-based TTS system for real applications

One of the objectives of this thesis was to develop and apply the HMM-based TTS system to real scenarios. In the following Section, a set of experiments will be presented in order to describe the performance of the implemented synthesis engine. In this experiment, performance refers to speed in real time. If a sentence of duration T seconds is synthesized in T_s seconds, the real time performance is

$$R = \frac{T}{T_s} \tag{5.6}$$

Firstly, an objective test was designed for measuring the real time performance and a subjective test helped us to decide the optimal system configuration using a trade-off between quality and performance (see Section 5.5.2.1).

Secondly, a subjective test evaluates the quality of an HMM system against a concatenative system. Note that the Co-TTS system under analysis here is different from the one used in previous experiments and so it is the language. Rather than using the small Spanish voice, larger databases are evaluated here and therefore English voices are used (see Section 5.5.2.2).

The HMM system used in this Section utilizes a non-explicit duration model (Section 2.4) and a discontinuous F0 modelling (see Section 3.4). Vocal-tract is modelled using a 39-order STRAIGHT mel-cepstral coefficients (see Section 3.3.5). Mixed excitation is designed using aperiodicity as described in Section 3.5.4. No mixed F0 contour is used and HMM parameters are enhanced by means of GV (Section 2.6.1). The Co-TTS used in the subjective experiment is based on a Manhattan distance (MHD) unit selection algorithm. A brief description comes as follows:

- The HMM uses a reduced set of English features similar to the one used in the HTS approach (Tokuda et al., 2002b). The main difference lies on improving the training time. Therefore most of the redundancy introduced in the HTS approach is dismissed (e.g., Part-Of-Speech (POS) tagger or numerical phoneme indices positions).
- The Co-TTS system uses a Manhattan distance for the unit selection algorithm (see Section 1.2.2 and (Hunt and Black, 1996; Taylor, 2009)) where weights are hand-tuned.

5.5.2.1 Real time performance

The goal of this objective test was to evaluate speed and size of different HMM-based TTS system configurations. Although one of the advantages of the HMM-based TTS systems is the low footprint of the voices and its performance to run in special devices such as game's engines (e.g., PlayStation®), sometimes further reduction is essential to use this system on devices which have limited memory (Oura et al., 2009). The machine under analysis is a Windows XP 32 bits with a 2.31GHz Core2 CPU with 2Gb of RAM.

Table 5.6 shows a relation between real time performance and quality of the HMM system for different excitation models. On the one hand, real time performance [5] is calculated using Equation 5.6. On the other hand, subjective quality is evaluated using a five steps (1-5) MOS corresponding to the following quality evaluation: bad, poor, fair, good and excellent.

System	R (Win32)	Quality (MOS)
Multiband excitation	7.71	1.8
Aperiodicity FFT=1024	2.74	2.5
Aperiodicity FFT=512	3.68	2.2
Aperiodicity FFT=256	5.36	1.7

Table 5.6: Real time performance for different excitation models.

On the one hand, multiband excitation described in Section 3.5.3 is a very efficient algorithm to compute a mixed excitation since it only involves time domain filters. On the other hand, aperiodicity excitation has been analyzed for different FFT sizes [6] resulting in different system performances. Note that by reducing the resolution of the spectrum, the quality also decreases. A trade-off is needed between performance and quality. During the development of this work, a FFT size of 512 was found to be the best size which guarantees a good quality and significantly reduces the computational cost.

Note that, although different MOS experiments cannot be compared between them, the average MOS obtained in this experiment is slightly below the average result obtained in other experiments. This is basically due to the fact that synthesis experts were evaluating this system. Nevertheless, the difference between MOS is the interesting point here.

5.5.2.2 A subjective quality test

Similarly to the subjective test performed in Experiment 5.5.1, the following evaluation compares the English HMM-based TTS system against a Co-TTS system. In this case the purpose is to evaluate the synthetic naturalness of the HMM technique in a real video game scenario. Most specifically, in the following test, these circumstances take place:

- English data (see Section 5.1.2) is used. As it is described, the size of the voices are significantly larger than the Spanish voice. Also, different accents are taken into account.
- In a real scenario out-of-domain sentences might happen often. Table 5.1 briefly classify the domain of each voice. Out-of-domain sentences are selected in order to clearly differ from the original context.

[5] Real time of the aperiodicity-based system is measured by adding the time to generate the excitation and the time to perform the conventional parameter generation algorithm. This is due to the use of different code implementations (Matlab for the reduced version of STRAIGHT (Kawahara, 1999) and C++ for the synthesis engine).

[6] Different FFT lengths have been forced into the system by aperiodicity frequency bins manipulation.

- Target consumers of video games are usually high-quality demanding users although they are not speech technology experts. Unlike the general conception of a video game player is a young boy, according to Source Entertainment Software Association age distribution of video game players shows an average of 33 years old. Nevertheless, if we base the statistics on frequency of play, the picture is quite a different one. On the basis of this definition, video game player seems to be much younger than 33. Particularly, a more detailed study is shown in Table 5.7 being the rank age from 9-15 the main one. For these reasons, 15 listeners were selected within this age rank to perform the experiment.

Age	9-12	13-15	16-19	20-24	25-44	456-6	67-79
Percentage	51	39	22	19	8	5	5

Table 5.7: Distribution by age of users of video games on a random day (2006). Source Norwegian Media Barometer 2006, Statistics Norway.

All English voices described in Section 5.1.2 were evaluated in this test. Both HMM and concatenative systems use the whole corpus for training and to build the unit inventory, respectively. For each voice, 100 out-of-domain test sentences are used.

HMM and Co-TTS systems (standing MHD) are compared (see Figures 5.24 and 5.25) by means of a nine steps (1-9) MOS corresponding to the following quality evaluation: 5 (perfect, indistinguishable from natural speech), 4.5, 4.0 (very good, only slightly unnatural), 3.5, 3.0 (clear audible problems, but intelligible), 2.5, 2.0 (bad audible problems, unacceptable), 1.5, 1.0 (catastrophic). Unlike previous Experiment 5.5.1, the following subjective test were conducted for each system independently (i.e., each test evaluated an HMM or a Co-TTS approach). The reason for that is because HMM and Co-TTS system are intended to work in completely different applications or at least not performing the same task [7]. Listening tests took place in different days with different subjects which were randomly presented with either HMM or Co-TTS system first.

Figure 5.26 summarizes the average MOS from Figures 5.24 and 5.25. As depicted in the bar graph of Figure 5.24, MHD system is closer to natural speech (in other words, highest percentage of 5 in the MOS scale) for in-domain sentences, whereas concatenation errors affect the overall assessment in out-of-domain utterances (in this case, the percentage of 5 scores decreases from 28% to 7.6%, that is, 4.01 to 3.26 in the MOS scale). On the other hand, HMM system is not as dependent on the domain so the quality is nearly constant between in and out-of-domain tests (MOS of 3.95 and 3.84).

From results shown in Figure 5.24 it is possible to infer that the perception of naturalness for an HMM-based TTS system is comparable to that of a Co-TTS (or even better for out-of-domain

[7] The idea is the following: as we have seen in previous Experiments, it is a matter of fact that HMM and Co-TTS systems produce significant differences in the type of synthetic speech. In a game, the final user is not going to have the opportunity to compare both synthesis approaches with the same voice. The purpose here is to evaluate the viability of the HMM technique as if it were presented to a real user in a real game with no other reference of how a particular voice would sound like.

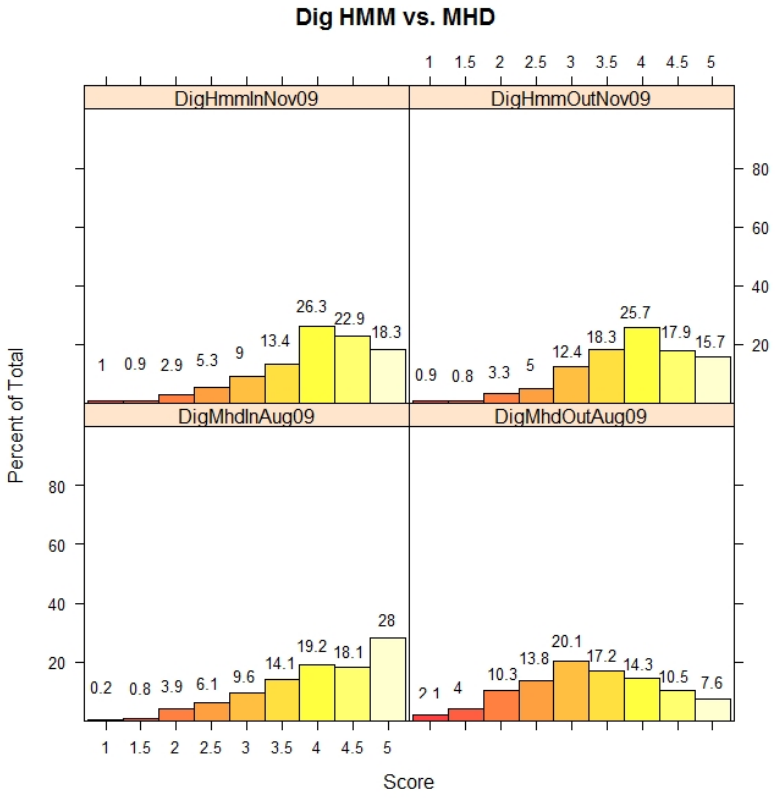


Figure 5.24: Comparison of the DIG voice for texts in and out of context. Each graphic is encoded with a name (e.g., DigMhdInAug09) indicating the voice, the synthesis technique, in or out of context test sentences and the date of the test. Therefore the first row shows results for the HMM system whereas the second row refers to the Co-TTS system. The horizontal axis is the MOS and the vertical axis contains the percentage of each score.

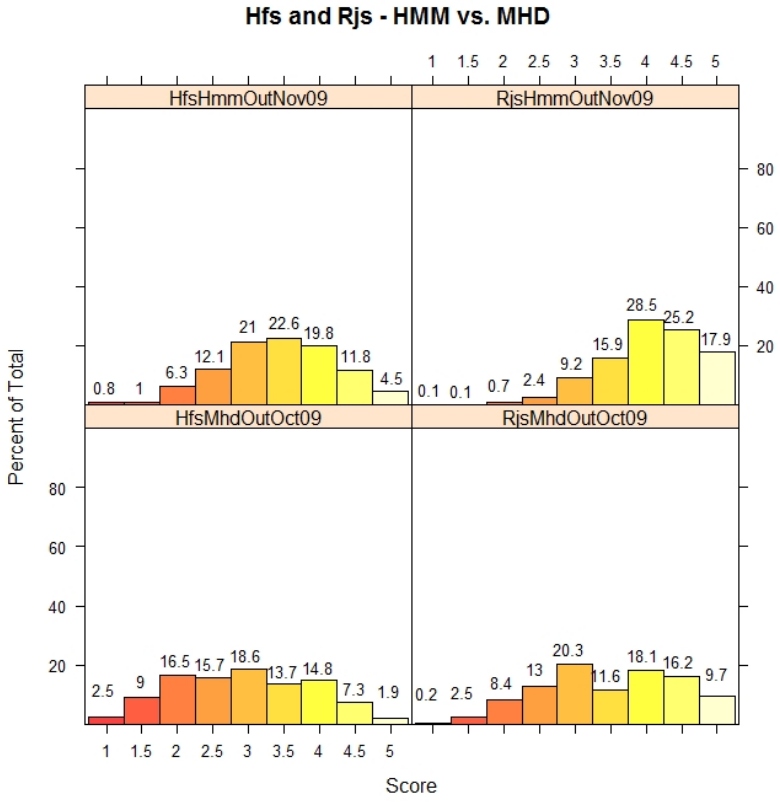


Figure 5.25: Comparison of HFS and RJS voices for texts out of domain. Each graphic is encoded with a name (e.g., HfsMhdOutOct09) indicating the voice, the synthesis technique, out of context test sentences and the date of the test. Therefore the first row shows results for the HMM system whereas the second row refers to the Co-TTS system. The horizontal axis is the MOS and the vertical axis contains the percentage of each score.

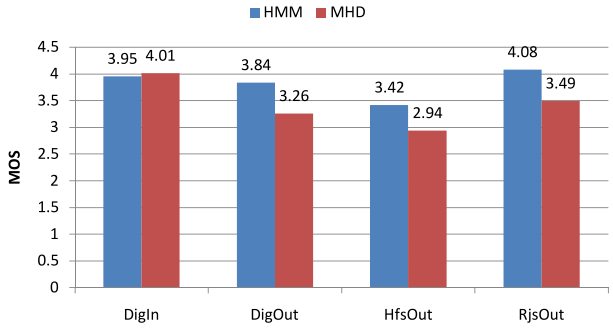


Figure 5.26: Average MOS for all types of systems.

sentences) if the utterances presented to listeners do not have a natural speech reference. Moreover, the robustness of the HMM system makes it a good candidate for open domain TTS systems.

Unlike Figure 5.24, Figure 5.25 only shows out-of-domain results for HFS and RJS voices. Although overall MOS for HFS (3.42 in Figure 5.26) indicates the quality is worse than for RJS (MOS of 4.08 in Figure 5.26), it has to be said that in both cases, HMM is assessed as more natural than MHD (with MOS of 2.94 and 3.49 in Figure 5.26). The main difference between HFS with respect to DIG and RJS is the natural expressiveness of the speaker and the domain. Specifically, the target domain is an adventure game focused on magic tales which actually results in very complicated linguistic structure which clearly differ from what normal English is. Note that the quality of the HFS is also causing a problem in the upper part of the Co-TTS system's score (in this case, the percentage of 5 in the MOS scale is 1.9% wheres it is 9.7% for RJS and 7.6% for DIG). Undoubtedly, RJS results in the best quality with respect to DIG and HFS. As expected, this indicates the advantages of using a neutral domain (i.e, news) and a professional speaker with a constant speaking rate. Although this is a very convenient voice for any TTS system including HMM, the quality improvement is to the detriment of natural expressiveness.

To conclude the results of this test it is interesting to highlight the high MOS scores reached by the HMM system, even more than we expected prior to the start of the test. This is due to the fact that this test was performed independently for HMM and MHD systems and non-expert users. Hence, the natural quality reference is not seen as an obstacle for HMM to be used in a real application such as video games. This is a great advantage for HMM since it shows that actual listeners do not seem to appreciate the lack of naturalness introduced by the vocoder but the unnaturalness of concatenation errors such as glitches or prosody inconsistencies.

5.6 Conclusions

In this chapter we have presented results for all the experiments concerning topics of this thesis. We have described the three types of corpora we have used (i.e., Spanish, English and emotional

Spanish) and then we have classified the experiments into four types:

- **Experimental tests**. On the one hand, we have shown that we can use either diphones or phonemes in an HMM-based TTS system. As a phoneme-based system is more efficient (i.e., less training time and less memory usage), phonemes have been used in the rest of the experiments. On the other hand, we have concluded that in terms of quality and efficiency, 22k Hz is the most optimal sample rate. In case an HMM-based TTS system is to be used with a real application, we recommend this sample rate. However, we consider that this is an isolate configuration that can be used at any time and should not interfere the rest of the experiments. As a consequence, they have been designed with 16k Hz.

- **Proposed work and baseline improvement**. In this section, we have shown the improvement of all the proposed ideas of this thesis:

  - Expressive improvement. By using an external CBR-based F0 estimator and a mixing procedure, we have shown that it is possible to improve the expressiveness of the conventional HMM-based TTS system. Also, we described how to guarantee the stability of the proposed mixed F0 system by setting a weight to control the intensity of the mean and the variance of the external F0 contour.

  - Linguistic features for Spanish. We have validated our proposed Spanish contextual factors for the HMM-based TTS system. In addition, an analysis have also been performed in order to show the percentage of IG and AG used in the decision tree-based clustering scheme.

  - Mixed excitations have also been compared. In particular, the APME and the VSME models have been subjectively compared. As a conclusion, we can say that the aperiodicity is a more natural approach although the VSME approach can still be an option in applications where system performance is a key issue.

  - Finally, the quality of the hybrid system has also been assessed. Firstly, by using a subjective test we have shown the preference of this system in comparison with a state-of-the-art HMM-based TTS system. Secondly, we have shown an objective test. In this test, distortion is measured between hybrid and HMM parameter sequences with respect to the natural reference. As expected, the distortion of our approach is below the distortion of the HMM system using GV. Although the results are promising, we conclude that more work can be done in order to efficiently increase the naturalness.

- **Applications**. We have also presented two types of applications mainly focused on a game application:

  - An emotional HMM-based TTS system. In this experiment, we tested two emotions (i.e., happy and sad) and show how a robust adaptation can be obtained even using a few utterances. As a matter of fact, “sad” style is better conveyed and more natural

than the "happy" emotion. The "sad" emotion is more dependent on durations and, therefore, it is better transformed with less utterances as soon as durations have been slightly transformed. Moreover, "happy" style is more affected by vocal-tract and F0 variations.

- A speaker adaptation test. This experiment shows the performance of creating an average voice (using a speaker-independent approach) and using two different amounts of data to adapt to a target speaker. Unsurprisingly, the more adaptation data is used, the better the adaptation becomes. The interesting point is that, by using an average voice, the robustness of the estimated HMMs increases.

- **TTS system performance**. Ultimately, we have shown the overall system performance for Spanish and English.

  - Spanish has been assessed with the VSME approach and the proposed linguistic contextual factors. The system is compared against a concatenative unit selection system. We can conclude that the Spanish HMM-based TTS system is preferred in terms of acceptability and intelligibility whereas, as expected, lacks of naturalness in comparison with the Co-TTS system. Acceptability is a measure of the overall quality of the system and it is useful in order to asses how good the system is for an application. In general, acceptability is higher for an HMM-based TTS system due to its quality stability in terms of naturalness and expressiveness. This measure can be related with the subjective result of the English system.

  - We have presented two different tests for English. Firstly, we have presented the performance of the APME and the VSME models. In this case, the real-time measure shows that the VSME approach is more efficient, though, it is less natural than the APME system. Secondly, a subjective test have been carried out among teenagers (i.e., game users) in order to assess the quality of the HMM-based TTS system and compare it against a Co-TTS system. We considered two types of sentences, in and out of domain. Generally, quality of TTS systems dramatically drop for out-of-domain sentences. In this experiment, this is exactly what happened with the Co-TTS system. However, the HMM-based TTS system was assessed very positively by the listeners in out-of-domain samples. In relation with the previously described acceptability, we can conclude that the HMM-based TTS is very acceptable for game applications. This is due to the fact that HMM systems have no abrupt quality variations and that expressiveness is fairly flat but constant.

Chapter 6

Conclusions and future work

6.1 General conclusions

This thesis has presented the use of HMMs for speech synthesis. HMMs has been introduced along with other synthesis techniques and their performance with respect to its quality and efficiency has been firstly discussed in Section 1.2. Originally, the main approach for speech synthesis was the creation of speech inventories using unit segmentation. This kind of system can usually produce a high-quality synthesis although special care and efforts need to be made on the unit selection and concatenation algorithms, the amount of data to be used and the target domain (open or limited). All these constraints clearly limit the use of this type of system and ultimately, they involve tuning several internal parameters (e.g., join cost weights). Eventually, a good open domain TTS system is a very hard task to achieve and, in order to guarantee a stable quality, most of the systems end up working as a limited domain application. On the other hand, HMMs has successfully emerged as the standard statistical TTS system producing synthetic speech directly from the HMM, being its spectral transition smoothness and stability one of its main characteristics.

HMM systems use a generic representation of speech based on parameters modelled in a probabilistic framework. As it has been described, it is a matter of fact that concatenative TTS systems can produce a better quality than the statistical approach, since the natural recorded segments of speech are concatenated. However, statistical synthesis using HMMs present a series of advantages which make them a very attractive solution. Basically, the possibility to model speech with a sequence of parameters make the system very suitable for efficient voice adaptation. Moreover, although synthetic speech is commonly felt as vocoded, the parameter generation algorithm makes it very reliable in terms of stability and also concatenation joins are no longer a problem. A summary of the details and objectives described on this thesis can be found in Table 6.1 which will be discussed throughout these conclusions.

Concatenative based on unit selection	HMM
Natural units	Statistical model
Very natural	Vocoded
Discontinuities	Smooth and stable
Cost function	HMM clustering
Large corpus	Small corpus
Fixed voice	Various voices

Table 6.1: A comparison between concatenative and HMM systems.

The use of HMMs for speech synthesis has not uniquely been related to producing parameters from the models but to other techniques such as unit segmentation and grapheme to phoneme conversion. Nevertheless, the main interest of this work is to describe the HMM-based speech synthesis systems that are usually based on a source-filter model approach. This is described in Chapter 2, briefly introduced in Section 1.2.4 and further detailed from Section 2.2 onwards. As it has been shown, the main training procedure is very similar to the one used in other applications such as speech recognition. One of the main differences is the use of continuous Gaussian distributions to simultaneously model vocal tract, mixed excitation and fundamental frequency parameters. In addition, a duration model is also needed. In order to synthesize speech, a parameter generation algorithm is used to produce sequences of parameters. This algorithm firstly estimates a duration for every state and then it produces the rest of the parameters. Source-filter model approach is conducted by filtering the reconstructed excitation signal (generated from excitation and F0 parameters) using the vocal-tract parameters.

According to Table 6.1, basic problems of the HMM-based TTS systems are over-smoothing and vocoded speech. On the one hand, different techniques have been described in Section 2.6 in order to alleviate the over-smoothing being Global Variance (GV) the most widely employed. Other techniques such as Minimum Generation Error (MGE) has also been described although GV stands as the most efficient model. In addition, a novel approach has been described in Section 2.6.3 where an external CBR-based F0 estimator is designed to work along with the HMM system during the synthesis stage. As a result, a more natural and expressive synthetic speech is obtained (see experiment in Section 5.3.1).

One of the main advantages of the HMM-based TTS systems is adaptation and this is described in Section 2.7. Using adaptation, a voice can be converted to a target voice even using a small amount of adaptation data (see experiment in Section 5.4.2). The main technique to modify models is a two stage process: first a linear regression stage (specifically, CMLLR where both mean and variances are transformed) and then a MAP stage. Linear regression works better when the amount of target data is limited whereas MAP has a better performance with a great amount of data.

By using a combined approach, robust transformations can be obtained as shown in the emotion adaptation experiment in Section 5.4.1.

During the construction of decision trees for HMM-based TTS systems (see Section 2.8), MDL criterion is preferred in front of ML approach. The main drawback of ML is the need of some external parameter to control the degree of clustering so the splitting process can be stopped at an optimal point. In contrast, MDL criterion is more suitable to be used for decision-tree based clustering since it sets a threshold in order to stop the splitting process. Decision trees use linguistic information in order to automatically cluster HMM states. This linguistic information is different from one language to another. With the purpose of adapting the conventional system to the Castilian Spanish language, a set of linguistic features has been proposed in Section 2.8.3. The quality of the Spanish system is assessed in experiments in Sections 5.3.2 and 5.5.1.

The second of the problems exposed in Table 6.1 is the quality of the vocoder. Some approaches to model the vocal-tract have been described in Section 3.3. On the one hand, a better spectrum envelope estimate based on the high-quality vocoder technique STRAIGHT has been used as it yield good results in other HMM-based TTS systems. This lets the system increase the number of parameters without suffering from the so-called F0 effect. Secondly, the buziness effect is also one of the main drawbacks of the HMM systems. This problem can be tackled by a proper design of a mixed excitation in the source-filter approach. Section 3.5 describes that a mixed excitation is a synthetic version of the prediction error obtained during the extraction of the vocal-tract coefficients (the so-called residual signal). This can be done by combining a filtered periodic pulse train and a filtered noise (voicing strengths, VSME in Section 3.5.3.2) or aperiodicity (APME in Section 3.5.4). The proposed design of the multiband mixed excitation used more parameters from the original residual signal than the conventional approach, and also, it incorporated complex amplitudes into the HMM. Complex amplitudes and voicing strengths improve the excitation representation in comparison with a system using only a basic excitation model. When comparing the performance of this excitation with the one produced with aperiodicity (see experiment in Section 5.3.3) it has been shown that the latter is a better representation of the multiband "voicedness". Therefore, aperiodicity yields a more natural synthetic speech although as we have seen in the subjective experiments, differences are not significantly high and a finer tuning of the voicing strength algorithm might improve the naturalness. In addition, the binary consideration during synthesis can also affect the quality. However, the aperiodicity is a very computationally expensive process and, in consequence, some applications can still prefer the VSME approach.

As we have seen in Section 3.5, the purpose of a mixed excitation is to introduce part of the missing information of the vocal-tract coefficients back to the filter. It is easy to see that, the better the vocal-tract modelling, the less information is left to the residual signal. In that respect, during the development of the experiments, we have observed that over-smoothing enhancement techniques (e.g., GV) can also reduce the buziness due to this reason.

Using the described HMM technique for synthesis in Chapter 2, an emotion adaptation applica-

tion has been proposed (see experiment in Section 5.4.1). As shown in this experiment, although emotional speech synthesis is a wide topic under research, we have verified through the Spanish HMM-based TTS design that synthesis using HMMs holds a set of advantages, such as the possibility to produce a stable synthesis using a reduced amount of recorded data for the target style. HMM-based speech synthesis has been shown to be a very reliable approach for an emotional generic purpose application.

Undoubtedly, HMM-based TTS systems can produce a good quality for open domain systems with a very low memory footprint. When using these properties appropriately (i.e, basically in applications where naturalness is not the main issue) the system performance is very high (see experiment in Section 5.5.2 for voice footprint information, real time performance measures and a naturalness assessment on a real scenario application). As a consequence of these results, when HMM samples are not presented along with the natural reference, listeners agree on assessing the naturalness as high. This is due to the fact that HMM can produce a very smooth and stable quality which is actually perceived positively by the listeners.

Apart from the two main trends being used to synthesize speech (i.e., concatenative or statistical) there is a third trend, the so-called hybrid systems. These systems are a combination of the previous approaches that aim to produce a more reliable synthesis where advantages of both systems are emphasized. Usually, hybrid systems are focused on synthetic speech quality but ideally they should extend their purpose to other advantages as well. On the one hand, one of the main advantages of HMMs is that they are trainable and adaptable. Furthermore, they guarantee stability in terms of intelligibility specially in open domain applications, even with small amounts of data. On the other hand, concatenative systems can produce a high-quality synthesis although its stability is a trade-off between the target domain, a robust unit selection system and the amount of data used.

As it is shown in the state-of-the-art in Chapter 4, hybrid systems are a relatively new area of research. So far, they have been categorized depending on the use of the natural units. In this work, these categories have been extended to two new types: concatenation and HMM driven. The former uses HMMs to create dynamic unit inventories. The novel system presented in this thesis, based on the second approach, uses a concatenative system in order to improve the naturalness of a state-of-the-art HMM-based TTS system based on parameter generation. Section 4.3 describes the use of the LMGE algorithm to blend natural and HMM generated sequences. The proposed hybrid system successfully smoothes the joins of the concatenative system and improves the quality of the HMM system baseline (see experiments in Section 5.3.4). It also reduces objectively the distortion of the synthetic speech with respect to the natural speech when compared with the GV system (see experiment in Section 5.3.4).

Although this hybrid system enhances mel-cepstral parameters using the LMGE algorithm, the rest of parameters (F0 and aperiodicity) remain unmodified. F0 is not enhanced because otherwise, there is no guarantee the overall utterance intonation could not be degraded since the concatenative model is not intended to produce a natural expressiveness but to select the optimal sequence of

natural units. As shown in the description of this hybrid system, the idea is to have a limited amount of target speaker's data. Under these conditions, F0 from natural units selected by the concatenative module could be very artificial. In this case, a good solution would be to use an external F0 estimator similar to the one described using the CBR-based CBR estimator (see Section 2.6.3). On the other hand, a justification for not enhancing aperiodicity is a bit more complicated. As shown in Section 3.5.4, although aperiodicity is a continuous representation of the length of the STRAIGHT spectrum, it is actually modelled in five subbands. After the statistical training process, the value of these parameters in an HMM is an average of the natural frequency representations. Unlike vocal-tract coefficients, this rough multiband averaging process produces undesirable artifacts during the blending process using the LMGE algorithm.

6.2 Future work

As we have described in this work, HMM-based speech synthesis system based on parameter generation basically lacks of naturalness and expressiveness. In order to increase the naturalness, efforts should be made on improving approaches such as context dependent Global Variance (a preliminary definition is described in Section D) or MGE. Although other speech production systems could be further analyzed continuing the work of other authors, results presented so far show that both mel-cepstral or LSP are the two best coefficients to be used along with STRAIGHT. Other vocal-tract coefficients could also be taken into consideration just reminding that for a new set of parameters, it is not only about the vocoding quality but also about their statistical manipulation properties. In addition, it is important to highlight that the number of parameters used in an HMM cannot be too elevated in order to preserve a good statistical modelling. In consequence, we cannot use the whole frequency representation for parameters such as aperiodicity.

Besides vocal-tract modelling, overall naturalness quality could also be improved by using a better mixed excitation model such as quantified residuals (e.g., CELP vocoder) or more complex models such as the one described in Section 3.5.6.

Not least important is expressiveness. In fact it seems to be a key aspect when improving the naturalness of HMM systems. It is not just the vocoder quality which degrades the perception of the users but also the flatness of the F0 and phone duration estimations. On top of the proposed approach presented in this thesis (using an external CBR-based F0 estimator), one of the solutions is to tune the decision-tree clustering by working on the linguistic contexts. Although this approach can produce good results, it is a very time consuming task and ultimately, the process should be repeated for every new voice. Another possibility would be to use weights along with the contextual factors in order to prioritize their significance. Alternatively, the clustering algorithm could also be improved (a finer MDL criterion) or techniques such as back-off trees could be revisited.

Thanks to the work presented by (Shannon and Byrne, 2009), we can conclude that the type of HMM is not a problem to solve since using Auto-Regressive (AR) HMM models does not result in a

better quality although they might be a better representation of speech. Nevertheless, an interesting research topic is the use of discrete HMMs for synthesis. Although the actual performance and properties of the system still seem unclear, the idea of not using continuous probability density functions may alleviate the over-smoothing effect.

In that respect, it is sensible to think that eventually, even having the best and most robust HMMs, an HMM synthesis based on parameter generation is going to produce a lower quality than the one obtained by a good sequence of concatenated natural units. As a matter of fact, by statistically modelling speech parameters, it is possible to obtain a set of advantages to the detriment of the quality. Hybrid systems appear as the candidates that balances this trade-off best.

More work should be done in the future is to investigate new hybrid approaches in order to improve flexibility and quality of synthetic speech. Results presented in this work for the HMM-driven hybrid system that this is a promising idea that could lead to create a high-quality speech synthesis system not based on concatenating natural units. In the future, it would be interesting to investigate alternatives to the LMGE algorithm presented in this work. Currently, the simplified assumptions of Equations 4.10 and 4.11 might be revised in order to obtain a better enhancing rule, specially for the variance.

The future of TTS systems is certainly difficult to predict. While some people think that a good unit selection algorithm and a good corpus design can lead to the best of the synthetic speech productions, other disagree on the assumption that this system is fairly crude and expensive to build. Neither is clear that HMM-based TTS systems can become a high quality system without losing part of its statistical advantages. As a consequence, it seems logical to think that hybrid systems will take the lead in the years to come so as to combine the advantages of concatenative and statistical approaches. In my opinion, this depends on how new approaches are going to be able to produce synthetic speech with the following constraints: reasonable quality, high expressiveness, high stability, low computational cost and low footprint.

Appendix A

Prosody in a Text-To-Speech system

Prosody is the phenomenon which carries expressive content in speech and involves intonation, rhythm and stress in phonetics and phonology. Intonation is the perceived fundamental frequency change, rhythm is the perceived structure in the timing of the speech signal and stress is perceived from local or short-term changes in fundamental frequency, amplitude and duration. In practise, prosody is described in terms of synthesis units as fundamental frequency (F0), durations and energy (Huang and Hon, 2001).

Prosody prediction in natural speech and dialogues is necessary for a synthesis system to offer a wide range of prosodic variability which can be described at an appropriate level of abstraction (Black and Campbell, 1995). If the TTS is based on a Co-TTS system, a correct estimation of prosody parameters is important for a good pitch synchronization and concatenation. On the one hand, some systems can use an external prosody estimation module (Pitrelli and Eide, 2003). On the other hand, the best prosody is part of the natural units themselves so prosody estimation is undertaken as part of the target and joint cost selection algorithm (Taylor, 2009). In the case of HMM-based TTS systems, F0 is crucial during speech generation and excitation signal construction. In general, most of the approaches using HMMs use a simultaneous modelling of parameters including F0 and duration (Yoshimura et al., 1999) (see also Section 3.4).

To predict the intonation of discourse segments from examples, a multi-level intonation system is necessary based on input labelled with high level discourse information. The model to predict the prosody can be achieved either by,

- **Knowledge approach (rule-driven)**. The rules for predicting the prosody must be obtained from the corpus using a set of descriptive labels which can be estimated using any suitable machine learning system. These rules decide the most suitable prosody pattern for the speech generation. The most common method for English is to use ToBI labels (Silverman et al., 1992). These labels are used in a system to transcribe intonation patterns and other aspects

of the prosody of English utterances.

- **Corpus approach**. The aim is to train a machine learning technique and then estimate prosody from the prediction of the closest example in the memory of cases through similarity of features. A good example of these kind of systems was presented in a Spanish expressive TTS system (Iriondo et al., 2006) where a Machine Learning strategy based on Case Based Reasoning (CBR) (see Section A.1) is used for prosody estimation.

Intonation models are classified according to the type of measurement of the fundamental frequency,

- **Acoustic level (direct)**. It represents changes of fundamental frequency, independently of linguistic knowledge. These models can accurately represent various fundamental frequency contours, but it is not straightforward to extract all the necessary features.
- **Perceptual level**. It is based on the assumption that every prosodic event is not perceived by the listener, so the fundamental frequency contour is represented by some of the lines which are simplified forms of the original fundamental frequency contour.
- **Linguistic level**. It describes the changes of intonation and generates the fundamental frequency from the linguistic descriptions.

A.1 A Case Based Reasoning system

This approach is appropriate to create prosody models from speech corpus to produce expressive speech as shown by (Iriondo et al., 2006). As depicted in Figure A.1, characteristics extracted from the text (i.e., a set of attribute-value pairs) are used to build prosody cases. Analysis of texts is carried out by the linguistic module (see SinLib library in Appendix E), an engine developed to Spanish text analysis. Hence, it is defined as a corpus oriented method for the quantitative modelling of prosody at acoustic level.

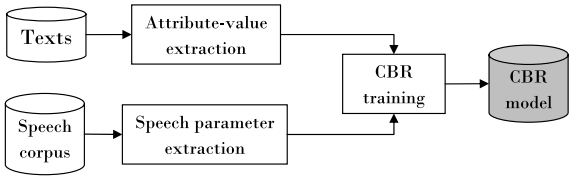


Figure A.1: CBR Training workflow.

Each utterance of the corpus is analysed in order to get a new set of cases (i.e., a group of attribute-value pairs). The goal is to obtain a prosody estimation from the memory of cases that best matches a known problem included in the model. For a given set of attribute-value pairs, CBR

looks for the best cases so as to retrieve prosody information from the most similar case it has in memory.

A.1.1 Attribute-value pair

The CBR system was designed to predict the prosody parameters per phoneme using a set of suitable features to characterize each unit. The set of attributes for F0, energy and duration are extracted from (Iriondo et al., 2007a) and shown in Table A.1.

Attribute extraction from each utterance is performed automatically and is labelled using IG and AG, words and syllables.

The main difference of the Spanish prosodic features with respect to other languages is the use of AG and IG. As defined by (Garrido, 1996), an AG is the sequence of syllables from one stressed syllable to the next stressed syllable. AG incorporates syllable influence and is related to speech rhythm. The type of AG is specified in Spanish as *agudo*, *plano*, *esdrújula* and *sobre-esdrújula* depending on the position of the accented syllable in the word. IG is defined as a coherent structure with no important prosodic breaks. IGs are usually extracted according to text breaks (e.g., commas) since pauses are natural delimiters. There are three IG types: interrogative, declarative and exclamative. Structure at this level is reached concatenating AGs.

F0	Energy	Duration
IG type	Current phoneme	Previous phoneme
Position of AG in IG	Stressed phoneme	Current phoneme
Position of the stressed syllable	Position of AG in IG	Next phoneme
Position of AG in the sentence	Position of phoneme in IG	Stressed phoneme
Number of syllables of AG	Position of phoneme in AG	Position of AG in IG
		Position of current phoneme in IG

Table A.1: Attribute-value pair for F0, energy and duration prediction using the CBR system.

A.1.2 Training and retrieval

The training of the CBR system involves two stages: selection and adaptation. In order to optimize the system, case reduction is carried out by grouping cases with exactly the same attributes and acoustic data is then averaged.

Once the memory of cases is created, the system looks for the most similar stored example. Duration is the first estimated parameter as it serves as the base for the estimation of the pitch and energy contours. Once the duration is predicted and normalized in the temporal axis, the system reconstruct the intonation by using a set of polynomial coefficients that will use the duration information to generate the F0 contour and energy.

A.2 Prosodic adjustments

In addition to the conventional techniques to estimate prosody in a Co-TTS system, an additional technique is to retrieve natural prosody (the so-called copy prosody). The goal is to reduce the use of signal processing (e.g., PSOLA) which can degrade the quality while maintaining the expressivity of the speaker. Note that this approach is used in the Spanish Co-TTS systems used in the experiments of Section 5.

The main problem of this technique arises when two concatenated units have very different F0 contours. In those cases, the F0 curve usually needs to be adjusted to avoid pitch discontinuities at the concatenation point. A Pitch Adjustment Module (PAM) was proposed by (Alías et al., 2005) for that purpose.

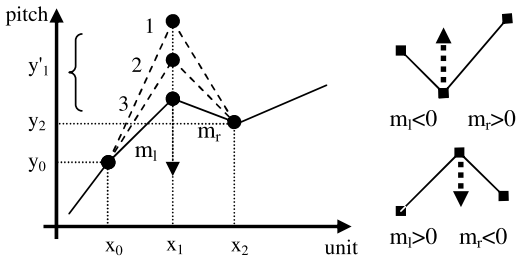


Figure A.2: Iterative pitch curve peak smoothing at concatenation point. The adjustment depends on the point concavity.

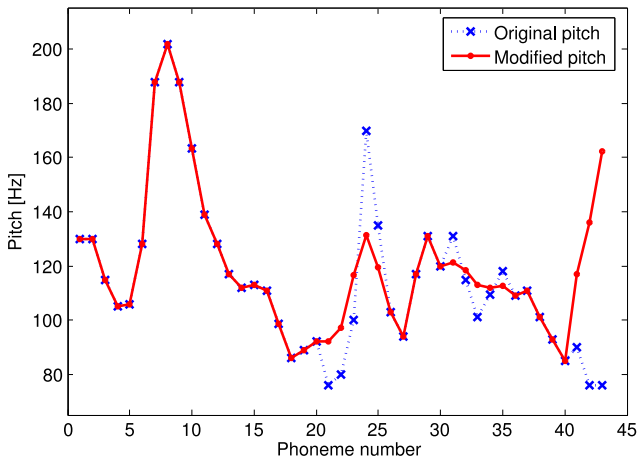


Figure A.3: Adjustment and smoothing of the pitch curve at a point of concatenation (phoneme 23), for pause insertion (phoneme 31).

Basically, a progressive smoothing of the pitch curve is carried out by means of an effective

iterative procedure (see Figure A.2). This pitch adjustment is conducted on n units around the joint (x_1). The iterative process continues until the difference between the left and the right slopes of the pitch curve in that point is lower than a threshold. For unit x_1, the new ordinate value y'_1 of the pitch curve is fixed depending on the concavity (obtained from the left m_l and right m_r gradients) of each n-point, increasing or decreasing its value until the discontinuity is smoothed. The effect of the F0 contour smoothing can be observed in Figure A.3, where a concatenation point has been smoothed using PAM and $n = 3$.

Appendix B

MGE definition

The following appendix is a detailed description of the MGE algorithm. Firstly, the conventional parameter generation algorithm is briefly reviewed. Secondly, a complete mathematical demonstration is described for the conventional and the reduced MGE.

B.1 Summary of the parameter generation algorithm

As described in Section 2.5.1 and assuming that each HMM uses a single multivariate Gaussian distribution, the observation feature vector ($\mathbf{O}$) for training purposes contains static and dynamic features and it is defined as:

$$\begin{aligned} \mathbf{O} &= \left[\mathbf{C}^{\mathrm{T}}, \Delta\mathbf{C}^{\mathrm{T}}, \Delta^2\mathbf{C}^{\mathrm{T}}\right]^{\mathrm{T}} = \mathbf{W}\mathbf{C} \\ \mathbf{C} &= \left[\mathbf{c}_1^{\mathrm{T}}, \mathbf{c}_2^{\mathrm{T}}, \ldots, \mathbf{c}_T^{\mathrm{T}}\right]^{\mathrm{T}} \\ \mathbf{c}_t &= \left[c_t(1), c_t(2), \ldots, c_T(L)\right]^{\mathrm{T}} \end{aligned}$$

where L is the dimension of the Gaussian, T is the total number of frames in the utterance and $\mathbf{W}$ is a matrix to obtain the dynamic features. This matrix is of dimension $3TL \times TL$ and contains dynamic window coefficients (Tokuda et al., 2000):

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_T]^{\mathrm{T}} \tag{B.1}$$

$$\mathbf{w}_t = [\mathbf{w}_t^{(0)}, \mathbf{w}_t^{(1)}, \mathbf{w}_t^{(2)}] \tag{B.2}$$

where $\mathbf{w}_t^{(d)}$ is defined in Equation 2.41.

In this case, parameters generated by the HMM are obtained using the following equation:

$$\hat{\mathbf{C}} = (\underbrace{\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{W}}_{\mathbf{R}\ (TL\times TL)})^{-1}\underbrace{\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}}_{\mathbf{r}\ (TL\times 1)} = \mathbf{R}^{-1}\mathbf{r}, \tag{B.3}$$

Equation B.3 is obtained by maximizing the probability $P(\mathbf{W}\hat{\mathbf{C}}|\mathbf{Q},\lambda)$, that is, the probability of generating observations with respect to a fixed states sequence $\mathbf{Q} = \{q_1, q_2, \ldots, q_T\}$ where q_t represents being in state q_t at frame t.

$$P(\mathbf{O}|\lambda) = \sum_{\forall \mathbf{Q}} P(\mathbf{W}\hat{\mathbf{C}}|\mathbf{Q},\lambda)$$

The sequence of means and variances from the HMM are:

$$\begin{aligned}\boldsymbol{\Gamma} &= [\boldsymbol{\Gamma}_{q_1}^{\mathrm{T}}, \boldsymbol{\Gamma}_{q_2}^{\mathrm{T}}, \ldots, \boldsymbol{\Gamma}_{q_T}^{\mathrm{T}}]^{\mathrm{T}}\\ \boldsymbol{\Sigma} &= \mathrm{diag}[\boldsymbol{\Sigma}_{q_1}, \boldsymbol{\Sigma}_{q_2}, \ldots, \boldsymbol{\Sigma}_{q_T}]\end{aligned}$$

where $\boldsymbol{\Gamma}$ is a $3TL \times 1$ vector of means and $\boldsymbol{\Sigma}$ is the $3TL \times 3TL$ diagonal matrix of covariances. The mean vector and the covariance matrix contain information of the associated state q_t at frame t:

$$\begin{aligned}\boldsymbol{\Gamma}_{q_t} &= \left[\boldsymbol{\mu}_{q_t}, \Delta\boldsymbol{\mu}_{q_t}, \Delta^2\boldsymbol{\mu}_{q_t}\right]^{\mathbf{T}}\\ \boldsymbol{\mu}_{q_t} &= [\mu_{q_t}(1), \mu_{q_t}(2), \ldots, \mu_{q_t}(L)]^{\mathrm{T}}\\ \boldsymbol{\Sigma}_{q_t} &= \mathrm{diag}\left[\boldsymbol{\sigma}_{q_t}^2, \Delta\boldsymbol{\sigma}_{q_t}^2, \Delta^2\boldsymbol{\sigma}_{q_t}^2\right]\\ \boldsymbol{\sigma}_{q_t}^2 &= \mathrm{diag}\left[\sigma_{q_t}^2(1), \sigma_{q_t}^2(2), \ldots, \sigma_{q_t}^2(L)\right]\end{aligned}$$

B.2 MGE: theoretical definition

MGE (see Section 2.6.2) is based on the Generalized Probabilistic Descent (GPD) algorithm to iteratively update a statistical model using N examples. For example n the updated model becomes:

$$\lambda(n+1) = \lambda(n) - \varepsilon(n)\cdot\mathbf{S}_n\cdot\nabla\ell(\mathbf{C}_n,\lambda)|_{\lambda=\lambda(n)} \tag{B.4}$$

where λ represents the HMM parameters, n is the current example (i.e., in the context of HMM-based TTS system training, n is the current utterance), $\varepsilon(n)$ is the learning factor controlling the speed and accuracy of the convergence process, $\mathbf{S}_n$ is a definite positive matrix (which in practise can be set to the identity matrix) and $\ell(\mathbf{C}_n,\lambda)$ is the cost function defining the generation error.

Usually, the Euclidean Squared distance is used,

$$\ell(\mathbf{C},\lambda) = D(\hat{\mathbf{C}},\mathbf{C}) = \left\|\hat{\mathbf{C}}-\mathbf{C}\right\|^2 \tag{B.5}$$

By using the cost function of Equation B.5 into Equation B.4, we can define the corresponding updating criterion as,

$$\lambda(n+1) = \lambda(n) - \varepsilon(n) \cdot \left. \frac{\partial \ell(\mathbf{C}_n, \lambda)}{\partial \lambda} \right|_{\lambda=\lambda(n)} \tag{B.6}$$

The model during example n is updated using the first derivative of the error with respect to the model.

According to the previous equations, we can derive the formulas to update the mean and the variance of the visited models for the t-th frame and j-th dimension as [1],

$$\frac{\partial \ell(\mathbf{C}, \lambda)}{\partial \mu_{tj}} = 2 \left(\hat{\mathbf{C}} - \mathbf{C} \right)^{\mathrm{T}} \frac{\partial \hat{\mathbf{C}}}{\partial \mu_{tj}}$$

and,

$$\frac{\partial \ell(\mathbf{C}, \lambda)}{\partial v_{tj}} = 2 \left(\hat{\mathbf{C}} - \mathbf{C} \right)^{\mathrm{T}} \frac{\partial \hat{\mathbf{C}}}{\partial v_{tj}}$$

where $\mu_{tj} = \mu_{q_t}(j)$ and v_{tj} is the j-th dimension at the t-th frame of the mean and the inverse variance ($v_{tj} = {}^{1}\!/_{\sigma_{tj}^2}$, $\sigma_{tj}^2 = \sigma_{q_t}^2(j)$), respectively.

For the mean, it is straightforward to derive the following solution:

$$\frac{\partial \hat{\mathbf{C}}}{\partial \mu_{tj}} = \left(\mathbf{W}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \mathbf{W} \right)^{-1} \mathbf{W}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\psi}^{(tj)} = \mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\psi}^{(tj)} \tag{B.7}$$

where $\boldsymbol{\psi}^{(tj)}$ is the elementary vector, one at $(3(t-1)L + j) \in [1, 3TL]$ and zero elsewhere.

For the variance, it yields:

$$\begin{aligned} \frac{\partial \hat{\mathbf{C}}}{\partial v_{tj}} &= \frac{\partial \mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}}{\partial v_{tj}} \boldsymbol{\Gamma} = \left\{ \underbrace{\frac{\partial (\mathbf{YZ})}{\partial \mathbf{X}} = \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} \mathbf{Z} + \mathbf{Y} \frac{\partial \mathbf{Z}}{\partial \mathbf{X}}}_{\text{Product rule}} \right\} \\ &= \left(\frac{\partial \mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}}}{\partial v_{tj}} \boldsymbol{\Sigma}^{-1} + \mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}} \boldsymbol{\Psi}^{(tj)} \right) \boldsymbol{\Gamma} \end{aligned}$$

where $\boldsymbol{\Psi}^{(tj)}$ is the elementary matrix with dimension $3TL \times 3TL$, one at $(3(t-1)L+j, 3(t-1)L+j)$ and zero elsewhere. As the derivative of matrix $\mathbf{R}^{-1}$ is

$$\frac{\partial \mathbf{R}^{-1}}{\partial v_{tj}} = -\mathbf{R}^{-1} \mathbf{W}^{\mathrm{T}} \boldsymbol{\Psi}^{(tj)} \mathbf{W} \mathbf{R}^{-1}$$

[1] The derivative of the norm is:

$$\frac{\partial \left\| \hat{\mathbf{C}} - \mathbf{C} \right\|^2}{\partial \lambda} = \frac{\partial \left\| \left(\hat{\mathbf{C}} - \mathbf{C} \right)^{\mathrm{T}} \left(\hat{\mathbf{C}} - \mathbf{C} \right) \right\|}{\partial \lambda} = 2 \left(\hat{\mathbf{C}} - \mathbf{C} \right)^{\mathrm{T}} \frac{\partial \hat{\mathbf{C}}}{\partial \lambda}$$

then

$$\begin{aligned}\frac{\partial \hat{\mathbf{C}}}{\partial v_{tj}} &= -\mathbf{R}^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Psi}^{(tj)}\mathbf{W}\mathbf{R}^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma} + \mathbf{R}^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Psi}^{(tj)}\boldsymbol{\Gamma} \\ &= \mathbf{R}^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Psi}^{(tj)}\left(-\mathbf{W}\mathbf{R}^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma} + \boldsymbol{\Gamma}\right)\end{aligned}$$

By using Equation B.3, previous derivative can be simplified as

$$\frac{\partial \hat{\mathbf{C}}}{\partial v_{tj}} = \mathbf{R}^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Psi}^{(tj)}\left(-\mathbf{W}\hat{\mathbf{C}} + \boldsymbol{\Gamma}\right) \tag{B.8}$$

Now, using Equation B.7 and B.8, the updating rule in Equation B.6 can be formulated as

$$\mu_{tj}(n+1) = \mu_{tj}(n) - 2\varepsilon(n)\left(\hat{\mathbf{C}} - \mathbf{C}\right)^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\psi}^{(tj)} \tag{B.9}$$

$$v_{tj}(n+1) = v_{tj}(n) - 2\varepsilon(n)\left(\hat{\mathbf{C}} - \mathbf{C}\right)^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Psi}^{(tj)}\left(-\mathbf{W}\hat{\mathbf{C}} + \boldsymbol{\Gamma}\right) \tag{B.10}$$

B.3 Reduced MGE

The parameter updating rules in Equation B.7 and B.8 are very time consuming due to the sample-by-sample process and the calculation of $\mathbf{R}^{-1}$. For this to be alleviated, the following updating rule is proposed by (Wu et al., 2006):

$$\lambda^{(new)} = \lambda^{(old)} - \varepsilon \sum_{n=1}^{N}\sum_{t=1}^{T_n}\varphi(n,t)\left.\frac{\partial \ell(\mathbf{C}_n,\lambda)}{\partial \lambda}\right|_{\lambda=\lambda_{old}}$$

where T_n is the total number of frames in utterance n and:

$$\varphi(n,t) = \begin{cases} 1 & \text{if model } \lambda \text{ is visited in utterance } n \text{ and frame } t \\ 0 & \text{otherwise} \end{cases}$$

In this case, all samples where model λ is visited are simultaneously used to update this model. In order to reduce the computational cost, the following approximation is taken into account:

$$\mathbf{W}\mathbf{W}^{\mathrm{T}} \approx \mathbf{I} \tag{B.11}$$

where $\mathbf{I}$ is the identity matrix. Applying this simplification to the partial derivative of the mean we

obtain:

$$\begin{aligned}\frac{\partial \ell(\mathbf{C}_n, \lambda)}{\partial \mu_{ij}} &= 2\left(\hat{\mathbf{C}}_n - \mathbf{C}_n\right)^{\mathrm{T}} \mathbf{R}_n^{-1} \mathbf{W}_n^{\mathrm{T}} \boldsymbol{\Sigma}_n^{-1} \boldsymbol{\psi}_n^{(ij)} \\ &= 2\left(\hat{\mathbf{C}}_n - \mathbf{C}_n\right)^{\mathrm{T}} \mathbf{R}_n^{-1} \mathbf{W}_n^{\mathrm{T}} \boldsymbol{\Sigma}_n^{-1} \underbrace{\mathbf{W}_n \mathbf{W}_n^{\mathrm{T}}}_{\approx \mathbf{I}} \boldsymbol{\psi}_n^{(ij)} \\ &= 2\left(\hat{\mathbf{C}}_n - \mathbf{C}_n\right)^{\mathrm{T}} \mathbf{W}_n^{\mathrm{T}} \boldsymbol{\psi}_n^{(ij)} \\ &= 2\left(\hat{\mathbf{O}}_n - \mathbf{O}_n\right)^{\mathrm{T}} \boldsymbol{\psi}_n^{(ij)} \end{aligned} \tag{B.12}$$

where μ_{ij} is the j-th coefficient of the mean of model i. Note that in this case, the mean is not straightforwardly related to frame t since we are using N utterances simultaneously. Applying previous approximation to Equation B.7 yields:

$$\mu_{ij}^{(new)} = \mu_{ij}^{(old)} - 2\varepsilon \sum_{n=1}^{N} \sum_{t=1}^{T_n} \varphi_i(n,t) \left(\hat{o}_{nt}(j) - o_{nt}(j)\right) \tag{B.13}$$

where $\hat{o}_{nt}(j)$ and $o_{nt}(j)$ are the j-th dimension of observation from utterance n at frame t generated by the HMM and natural reference, respectively. Moreover, $\mu_{ij}^{(new)}$ and $\mu_{ij}^{(old)}$ are j-th coefficient of the new and the old mean of model i, respectively.

For the definition of the reduced equation of the variance, the following approximation is made: matrix $\mathbf{W}$ is redefined for each dynamic feature individually. Therefore, for each feature $d \in \{0, 1, 2\}$, $\mathbf{W}^{(d)}$ is a square matrix with dimensions $TL \times TL$:

$$\mathbf{W}^{(d)} = \left[\mathbf{w}_1^{(d)}, \ldots, \mathbf{w}_T^{(d)}\right]^{\mathrm{T}} \tag{B.14}$$

Applying the simplification defined in Equation B.11 to the partial derivative of the variance yields:

$$\begin{aligned}\frac{\partial \ell(\mathbf{C}_n, \lambda)}{\partial v_{tj}} &= \left(\hat{\mathbf{C}}_n - \mathbf{C}_n\right)^{\mathrm{T}} \mathbf{R}_n^{(d)-1} \mathbf{W}_n^{(d)\mathrm{T}} \boldsymbol{\Psi}_n^{(tj)} \left(-\mathbf{W}_n^{(d)} \hat{\mathbf{C}}_n + \boldsymbol{\Gamma}_n^{(d)}\right) \\ &= \left(\hat{\mathbf{C}}_n - \mathbf{C}_n\right)^{\mathrm{T}} \mathbf{W}_n^{(d)\mathrm{T}} \boldsymbol{\Sigma}_n^{(d)} \boldsymbol{\Psi}_n^{(ij)} \left(-\mathbf{W}_n^{(d)} \mathbf{C}_n + \boldsymbol{\Gamma}_n^{(d)}\right) \end{aligned} \tag{B.15}$$

In Equation B.15, $j \in [1, L]$ for each d. Note that, although considering $\mathbf{W}^{(d)}$ as a square matrix, parameters $\hat{\mathbf{C}}$ are produced using Equation B.3 whereas Equation B.15 is only used as an updating rule. As one can see, matrix $\mathbf{W}^{(d)}$ has been used for the sake of the mathematical demonstration. However, for implementation purposes, d is indistinct and in consequence, this equation can be rewritten as:

$$v_{ij}^{(new)} = v_{ij}^{(old)} - \frac{2\varepsilon}{v_{ij}^{(old)}} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \varphi_i(n,t) \left(\hat{o}_{nt}(j) - o_{nt}(j)\right) \left(\mu_{ij}^{(old)} - o_{nt}(j)\right) \tag{B.16}$$

The step size is set in order to normalize the total number of observed frames:

$$\varepsilon = \frac{1}{2N_i}$$

where N_i is the total number of frames where model λ_i is used:

$$N_i = \sum_{n=1}^{N} \sum_{t=1}^{T_n} \varphi_i(n,t) \tag{B.17}$$

Appendix C

Distance between two HMMs

C.1 Distance between distributions

The distance between two HMMs λ_1 and λ_2 is defined as:

$$D\left(\lambda_1, \lambda_2\right) = \frac{1}{T}\left(\log\left(p(X^{(2)}|\lambda_1\right) - \log\left(p(X^{(2)}|\lambda_2\right)\right) \tag{C.1}$$

where $X^{(2)} = \{X_1, X_2, \cdots, X_T\}$ is a sequence of observations generated by model λ_2. Basically is a measure of how well model λ_1 matches observations generated by λ_2, relative to how well model λ_2 matches observations generated by itself. As this distance is non-symmetric, the symmetric version is:

$$D_s\left(\lambda_1, \lambda_2\right) = \frac{D\left(\lambda_1, \lambda_2\right) + D\left(\lambda_2, \lambda_1\right)}{2} \tag{C.2}$$

Another way of getting the distance between two HMMs is by using a Bhattacharyya distance. This distance is a suitable similarity measure for this task as it provides a measure of the overlap between two probability density functions. When dealing with two multivariate Gaussian distribution models, the distance is defined as:

$$D_B\left(p_1, p_2\right) = \frac{1}{8}\left(\boldsymbol{\mu}_2 + \boldsymbol{\mu}_1\right)^\top \left(\frac{\boldsymbol{C}_1 + \boldsymbol{C}_2}{2}\right)^{-1} \left(\boldsymbol{\mu}_2 + \boldsymbol{\mu}_1\right) + \frac{1}{2}\ln\frac{\left|\frac{\boldsymbol{C}_1+\boldsymbol{C}_2}{2}\right|}{\sqrt{|\boldsymbol{C}_1| \cdot |\boldsymbol{C}_2|}} \tag{C.3}$$

where $|\cdot|$ denotes determinant, $\boldsymbol{\mu}_i$ is is the mean vector and $\boldsymbol{C}_i$ is the covariance matrix of class i. The first term gives the class separability due to the difference between class means, while the second term gives the class separability due to the difference between class covariance matrices. The process is the following:

- For each utterance, use the data obtained for the training (i.e. NxL_i matrix where N is the

order of the observation and L_i is the number of observations of utterance i).

- Use the state alignment information to extract all the observation vectors of dimension Nx1 that belong to each state of the HMM.
- Estimate the centroid of each state. If we have a n-states HMM, we will have: $\boldsymbol{\mu}_h = \{\mu_1, \mu_2, \cdots, \mu_n\}$ and $\boldsymbol{\sigma}_h = \{\sigma_1, \sigma_2, \cdots, \sigma_n\}$ representing the mean and the variances of the HMM of phoneme h.
- Compute the D_B of two HMM using $\boldsymbol{\mu}_{h_i}$ and $\boldsymbol{\sigma}_{h_i}$ where $i = \{1, 2\}$.

C.2 Simplified distance

As the models under test share the transition matrix, the distance can be simplified as,

$$\mathrm{D}(\lambda_1, \lambda_2) \leq \sum_{i=1}^{S} \frac{1}{1 - a_{ii}} \left[\mathrm{D}\left(\mathcal{N}_{1_i}, \mathcal{N}_{2_i}\right) + \mathrm{D}\left(\mathcal{N}_{2_i}, \mathcal{N}_{1_i}\right)\right] \tag{C.4}$$

where for each state the Kullback-Leibler (KL) divergence between two L-dimensional single mixture Gaussian distributions can be calculated as,

$$\mathrm{D}\left(\mathcal{N}_{1_i}, \mathcal{N}_{2_i}\right) = \frac{1}{2} \log\left(\frac{\mathbf{U}_{2_i}}{\mathbf{U}_{1_i}}\right) - \frac{L}{2} + \frac{1}{2} Tr\left(\mathbf{U}_{2_i}^{-1} \mathbf{U}_{1_i}\right) + \frac{1}{2} \left(\boldsymbol{\mu}_{2_i} - \boldsymbol{\mu}_{1_i}\right)^{\mathrm{T}} \mathbf{U}_{2_i}^{-1} \left(\boldsymbol{\mu}_{2_i} - \boldsymbol{\mu}_{1_i}\right) \tag{C.5}$$

where $Tr(\cdot)$ is the trace of a matrix.

For a symmetric KL divergence:

$$\mathrm{D}\left(\lambda_1, \lambda_2\right) = \frac{1}{2} \left(\mathrm{D}\left(\mathcal{N}_{1_i}, \mathcal{N}_{2_i}\right) + \mathrm{D}\left(\mathcal{N}_{2_i}, \mathcal{N}_{1_i}\right)\right)$$

Appendix D

Context dependent GV

As described in Section 2.6.1, one of the most used method to alleviate the problem of over-smoothing is the so-called Global Variance. As a matter of fact, this global model can certainly introduce variability to the vocal tract coefficients and enhance the speech. It reduces not only the muffleness but also the buziness. In the following Section, Context Dependent GV (CGV) is introduced as a preliminary proposal to extend the conventional GV technique.

One of the main problems though, is the simplicity of this GV model. Each voice will have a unique global indicator of how the variance should be. This produces some unnaturalness and more importantly, some instabilities such as clipping and glitches during the steepest descent stage. In order to avoid these problems, a logical solution is to design a more accurate model of GV. As described in Section 2.8, the use of context dependent information and decision tree-based clustering helps to create a model that can be used with unseen units. In a similar way, a context dependent GV (CGV) is built.

The CGV is a continuation of the GV model which is proposed in order to improve and control the effect of the GV in the synthesized speech. The details given in this Section describe the preliminary design related to the proposal of the latest versions of HTS (HTS, a).

A GV model is built using natural speech and then the generated parameters are modified in order to match this target variance. In this contextual GV there is not a single GV but a global variance model attached to every contextual unit. In this case, CGV for unit u in an utterance is described as follows:

$$v_u(l) = \frac{1}{T_u} \sum_{t=t_u^{(i)}}^{t_u^{(e)}} \left(\hat{\mathbf{c}}_t(l) - \mu_{\hat{\mathbf{c}}_u(l)}\right) \tag{D.1}$$

where $t_u^{(i)}$ and $t_u^{(e)}$ are the initial and final frames of the current unit u and $T_u = t_u^{(e)} - t_u^{(i)}$ is the total duration of this unit. The mean $\mu_{\hat{\mathbf{c}}_u(l)}$ is computed as in Equation 2.48 taking into account

T_u. From now on, we will refer to $\hat{\mathbf{c}}_u(l)$ as the generated parameter sequence for unit u.

Let's describe now the steepest descent algorithm. In the conventional GV the goal was to approach the likelihood of each coefficient within an utterance. In the CGV, the likelihood is adapted to each unit. In fact, although the change of philosophy, the likelihood can be computed in the same way but an interpolation process is introduced in order to smooth the transition between units. The first step of the steepest descent stage was defined in Equation 2.52. In this case, the initialization of each unit is defined as follows:

$$\hat{\mathbf{c}}_t^{(0)}(l) = r_i \left(\hat{\mathbf{c}}_t(l) - \mu_{i,\hat{\mathbf{c}}_u(l)}\right) \tag{D.2}$$

for $t \in [t_u^{(i)}, t_u^{(e)}]$ where the interpolated r_i and $\mu_{i,\hat{\mathbf{c}}_u(l)}$ are defined using the current (c) and the next (n) values for $i \in [0, T_u)$. The ratio for each frame is incremented a successive step size from the current ratio:

$$r_i = r_c + i\frac{r_n - r_c}{T_u} \tag{D.3}$$

and similarly, the mean $\mu_{i,\hat{\mathbf{c}}_t(l)}$ becomes:

$$\mu_{i,\hat{\mathbf{c}}_u(l)} = \mu_{c,\hat{\mathbf{c}}_u(l)} + i\frac{\mu_{n,\hat{\mathbf{c}}_u(l)} - \mu_{c,\hat{\mathbf{c}}_u(l)}}{T_u} \tag{D.4}$$

According to the definitions, for each interpolation step, r_c and r_n are the current and next ratios and $\mu_{c,\hat{\mathbf{c}}_u(l)}$ and $\mu_{n,\hat{\mathbf{c}}_u(l)}$ are the current and next mean of the generated parameters for unit u, respectively.

For the rest of steps, the process is also very similar to the conventional GV. The variance and the mean of the synthesized utterance are also interpolated for each unit. In this case, only the first derivative has been applied for two reasons. On the one hand, it shows more stability and on the other hand, since CGV implies a more accurate value of the GV within the current unit itself, the first derivative is accurate enough.

In this case, the likelihood is expressed using the following Equation derived from Equation 2.60:

$$\frac{\partial \mathbf{v}(\hat{\mathbf{C}})}{\partial \hat{\mathbf{C}}} = -\frac{2\sigma_{v(l)}^2}{T_u}\left(\sigma_{i,\hat{\mathbf{c}}(l)}^2 - \mu_{v(l)}\right)\left(\sum_{t=t_u^{(i)}}^{t_u^{(e)}}\left(\hat{\mathbf{c}}_t(l) - \mu_{i,\hat{\mathbf{c}}_u(l)}\right)\right) \tag{D.5}$$

where

$$\sigma_{i,\hat{\mathbf{c}}(l)}^2 = \sigma_{c,\hat{\mathbf{c}}(l)}^2 + i\frac{\sigma_{n,\hat{\mathbf{c}}(l)}^2 - \sigma_{c,\hat{\mathbf{c}}(l)}^2}{T_u} \tag{D.6}$$

$$\sigma_{c,\hat{\mathbf{c}}(l)}^2 = \frac{1}{T_u}\sum_{t=t_u^{(i)}}^{t_u^{(e)}}\left(\hat{\mathbf{c}}_t(l) - \mu_{i,\hat{\mathbf{c}}_u(l)}\right)$$

is the interpolated variance of the CGV model.

Appendix E

Contribution to specific tools

During this development of this thesis, research projects have provided a good environment for the development, upgrading and improvement of many tools. The most important ones are presented in the following sections.

E.1 RST (Research Speech Toolkit)

This graphical interface is oriented to automate the processes of corpus creation through the following modules:

- The "transcription" part performs the automatic grapheme to phoneme conversion. It allows to work with Catalan, Spanish and English languages.

- The "Greedy" module is oriented to sentences selection. From a set of initial texts, the most reduced subset of them is selected in order to cover the input requisites. Algorithm is performed in five stages:

    1. Initial statge. Original sentences F, whole requisites R, selected sentences S which is initially empty and selected units (each requisite) U.

    2. Sentence score calculation P_i. Considering a sentence i (F_i), its score is P_i. Sentence selection is carried out on short (see equation E.1) or long sentences (see equation E.2).

$$P_i = \frac{UF_i \cdot mUF_i}{\text{length}(F_i)} \tag{E.1}$$

$$P_i = UF_i \cdot mUF_i \cdot \text{length}(F_i) \tag{E.2}$$

where UF_i is the number of different units that exist in the sentence i (no covered requisites), mUF_i is the arithmetic mean of the number of selected units for each type and length(F_i) is the number of units per sentence.

The sentence score is initialized to a negative value. During the analysis process it is updated depending on the requisites. At the end, a sentence can be zero scored if either no units appears in the sentence or if the sentence length is out of bounds.

3. Sentence selection. Sentence with the largest score is selected. Different fields are updated: sentence F_i is added to S, sentence F_i is deleted from F and units of sentence F_i are added to U.
4. Checking of requisites accomplishment.
5. Stages 2, 3 and 4 are repeated until complete all requisites.

E.2 Service manager for the TTS server

For the development of the Virtual Weather Project it was necessary to create two different interfaces for the TTS system: a shell command for batch processing a service controller to manage the Microsoft Server for the TTS core. The aim of this interface was to easily control the options of the TTS system, stop/start/restart the engine and verify the status of the server.

E.3 SinLib

The SinLib library for grapheme to phoneme conversion is a rule-based system used to preprocess the text for the TTS system and to get the unique phoneme representation of a word (e.g. "hola" as "_ola_"). During the former function, the non-text elements are converted into its representation (e.g., numbers). Basically, it handles:

- Special symbols (e.g. "?,!,:").
- Numbers. It converts a number either into a complete description (e.g. "21" as "twenty one") or into a digit-by-digit representation (e.g., "21" as "two, one").
- Full context labels for the Spanish HMM-based TTS system as described in Section 2.8.3.

The latter functionality codifies the text in the SAMPA standard (Llisterri and Mario, 1993). Rules are specific for each language and must be described in a specific programming language (see Table E.1) using the graphical interface in Figure E.1).

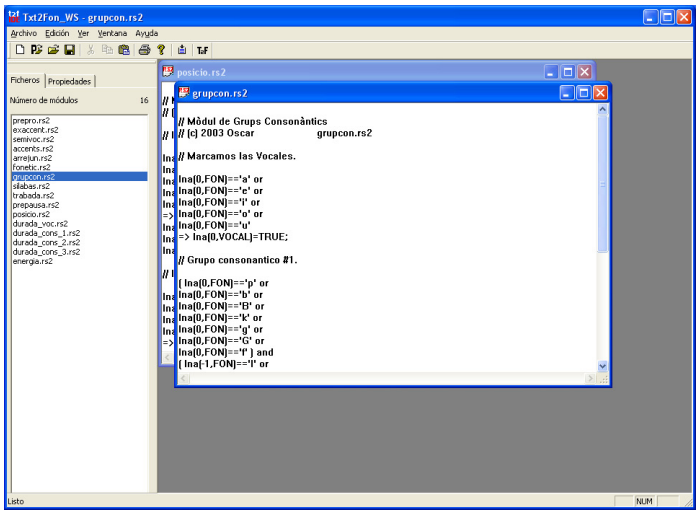


Figure E.1: Graphical interface for the SinLib programming language.

```
// -------------------------------------
// Regles de la a
// -------------------------------------
Ina(0,GRF)=='a' or
Ina(0,GRF)=='A' or
Ina(0,GRF)=='á' or
Ina(0,GRF)=='Á'
=>Ina(0,FON)='a';
```

Table E.1: A simple example of the programming code for the rule of the “a” vowel.

E.4 Corpus Tester

The robust labelling of the corpus for a Co-TTS is essential in order to guarantee the non-existence of problems during the synthesis stage. Due to the size of these kind of corpus, it was developed a tool that allows to detect the label inconsistencies and mistakes of the corpus in order to automatically check the reliability of the automatic labelled data.

This tool controls the corpus structure, the grapheme to phoneme conversion, the segmentation labels (phoneme time boundaries) and the pitch marks. The main problems checked are for excessively closed segmentation marks (a threshold is defined, e.g. 20 ms), pitch discontinuities (e.g. local peridiodicity of $\pm 30\%$) and controls that the phonemes presents values of F0 out of the stabilished rank (e.g. $[50, 550]$ Hz). All this information is stored in an output file so the expert supervisor uses to classify the files to mend. This avoids to manually check all the corpus.

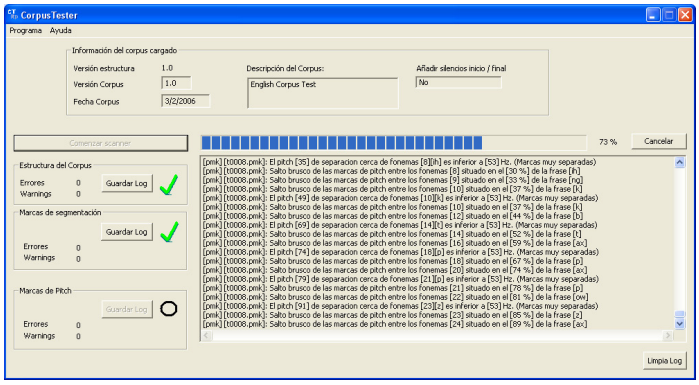


Figure E.2: Graphical interface of the Corpus Tester. There are three informations: structure of the corpus, time segmentation and pitch marks.

E.5 Speech Processing Interface (SPI) v2.0

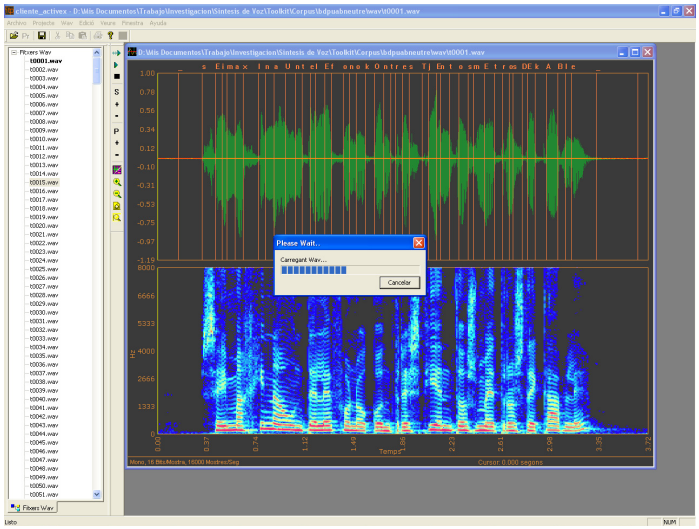


Figure E.3: Graphical interface SPI. Different informations are showed: project files and time segmentation.

The original SPI v1.0 is an application that involves different algorithms and processes related with the corpus labelling oriented to Co-TTS. The upgrade E.3 is based on the original interface and introduces some new features, such as, project management (i.e. each corpus is seen as a project of n files), improved processes (e.g. faster time segmentation with integrated HTK Library), XML based corpus files as defined in the SALERO project (Socoró et al., 2007)).

E.6 Multimodal system

One of the supervised projects as a PFC developed a multimodal interface with the following modules: a TTS based on the US-TTS, a speech recognition system to control a domotic house based on ATK (Young, 2007) and a visual interface to detect the presence of a face.

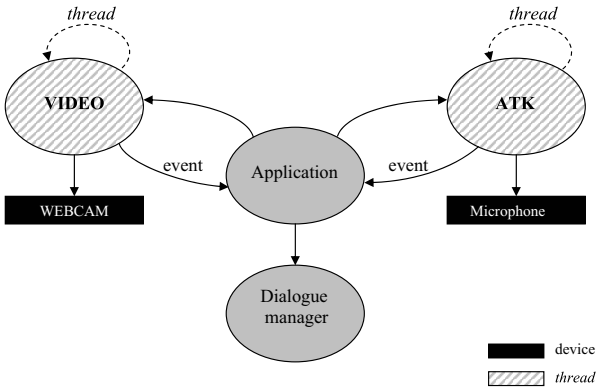


Figure E.4: Multimodal application: The video and the output perform events to the core system.

The application has two main threads (Figure E.4). On the one hand, the voice process is based on the ATK recognizer that generates a set of event whenever the buffer of the input registers some information. On the other hand, the image process includes two main clocks: one is used before the dialogue has started and the other is active while the dialogue is being performed. The aim of the latter clock is to improve the fiability of the ASR module because it guarantee that the person is still speaking and looking to the webcam interface.

E.7 Nabu

During the realization of this thesis an implementation of the HMM training process has been developed in C++. The resulting library is inspired in HTK implementation and other speech recognition toolkits. Personal implication in this project is related to beam pruning in embedded re-estimation, decision tree clustering with MDL criterion, adaptation transformations and regression classes.

Appendix F

Contributions

Scientific contributions associated to the current work are briefed in the following appendix. In addition, it also includes participation in research projects with public and private funds.

F.1 Scientific contributions

F.1.1 International conferences

1. Xavi Gonzalvo, Ignasi Iriondo, Joan Claudi Socoró, Francesc Alías, Carlos Monzo (2007), "**HMM-based Spanish speech synthesis using CBR as F0 estimator**", In *Proceedings of Advances in Non-Linear Speech Processing (NoLISP)*, Paris (France).

   Abstract Hidden Markov Models based text-to-speech (HMM-TTS) synthesis is a technique for generating speech from trained statistical models where spectrum, pitch and durations of basic speech units are modelled altogether. The aim of this work is to describe a Spanish HMM-TTS system using CBR as a F0 estimator, analysing its performance objectively and subjectively. The experiments have been conducted on a reliable labelled speech corpus, whose units have been clustered using contextual factors according to the Spanish language. The results show that the CBR-based F0 estimation is capable of improving the HMM-based baseline performance when synthesizing non declarative short sentences and reduced contextual information is available.

2. Xavi Gonzalvo, Ignasi Iriondo, Joan Claudi Socoró, Francesc Alías and Carlos Monzo (2007), "**Mixing HMM-Based Spanish Speech Synthesis with a CBR for Prosody Estimation**", In *Lecture Notes in Computer Science*, pp. 78-85, Volume 4885, Springer, Heidelberg (Germany).

   Abstract Hidden Markov Models based text-to-speech (HMM-TTS) synthesis is a technique for generating speech from trained statistical models where spectrum, pitch and durations of basic

speech units are modelled altogether. The aim of this work is to describe a Spanish HMM-TTS system using an external machine learning technique to help improving the expressiveness. System performance is analysed objectively and subjectively. The experiments were conducted on a reliably labelled speech corpus, whose units were clustered using contextual factors based on the Spanish language. The results show that the CBR-based F0 estimation is capable of improving the HMM-based baseline performance when synthesizing non-declarative short sentences while the durations accuracy is similar with the CBR or the HMM system.

3. Xavier Gonzalvo, Joan Claudi Socoró, Ignasi Iriondo, Carlos Monzo, Elisa Martínez (2007), "**Linguistic and Mixed Excitation Improvements on a HMM-based speech synthesis for Castilian Spanish**", In *Sixth ISCA Workshop on Speech Synthesis (SSW6)*, pp. 362-367, Bonn (Germany).

   Abstract Hidden Markov Models based text-to-speech (HMM-TTS) synthesis is one of the techniques for generating speech from trained statistical models where spectrum and prosody of basic speech units are modelled altogether. This paper presents the advances in our Spanish HMM-TTS and a perceptual test is conducted to compare it with an extended PSOLA-based concatenative (E-PSOLA) system. The improvements have been performed on phonetic information and contextual factors according to the Castilian Spanish language and speech generation using a mixed excitation (ME) technique. The results show the preference of the new HMM-TTS system in front of the previous system and a better MOS in comparison with a real E-PSOLA in terms of acceptability, intelligibility and stability.

4. Xavi Gonzalvo, Paul Taylor, Carlos Monzo, Ignasi Iriondo, Joan Claudi Socoró (2009), "**High Quality Emotional HMM-Based Synthesis in Spanish**", In *Proceedings of Advances in Non-Linear Speech Processing (NoLISP)*, Vic (Spain).

   Abstract This paper describes a high-quality Spanish HMM-based speech synthesis of emotional speaking styles. The quality of the HMM-based speech synthesis is enhanced by using the most recent features presented for the Blizzard system (i.e., STRAIGHT spectrum extraction and mixed excitation). Two techniques are evaluated. First, a method simultaneously model all emotions within a single acoustic model. Second, an adaptation techniques to convert a neutral emotional style to a target emotion.We consider 3 kinds of emotions expressions: neutral, happy and sad. A subjective evaluation will show the quality of the system and the intensity of the produced emotion while an objective evaluation based on voice quality parameters evaluates the effectiveness of the approaches.

5. Xavi Gonzalvo, Alexander Gutkin, Joan Claudi Socoró, Ignasi Iriondo, Paul Taylor (2009), "**Local minimum generation error criterion for hybrid HMM speech synthesis**", In *10th Annual Conference of the International Speech Communication Association (ICSLP)*, pp. 416–419, Brighton (UK).

   Abstract This paper presents an HMM-driven hybrid speech synthesis approach in which unit selection concatenative synthesis is used to improve the quality of the statistical system using a Local Minimum Generation Error (LMGE) during the synthesis stage. The idea behind this approach is to combine the robustness due to HMMs with the naturalness of concatenated units.

> Unlike the conventional hybrid approaches to speech synthesis that use concatenative synthesis as a backbone, the proposed system employs stable regions of natural units to improve the statistically generated parameters. We show that this approach improves the generation of vocal tract parameters, smoothes the bad joins and increases the overall quality.

Publications 1 and 2 refer to the novel Spanish HMM-based TTS system. In the former, a mixed F0 estimator approach is presented as described in Section 2.6.3. In the latter, the prosody of the external F0 estimator is compared to the one produced by the HMM system.

In publication 3, the improvement to the basic Spanish HMM-based TTS system are presented. Specifically, a sub-band mixed excitation (see Section 3.5.3) is presented and finer linguistic features are also used (see Section 2.8.3).

Publication 4 presents an HMM-based emotional TTS system. Adaptation is used to transform emotions from a neutral style (adaptation applied in this paper is described in Section 2.7).

Publication 5 introduces a novel approach for a hybrid system based on an HMM-driven approach as described in Section 4.3.

F.1.2 In collaboration with the research group

The following publications refer to the work develop within the GPMM group as part of the work develop throughout this work.

1. Francesc Alías, Xavier Sevillano, Joan Claudi Socoró, Xavier Gonzalvo (2008), "**Towards high quality next-generation Text-to-Speech synthesis: a Multidomain approach by automatic domain classification**", In *IEEE Transactions on Audio, Speech and Language Processing (Special issue on New Approaches to Statistical Speech and Text Processing)*, vol. 16 (7), pp. 1340-1354.

2. Carlos Monzo, Francesc Alías, Ignasi Iriondo, Xavier Gonzalvo, Santiago Planet (2007), "**Discriminating Expressive Speech Styles by Voice Quality Parameterization**", In *Proceedings of 16th International Congress of Phonetic Sciences*, pp. 2081-2084, Saarbrücken (Germany).

3. Francesc Alías, Joan Claudi Socoró, Xavier Sevillano, Ignasi Iriondo, Xavier Gonzalvo (2006), "**Multi-domain Text-to-Speech Synthesis by Automatic Text Classification**", In *Proceedings of 9th International Conference on Spoken Language Processing (ICSLP)*, pp. 1304-1307, Pittsburgh (USA).

4. Francesc Alías, Xavier Gonzalvo, Xavier Sevillano, Joan Claudi Socoró, José Antonio Montero, David García (2006), "**Text Classification adapted to Multi-domain Text-to-Speech Synthesis**", In *Procesamiento del Lenguaje Natural (PLN)*, n 37, pp. 267-274, Zaragoza (Spain).

5. Carlos Monzo, Francesc Alías, José Antonio Morán, Xavier Gonzalvo (2006), "**Phonetic Transcription of Spanish Acronyms by using C4.5 algorithm**", In *Procesamiento del Lenguaje Natural (PLN)*, n 37, pp. 275-282, Zaragoza (Spain).

6. Ignasi Iriondo, Joan Claudi Socoró, Lluís Formiga, Xavier Gonzalvo, Alías, Pere Miralles, (2006), "**Modelado y estimación de la prosodia mediante razonamiento basado en casos**", In *IV Jornadas en Tecnología del Habla*, pp. 183-188. Zaragoza (Spain).

7. Frances Alías, Ignasi Iriondo, Lluís Formiga, Xavier Gonzalvo, Carlos Monzo, Xavier Sevillano, (2005), "**High quality Spanish restricted-domain TTS oriented to a weather forecast application**", In *The 9th European Conference on Speech Communication and Technology (ICSLP)*, pp. 2573-2576. Lisbon (Portugal).

8. Francesc Alías, Xavier Sevillano, Joan Claudi Socoró, Xavier Gonzalvo (2008), "**Towards high quality next-generation Text-to-Speech synthesis: a Multidomain approach by automatic domain classification**", In *IEEE Transactions on Audio, Speech and Language Processing (Special issue on New Approaches to Statistical Speech and Text Processing)*, vol. 16 (7), pp. 1340–1354

F.2 Thesis collaborations

This thesis has been conducted in collaboration with the Media Technologies Research Group and Phonetic Arts Ltd.

F.2.1 Media Technologies Research Group

This group was born as the fusion of two emergent groups: the Group of Multimodal Processing (GPMM) and the Group of Audiovisual Technologies and Multimedia (GTAM). The goal is to lead the so-called media technologies (e.g., music, video and image processing, mobile technologies, digital TV or internet) and to develop HCI frameworks which have gradually been designed through many years of expertise.

This is a multidisciplinary group. Research is focused on the following topics:

- Media processing (e.g., speech and image).
- Acoustics.
- Media communications.
- Computer animation.
- Usability and user interaction.

F.2.2 Phonetic Arts Ltd.

PA Studio is the product developed by Phonetic Arts Ltd. It consists of a graphical interface and cutting edge speech technology which intends to:

- Produce high-quality synthetic speech.
- Real time speech modifications.
- Transfer the building process to the customer side so everyone can build their own voice.

As the HMM leader in the research team of Phonetic Arts Ltd. the idea was to improve the baseline quality of the state-of-the-art HMM-based TTS systems. During this time, research and work was developed in the following parts:

- An HMM-based TTS system in English using parameter generation. A real-time statistical TTS system was developed and the quantization techniques were used to reduce the footprint.
- Emotion adaptation for Spanish (Gonzalvo et al., 2009b).
- The development of an HMM toolkit for training and adaptation.
- Research on hybrid approaches (Gonzalvo et al., 2009a).

F.3 Research projects

F.3.1 Public funding

F.3.1.1 Semantic AudiovisuaL Entertainment Reusable Objects (SALERO)

This is a research project co-funded by the European Union through the IST programme under FP6 (IST-FP6-027122). It aims at making cross media-production for games, movies and broadcast faster, better and cheaper by combining computer graphics, language technology, semantic web technologies as well as content based search and retrieval. It is conducted in collaboration with thirteen partners including companies and research centres.

Impact of the project will define and develop “intelligent content” for media production, consisting of multimedia objects with context-aware behaviours for self-adaptive use and delivery across different platforms. “Intelligent Content” should enable the creation and re-use of complex, compelling media by artists who need to know little of the technical aspects of how the tools that they use actually work.

The project main innovation is based on research into methodologies for describing, creating and finding intelligent content, SALERO will develop tool sets to create, manage, edit, retrieve and deliver content objects, addressing characters, objects, sounds, language sets, and behaviours. The tool sets developed and the concept of intelligent content will be verified by experimental productions.

F.3.1.2 IntegraTV4all

Audiovisual technology for interactive television (e-inclusion) is already finished and funded by the *Ministerio de Ciencia y Tecnología* as FIT-350301-2004-2.

This R&D project was developed by the TMT software factory in collaboration with the *Fundación ONCE*, universities Carlos III and Politécnica de Madrid. The aim was to create services adapted to leisure, information and remote work by using the hotels televisions. These services where designed with advanced vision and speech interfaces. Work developed through this project was related with the design of the a reusable talking head using the Virtual Speaker as the baseline.

F.3.2 Private funding

F.3.2.1 TTS system for weather forecasting

During 2004 this project created a high-quality limited domain speech synthesis module for an automatic weather forecast program. Concretely, this TTS has been developed in a research project titled "Virtual Speakers" in whom participates the Catalan Corporation of Radio and Television (CCRTV), leading the project, the Group of Interactive Technologies (GTI) of the University Pompeu Fabra (UPF) and our research group.

Taking into account these objectives, the following system was designed:

- Rich expressiveness of the synthetic speech. The prosody estimation retrieves the original intonation from the recorded corpus.
- Simplified cost function for the unit selection module that significantly reduces the computational cost.
- It works with as a multidomain system though there is no Text Classification (TC) module since texts are automatically tagged for the TTS query.
- The system is designed to automatically synchronize with audio-visual events (i.e., lip synching).

"Lo último que uno descubre al realizar un trabajo es por donde empezar."

"The last thing one discovers in composing a work is what to put first."

Blaise Pascal

References

Abdi, H., Lewis-Beck, M., Bryman, A., and Futing, T. (2003). *Least Squares*, pages 559–561. Sage, Thousand Oaks (CA).

Adell, J. and Bonafonte, A. (2006). Towards phone segmentation for concatenative speech synthesis. In *Proc. of ICSLP*, pages 139–144, Pittsburgh, PA, USA.

Akyol, A. and Erdogan, H. (2004). Filler model based confidence measures for spoken dialogue systems: a case study for turkish. In *Proc. of ICASSP*, volume I, pages 781–784.

Alías, F., Gonzalvo, X., Sevillano, X., Socoró, J., Montero, J., and García, D. (2006a). Clasificación de textos adaptada para conversión de texto en habla multidominio. *Procesamiento del Lenguaje Natural*, 37:267–274.

Alías, F., Iriondo, I., Formiga, L., Gonzalvo, X., Monzo, C., and Sevillano, X. (2005). High quality spanish restricted-domain TTS oriented to a weather forecast application. In *Proc. of ICSLP*, pages 2573–2576, Lisbon, Portugal.

Alías, F., Monzo, C., and Socoró, J. (2006b). A pitch marks filtering algorithm based on restricted dynamic programming. In *Proc. of ICSLP*, pages 1304–1307, Pittsburgh, PA, USA.

Alías, F., Sevillano, X., Socoró, J. C., and Gonzalvo, X. (2008). Towards high quality next-generation Text-to-Speech synthesis: a Multidomain approach by automatic domain classification. *IEEE Transactions on Audio, Speech and Language Processing*, 16(7):1340–1354.

Aoki, N., Takaya, K., Aoki, Y., and Yamamoto, T. (2000). Development of a rule-based speech synthesis system for the japanese language using a melp vocoder. In *Proc. of IEEE Int. Sympo. on Intelligent Signal Processing and Communication Systems*, pages 773–776, Tampere, Finland. X European Signal Processing Conference.

Badin, P., Bailly, G., and Bo, L. (1998). Towards the use of a virtual talking head and of speech mapping tools for pronunciation training. In *Proc. of the ESCA Tutorial and Research Workshop on Speech Technology in Language Learning*, Stockholm, Sweden.

Black, A. and Campbell (1995). Predicting the intonation of discourse segments from examples in dialogue speech. In *Proc. of ESCA (European Speech Communication Association) Workshop on Spoken Dialogue Systems*, pages 197–200, Vigso, Denmark.

Black, A. and Taylor, P. (1997). Automatically clustering similar units for unit selection in speech synthesis. In *Proc. of Eurospeech*, pages 601–604, Rhodes, Greece.

Black, A., Zen, H., and Tokuda, K. (2007). Statistical parametric speech synthesis. In *Proc. of ICASSP*, volume IV, pages 1229–1232.

Black, A. W. (2002). Perfect synthesis for all of the people all of the time. In *Proc. of IEEE Workshop on Speech Synthesis*.

Black, A. W., Taylor, P., and Caley, R. (1999). The Festival Speech Synthesis System -for The Festival Speech Synthesis System. http://www.speech.cs.cmu.edu/festival/.

Bonafonte, A., Adell, J., Esquerra, I., Gallego, S., Moreno, A., and Pérez, J. (2008). Corpus and voices for catalan speech synthesis. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Breuer, S. and Hess, W. (2010). he bonn open synthesis system 3. In *International Journal of Speech Technology*.

Bulut, M., Narayanan, S., and Syrdal, A. (2002). Expressive speech synthesis using a concatenative synthesizer. In *Proc. of ICSLP*, Denver (USA).

Chazan, D., Hoory, R., Cohen, G., and Zibulski, M. (2000). Speech reconstruction from mel frequency cepstral coefficients and pitch. In *Proc. of ICASSP*.

Chazan, D., Hoory, R., Sagi, A., Shechtman, S., Sorin, A., Shuang, Z. W., and Bakis, R. (2006). High quality sinusoidal modeling of wideband speech for the purposes of speech synthesis and modification. In *Proc. of ICASSP*.

Chu, W. C. (2003). *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*. John Wiley & Sons, Inc., New York, NY, USA.

Damper, R., Marchand, Y., Adamson, M., and Gustafson, K. (1999). A Comparison Of Letter-To-Sound Conversion Techniques For English Text-To-Speech Synthesis. In *Proc. of the Institute of Acoustics*.

Digalakis, V. and Neumeyer, L. (1996). Speaker adaptation using combined transformation and bayesian methods. *IEEE Trans. Speech Audio Process.*, 4(3):294–300.

Donovan, R. E. and Eide, E. (1998). The IBM trainable speech synthesis system. In *Proc. of ICSLP*, volume 5, pages 1703–1706.

Donovan, R. E. and Woodland, P. C. (1995). Improvements in an HMM-based speech synthesiser. In *Proc. of Eurospeech*, volume 1, pages 573–576, Madrid, Spain.

Donovan, R. E. and Woodland, P. C. (1999). A hidden Markov-model-based trainable speech synthesizer. *Computer Speech and Language*, 13:223–241.

Drugman, T., Wilfart, G., Moinet, A., and Dutoit, T. (2009). Using a pitch-synchronous residual codebook for hybrid HMM/frame selection speech synthesis. In *Proc. of ICASSP*.

Dutoit, T. (1994). On the ability of various speech models to smooth segment discontinuities in the context of text-to-speech synthesis by concatenation. In *Proc. of EUSIPCO*, pages 8–11, Edinburgh, UK.

Engwall, O., Delvaux, V., and Metens, T. (2006). Interspeaker variation in the articulation of french nasal vowels. In *Proc. of the Seventh International Seminar on Speech Production*, Ubatuba. International Seminar on Speech Production.

Erro, D. (2008). *Intra-lingual and Cross-lingual Voice Conversion Using Harmonic plus Stochastic Models*. PhD thesis, Universitat Politècnica de Barcelona. Director: Dr. Asunción Moreno.

Forney, G. J. (1973). The viterbi algorithm. *Proc. of IEEE*, 61(3):268–278.

Freij, G. and Fallside, F. (1988). Lexical stress recognition using hidden Markov models. In *Proc. of ICASSP*, volume 1, pages 135–138.

Fukada, T., Komori, Y., Aso, T., and Ohora, Y. (1994). A study of pitch pattern generation using HMM-based statistical information. In *Proc. of ICSLP*, pages 723–726.

Fukada, T., Tokuda, K., Kobayashi, T., and Imai, S. (1992). An adaptive algorithm for mel-cepstral analysis of speech. In *Proc. of ICASSP*, pages 137–140.

G., J. and Lee, C. (1994). Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298.

Gales, M. (1997). Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition. *Computer Speech and Language*, 12:75–98.

Garrido, J. M. (1996). *Modelling Spanish Intonation for Text-to-Speech Applications*. PhD thesis, Universitat Autònoma de Barcelona.

Garrido, J. M. (2001). La estructura de las curvas melódicas del español: propuesta de modelización. *Lingüística Española Actual*, 23(2):173–209.

Giustiniani, M. and Pierucci, P. (1991). Phonetic ergodic HMM for speech synthesis. In *Proc. of Eurospeech*, pages 349–352, Genova, Italy.

Gonzalvo, X., Gutkin, A., Socoró, J. C., Iriondo, I., and Taylor, P. (2009a). Local minimum generation error criterion for hybrid HMM speech synthesis. In *Proc. of ICSLP*, pages 416–419, Brighton, UK.

Gonzalvo, X., Iriondo, I., Socoró, J., Alías, F., and Monzo, C. (2007a). HMM-based spanish speech synthesis using CBR as F0 estimator. In *Proc. of NoLISP*, Paris, France.

Gonzalvo, X., Socoró, J., Iriondo, I., Monzo, C., and Martínez, E. (2007b). Linguistic and Mixed Excitation Improvements on a HMM-based speech synthesis for Castilian Spanish. In *Proc. of the IEEE Speech Synthesis Workshop*, Bonn, Germany.

Gonzalvo, X., Taylor, P., Monzo, C., Iriondo, I., and Socoró, J. C. (2009b). High quality emotional HMM-based synthesis in spanish. In *Proc. of NoLISP*, Vic (Spain).

Hemptinne, C. (2006). *Integration of the Harmonic plus Noise Model (HNM) into the Hidden Markov Model-Based Speech Synthesis System (HTS)*. Idiap-rr, IDIAP.

Hirai, T., Yamagishi, J., and Tenpaku, S. (2007). Utilization of an HMM-based feature generation module in 5 ms segment concatenative speech synthesis. In *Proc. of the IEEE Speech Synthesis Workshop*.

Hon, H., Acero, A., Huang, X., Liu, J., and Plumpe, M. (1998). Automatic generation of synthesis units for trainable text-to-speech systems. In *Proc. of ICASSP*, pages 293–306.

HTS. Hmm-based speech synthesis system (HTS). http://hts.ics.nitech.ac.jp.

HTS. Hmm-based speech synthesis system (HTS) engine. http://hts-engine.sourceforge.net/.

Huang, X., Acero, A., Hon, H., Ju, Y., Liu, J., Meredith, S., and Plumpe, M. (1997). Recent improvements on Microsoft's trainable Text-To-Speech system - Whistler. In *Proc. of ICASSP*, pages 959–962.

Huang, X. and Hon, H.-W. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA. Foreword By-Raj Reddy.

Hunt, A. J. and Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *Proc. of ICASSP*, pages 373–376, Washington, DC, USA.

Hunt, A. J., Zwierzyriski, D. A., and Can, R. C. (1989). Issues in High Quality LPC Analysis and Synthesis. In *Proc. of Eurospeech*, pages 2348–2351, Paris (France).

Imai, S. (1983). Cepstral analysis synthesis on the mel frequency scale. In *Proc. of ICASSP*, pages 93–96.

Imai, S. and Furuichi, C. (1988). Unbiased estimator of log spectrum and its application to speech signal processing. In *Proc. of EURASIP*, pages 203–206.

Inanoglu, Z. and Young, S. (2007). A system for transforming the emotion in speech: Combining data-driven conversion techniques for prosody and voice quality. In *Proc. of ICSLP*, Antwerp (Belgium).

Iriondo, I., Alías, F., Sanchis, J., and Melenchón, J. (2003). A Hybrid Method Oriented to Concatenative Text-to-Speech Synthesis. In *Proc. of Eurospeech*, volume 4, pages 2953–2958, Geneve (Switzerland).

Iriondo, I., Alías, F., and Socoró, J. (2007a). Prosody modelling of spanish for expressive speech synthesis. In *Proc. of ICASSP*, volume IV, pages 821–824, Honolulu (HI).

Iriondo, I., Planet, S., Socoró, J., and Alías, F. (2007b). Objective and subjective evaluation of an expressive speech corpus. In *Proc. of NoLISP*, Paris (France).

Iriondo, I., Socoró, J., Formiga, L., Gonzalvo, X., Alías, F., and Miralles, P. (2006). Modeling and estimating of prosody through CBR. In *Proc. of JTH*, Zaragoza, Spain.

Jensen, U., Moore, R. K., Dalsgaard, P., and Lindberg, B. (1994). Modelling intonation contours at the phrase level using continuous density hidden Markov models. *Computer, Speech and Language*, 3(8):227–260.

Jiang, H., Li, X., and Liu, C. (2006). Large Margin Hidden Markov Models for Speech Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 14:1584–1595.

Kataoka, S., Mizutani, N., Tokuda, K., and Kitamura, T. (2004). Decision-Tree Backing-off in HMM-based Speech Synthesis. In *Proc. of ICSLP*, pages 1205–1208, Jeju Island (Korea).

Kawahara, H. (1997). Speech representation and transformation using adaptive interpolation of weighted spectrum: VOCODER revisited. In *Proc. of ICASSP*, volume 2, pages 1303–1306, Washington, DC (USA).

Kawahara, H. (1999). Straight, trial version. http://www.wakayama/u.ac.jp/ kawahara/STRAIGHTTrial/.

Kawahara, H., Estill, J., and Fujimura, O. (2001). Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT. In *Proc. of MAVEBA*, Firenze (Italy).

Kawai, H., Toda, T., Ni, J., Tsuzaki, M., and Tokuda, K. (2004). XIMERA: A new TTS from ATR based on corpus-based technologies. In *Proc. of the IEEE Speech Synthesis Workshop*, pages 179–184.

Kawanami, H., Iwami, Y., Toda, T., Saruwatari, H., and Shikano, K. (2003). GMM-based voice conversion applied to emotional speech synthesis. In *Proc. of Eurospeech*, Geneva (Switzerland).

Keller, E. and Keller, B. (2003). How much prosody can you learn from twenty utterances? *Linguistik Online*, 17(5):57–79.

Klatt, D. H. (1980). Software for a cascade/parallel formant synthesizer. *Journal of the Acoustical Society of America*, 67(3):971–995.

Kobayashi, T. and Imai, S. (1984). Spectral analysis using generalized cepstrum. *IEEE Trans. Acoust., Speech, Signal processing*, 32:1087–1089.

Kominek, J. and Black, A. (2004). The CMU ARCTIC speech databases. In *Proc. of the IEEE Speech Synthesis Workshop*, pages 223–226, Pittsburgh, PA, USA.

Lambert, T. and Breen, A. (2004). A database design for a TTS synthesis system using lexical diphones. In *Proc. of ICSLP*, pages 1381–1384, Jeju Island, Korea.

Latorre, J., Iwano, K., and Furui, S. (2006). New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer. *Speech Communication*, 48(10):1227–1242.

Lee, C. H. (1991). A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models. *IEEE Transactions on Signal Processing*, 39(4):806–814.

Leggetter, C. and Woodland, P. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9(2):171–185.

Lenzo, K. A. and Black, A. W. (2000). Diphone Collection and Synthesis. In *Proc. of ICSLP*, Beijing, China.

Ling, Z., Qin, L., Lu, H., Gao, Y., Dai, L., Wang, R., Jiang, Y., Zhao, Z., Yang, J., Chen, J., and Hu, G. (2007). The USTC and iFlytek speech synthesis systems for Blizzard Challenge 2007. In *Proc. of Blizzard Challenge Workshop*.

Ling, Z. and Wang, R. (2008). Minimum unit selection error training for HMM-based unit selection speech synthesis system. In *Proc. of ICASSP*, pages 3949–3952.

Ling, Z.-H., Wu, Y.-J., Wang, Y.-P., Qin, L., and Wang, R.-H. (2006). USTC system for blizzard challenge 2006 an improved HMM-based speech synthesis method. In *Proc. of Blizzard Challenge Workshop*.

Ljolje, A. and Fallside, F. (1986). Synthesis of natural sounding pitch contours in isolated utterances using hidden Markov models. In *Proc. of ICASSP*, volume 3, pages 1074–1080.

Llisterri, J. and Mario, J. (1993). Spanish adaptation of SAMPA and automatic phonetic transcription. Technical Report SAMA/ UPC/001. ESPRIT PROJECT 6819, UPC.

Maia, R., Toda, T., Zen, H., Nankaku, Y., and Tokuda, K. (2007). A trainable excitation model for HMM-based speech synthesis. In *Proc. of ICSLP*, pages 1909–1912, Antwerp, Belgium.

Maia, R., Toda, T., Zen, H., Nankaku, Y., and Tokuda, K. (2008). An excitation model for HMM-based speech synthesis based on residual modeling. In *Proc. of the IEEE Speech Synthesis Workshop*, pages 131–136.

Maia, R., Zen, H., Tokuda, K., Kitamura, T., and Resende Jr., F. (2003). Towards the development of a brazilian portuguese text-to-speech system. In *Proc. of Eurospeech*, pages 2465–2468.

Marcheret, E., Chu, S. M., Goel, V., and Potamianos, G. (2004). Efficient likelihood computation in multi-stream HMM based audio-visual speech recognition. In *Eighth International Conference on Spoken Language Processing*.

Masuko, T. (2002). *HMM-Based Speech Synthesis and Its Applications*. PhD thesis, Tokyo Institute of Technology.

Masuko, T., Tokuda, K., Kobayashi, T., and Imai, S. (1996a). Speech synthesis using HMMs with dynamic features. In *Proc. of ICASSP*, pages 389–392.

Masuko, T., Tokuda, K., Kobayashi, T., and Imai, S. (1996b). Speech synthesis using HMMs with dynamic features. In *Proc. of ICASSP*, pages 389–392, Washington, DC (USA).

McCree, A. and Barnwell, T. (1995). A mixed excitation LPC vocoder model for low bit rate speech coding. *IEEE Transactions on Speech and Audio Processing*, 3(4):242–250.

McCree, A. V., Supplee, L. M., Cohn, R. P., and Collura, J. S. (1997). MELP: The new federal standard at 2400 bps. In *Proc. of ICASSP*, pages 1591–1594.

McDermott, E. (2000). *Handbook of Neural Networks for speech processing*. Artech House.

Melenchón, J., la Torre, F. D., Iriondo, I., Alías, F., Martínez, E., and Vicent, L. (2003). Text to visual synthesis with appearance models. In *Proc. of ICSLP*, Barcelona, Spain.

Michaelis, D., Gramss, T., and Strube, H. W. (1997). Glottal to noise excitation ratio - a new measure for describing pathological voices. In *Acustica*, pages 700–706. Acta Acustica.

Montero, J. M., Guiterrez-Arriola, J., Colas, J., Macias, J., Enriquez, E., and Pardo, J. M. (1999). Development of an emotional speech synthesizer in spanish. In *In Proc of Eurospeech*, pages 2099–2102, Budapest.

Monzo, C., Alías, F., Iriondo, I., Gonzalvo, X., and Planet, S. (2007). Discriminating Expressive Speech Styles by Voice Quality Parameterization. In *Proc. of ICPhS*.

Monzo, C., Iriondo, I., and Martínez, E. (2008). Procedimiento para la medida y la modificación del jitter y del shimmer aplicado a la síntesis del habla expresiva. In *Proc of JTH*, Bilbao (Spain).

Moore, B. C. J. and Glasberg, B. R. (1996). A revision of Zwicker's loudness model. In *Proc. of Acta Acustica*, volume 82, pages 335–345.

Moulines, E. and Charpentier, F. (1990). Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9:453–467.

Moulines, E. and Verhelst, W. (1995). Time-Domain and Frequency-Domain. Techniques for Prosodic Modification of Speech. In *Speech coding and Synthesis*, pages 519–555.

Nam, Y. and Wohn, K. (1996). Recognition of space-time handgestures using Hidden Markov model. In *ACM Symposium on Virtual Reality Software and Technology*, pages 51–58.

Narayanan, S. and Alwan, A. (2005). *Text to Speech Synthesis: New Paradigms and Advances*. Prentice Hall.

Odell, J. (1995). *The use of context in large vocabulary speech recognition*. PhD thesis, University of Cambridge. Director: Dr. Steve Young.

Okubo, T., Mochizuki, R., and Kobayashi, T. (2006). Hybrid Voice Conversion of Unit Selection and Generation Using Prosody Dependent HMM. *IEICE Trans. Inf. Syst.*, E89-D(11):2775–2782.

Oliver, N. and Horvitz, E. (2005). A Comparison of HMMs and Dynamic Bayesian Networks for Recognizing Office Activities. In *Proc. of Int. Conf. on User Modeling (UM'05)*.

Oliver, N. and Pentland, A. (2000). Graphical models for driver behavior recognition in a smartcar. In *Proc. of IEEE Intl. Conference on Intelligent Vehicles*.

Oura, K., Zen, H., Nankaku, Y., Lee, A., , and Tokuda, K. (2009). Tying covariance matrices to reduce the footprint of HMM-based speech synthesis systems. In *Proc. of ICSLP*, pages 1759–1761, Brighton, UK.

Pitrelli, J. and Eide, E. (2003). Expressive speech synthesis using american english ToBI: questions and contrastive emphasis. In *Proc. of ASRU*, pages 694–699.

Plumpe, M., Acero, A., Hon, H., and Huang, X. (1998). HMM-based smoothing for concatenative speech synthesis. In *Proc. of ICSLP*, Sydney, Australia.

Pollet, V. and Breen, A. (2008). Synthesis by generation and concatenation of multiform segments. In *Proc. of ICSLP*, pages 1825–1828, Brisbane (Australia).

Price, P., Fisher, W., Bernstein, J., and Pallett, D. (1988). The DARPA 1000-word resource management database for continuous speech recognition. In *Proc. of ICASSP*.

Qin, L., Wu, Y.-J., Ling, Z.-H., and Wang, R.-H. (2008). Model adaptation for HMM-based speech synthesis under minimum generation error criterion. In *Proc. of ISM*.

Rabiner, L. R. (1990). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In Waibel, A. and Lee, K.-F., editors, *Readings in Speech Recognition*, pages 267–296. Kaufmann, San Mateo, CA (USA).

Raiffa, H. (1961). *Applied Statistical Decision Theory.* Wiley Classics Library.

Rathinavelu, C. and Deng, L. (1995). Use of generalized dynamic feature parameters for speech recognition: maximum likelihood and minimum classification error approaches. In *Proc. of ICASSP*, volume 1, pages 373–376. Detroit (USA).

Rennison, E. (1994). Galaxy of news: An approach to visualizing and understanding expansive news landscapes. In *Proc. of ACM Symposium on User Interface Software and Technology*, pages 3–12.

Rissanen, J. (1984). Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, 30:629–636.

Ross, K. and Ostendorf, M. (1994). A dynamical system model for generating f0 for synthesis. In *Proc. of the IEEE Speech Synthesis Workshop*, pages 131–134, New Paltz, NY (USA).

Rouibia, S. and Rosec, O. (2005). Unit selection for speech synthesis based on a new acoustic target cost. In *Proc. of ICSLP*, pages 2565–2568, Lisbon, Portugal.

Rozak, M. (2007). Text-to-speech designed for a massively multiplayer online role-playing game (MMORPG). In *Proc. of the IEEE Speech Synthesis Workshop*, Bonn (Germany).

Schwartz, R. M., Chow, Y., Kimball, O., Roucos, S., Krasner, M., and Makhoul., J. (1985). Context-dependent modeling for acoustic-phonetic recognition of continuous speech. In *Proc. of ICSLP*, pages 31.3.1–31.3.4.

Schweitzer, A., Braunschweiler, N., Dogil, G., and Möbius, B. (2004). Assessing the acceptability of the smartkom speech synthesis voices. In *Proc. of the IEEE Speech Synthesis Workshop*, pages 1–6.

Shannon, M. and Byrne, W. (2009). Autoregressive HMMs for speech synthesis. In *Proc. of ICSLP*, pages 400–403, Brighton, UK.

Shichiri, K., Sawabe, A., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. (2002). Eigenvoices for HMM-based speech synthesis. In *Proc. of ICSLP*, pages 1269–1272.

Shinoda, K. and Watanabe, T. (1997). Acoustic modeling based on the MDL criterion for speech recognition. In *Proc. of Eurospeech*, volume 1, pages 99–102.

Shinoda, K. and Watanabe, T. (2000). MDL-based context dependent subword modeling for speech recognition. *Acoust. Soc. Japan (E)*, 21:79–86.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J. (1992). Tobi: a standard for labelling english prosody. In *Proc. of ICSLP*, pages 867–870, Banff, Canada.

Sinder, D. J. (1999). *Speech synthesis using an aeroacoustic fricative model.* PhD thesis, Rutgers University. Director: James L. Flanagan.

Socoró, J., Iriondo, I., Martínez, E., Alías, F., Formiga, L., Monzo, C., Gonzalvo, X., Cullen, C., Payo, J., and de Vilar, G. (2007). *Early Speech Synthesis Software.* SALERO.

Stylianou, Y. (2001). Applying the harmonic plus noise model in concatenative speech synthesis. In *IEEE Transactions on Speech and Audio processing*, volume 9, pages 21–29.

Sugamura, N. and Itakura, F. (1986). Speech analysis and synthesis methods developed at ECL in NTT–from LPC to LSP. *Speech Communication*, 5(2):199–215.

Tachibana, M., Yamagishi, J., Masuko, T., , and Kobayashi, T. (2006). A style adaptation technique for speech synthesis using HSMM and suprasegmental features. *IEICE Transactions on Information and Systems*, E89-D(3):1092–1099.

Tachibana, M., Yamagishi, J., Masuko, T., and Kobayashi, T. (2005). Speech synthesis with various emotional expressions and speaking styles by style interpolation and morphing. *IEICE - Trans. Inf. Syst.*, E88-D(11):2484–2491.

Tamura, M., Kondo, S., Masuko, T., and Kobayashi, T. (1999). Text-to-audio-visual speech synthesis based on parameter generation from HMM. In *Proc. of Eurospeech*, pages 959–962.

Tamura, M., Masuko, T., Tokuda, K., and Kobayashi, T. (1998). Speaker adaptation for HMM-based speech synthesis system using mllr. In *Proc. of The Third ESCA/COCOSDA workshop on Speech Synthesis*, pages 273–276.

Tamura, M., Masuko, T., Tokuda, K., and Kobayashi, T. (2001). Text-to-speech synthesis with arbitrary speaker's voice from average voice from average voice. In *Proc. of Eurospeech.*

Taylor, P. (2000). Concept-to-speech synthesis by phonological structure matching. *Philosophical Transactions of the Royal Society*, 356:1403–1416.

Taylor, P. (2005). Grapheme-to-Phoneme conversion using Hidden Markov models. In *Proc. of Eurospeech*, pages 1973–1976, Lisbon, Portugal.

Taylor, P. (2006). Unifying Unit Selection and Hidden Markov Model Speech Synthesis. In *Proc. of ICSLP*, pages 1758–1761, Pittsburgh, PA, USA.

Taylor, P. (2009). *Text-to-Speech Synthesis.* Cambridge University Press.

Taylor, P. and Isard, S. (1991). Automatic phone segmentation. In *Proc. of Eurospeech*, pages 709–711, Genova (Italy).

Theodoridis, S. and Koutroumbas, K. (2006). *Patter recognition.* Elsevier Science.

Toda, T. and Tokuda, K. (2005). Speech parameter generation algorithm considering global variance for HMM-based speech synthesis. In *Proc. of ICSLP*, pages 2801–2804, Lisbon, Portugal.

Toda, T. and Tokuda, K. (2007). A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E90-D 5:816–824.

Tokuda, K. (1995). An algorithm for speech parameter generation from continuous mixture hmms with dynamic features. In *Proc. of Eurospeech*.

Tokuda, K., Kobayashi, T., and Imai, S. (1995). Speech parameter generation from HMM using dynamic features. In *Proc. of ICASSP*, volume 1, pages 660–663. Detroit (USA).

Tokuda, K., Kobayashi, T., Masuko, T., and Imai, S. (1994a). Mel-generalized cepstral analysis a unified approach to speech spectral estimation. In *Proc. of ICSLP*, pages 1043–1046, Denver, Colorado, USA.

Tokuda, K., Kobayashi, T., Masuko, T., and Imai, S. (1994b). Unified Approach To Mel-Generalized Cepstral Analysis. In *Proc. of ICSLP*, pages 1043–1046.

Tokuda, K., Masuko, T., Miyazaki, N., and Kobayashi, T. (1999). Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. *Acoustics, Speech, and Signal Processing, IEEE International*, 1:229–232.

Tokuda, K., Masuko, T., Miyazaki, N., and Kobayashi, T. (2002a). Multi-Space Probability Distribution HMM. *Special issue on the 2000 IEICE Excellent Paper Award*, E85-D 3:455–464.

Tokuda, K., Zen, H., and Black, A. W. (2002b). An HMM-based speech synthesis system applied to english. In *Proc. of the IEEE Speech Synthesis Workshop*.

Tokuda, Y., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis. In *Proc. of ICASSP*, pages 1315–1318.

Traber, C., Huber, K., Nedir, K., Pfister, B., Keller, E., and Zellner, B. (1999). From multilingual to polyglot speech synthesis. In *Proc. of Eurospeech*, pages 835–838, Budapest, Hungary.

Turetsky, R. and Ellis, D. (2003). Ground-truth transcriptions of real music from force-aligned MIDI syntheses. In *Proc. of 4th International Symposium on Music Information Retrieval ISMIR*, pages 135–141.

Vielhauer, C. and Scheidat, T. (2005). Multimodal biometrics for voice and handwriting. *Lecture Notes in Computer Science*, 3677:191–199.

W3C (2004). Speech synthesis markup language (SSML). http://www.w3.org/TR/speech-synthesis/.

Wouters, J. and Macon, M. W. (2000). Unit fusion for concatenative speech synthesis. In *Proc. of ICSLP*, pages 302–305, Beijinig, China.

Wu, Y.-J., Guo, W., and Wang, R. (2006). Minimum generation error criterion for tree-based clustering of context dependent HMMs. In *Proc. of ICSLP*, pages 2046–2049.

Wu, Y.-J. and Tokuda, K. (2009). Minimum generation error training by using original spectrum as reference for log spectral distortion measure. In *Proc. of ICASSP*.

Wu, Y.-J., Wang, R., and Soong, F. (2007). Full HMM training for minimizing generation error in synthesis. In *Proc. of ICASSP*, pages 517–520.

Wu, Y.-J. and Wang, R. H. (2006). Minimum generation error training for HMM-based speech synthesis. In *Proc. of ICASSP*, pages 89–92.

Yamagishi, J. and Kobayashi, T. (2007). Hidden Semi-Markov model and its speaker adaptation techniques. *IEICE Transactions on Audio, Speech and Language Processing*, 6.

Yamagishi, J., Masuko, T., and Kobayashi, T. (2004). HMM-based expressive speech synthesis – towards tts with arbitrary speaking styles and emotions. In *Proc. of Special Workshop in Maui (SWIM)*.

Yamagishi, J., Nose, T., Zen, H., Ling, Z., Toda, T., Tokuda, K., King, S., and Renals, S. (2009). Robust speaker-adaptive HMM-based text-to-speech synthesis. *IEEE Transactions on Audio, Speech and Language Processing*, 17(6):1208–1230.

Yamagishi., J., Ogata, K., Nakano, Y., Isogai, J., and Kobayashi, T. (2006). HSMM-based model adaptation algorithms for average-voice-based speech synthesis. In *Proc. of ICASSP*, volume 1, pages 77–80, Toulouse (France).

Yamagishi, J., Onishi, K., Masuko, T., and Kobayashi, T. (2005). Acoustic Modeling of Speaking Styles and Emotional Expressions in HMM-Based Speech Synthesis. *IEICE transactions on information and systems*, 88(3):502–509.

Yamagishi, J., Zen, H., Toda, T., and Tokuda, K. (2007). Speaker-independent HMM-based speech synthesis system - HTS-2007 system for the blizzard challenge 2007. In *Proc. of the IEEE Speech Synthesis Workshop*, Bonn (Germany).

Yi, J. R. W. and Glass, J. R. (1998). Natural sounding speech synthesis using variable length units. In *Proc. of ICSLP*.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kif, T. (1998). Duration modeling in HMM-based speech synthesis system. In *Proc. of ICSLP*, pages vol. 2, 29–32.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proc. of Eurospeech*.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. (2000). Speaker interpolation for HMM-based speech synthesis system. *Acoustical Science and Technology*, 21(4):199–206.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. (2001). Mixed excitation for HMM-based speech synthesis. In *Proc. of Eurospeech*, pages 2263–2266. Eurospeech.

Young, S. (2007). An Application Toolkit for HTK (ATK). http://mi.eng.cam.ac.uk/research/dialogue/atk_home.

Young, S., Evermann, G., Gales, M., Hain, T., and al. (2006). The HTK book. http://htk.eng.cam.ac.uk.

Young, S. J., Odell, J. J., and Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 307–312, Morristown, NJ, USA.

Young, S. R. (1994). Detecting misrecognitions and out-of-vocabulary words. In *Proc. of ICASSP*, volume II, pages 21–24.

Yu, K., Toda, T., Gasic, M., Keizer, S., Mairesse, F., Thomson, B., and Young, S. (2009). Probabilistic modelling of F0 in unvoiced regions in HMM based speech synthesis. In *Proc. of ICASSP*.

Zen, H. (2008). Statistical parametric speech synthesis. Talk at MIL Speech Seminar, Cambridge University.

Zen, H., Toda, T., Nakamura, M., and Tokuda, K. (2007a). Details of the nitech HMM-based speech synthesis system for the blizzard challenge 2005. *IEICE Transactions Trans. Inf. Syst.*, E90-D(1):325–333.

Zen, H., Toda, T., and Tokuda, K. (2006). The nitech-NAIST HMM-based speech synthesis system for the blizzard challenge 2006. In *Proc. of Blizzard Challenge Workshop*.

Zen, H., Tokuda, K., and Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*.

Zen, H., Tokuda, K., and Kitamura, T. (2004). An introduction of trajectory model into HMM-based speech synthesis. In *Proc. of the IEEE Speech Synthesis Workshop*, pages 191–196.

Zen, H., Tokuda, K., Masuko, T., Kobayasih, T., and Kitamura, T. (2007b). A Hidden Semi-Markov Model-Based Speech Synthesis System. *IEICE Trans. Inf. Syst.*, E90-D(5):825–834.

Zhang, L. and Renals, S. (2008). Acoustic-articulatory modelling with the trajectory HMM. *IEEE Singla Processing Letters*, 15:245–248.

Universitat Ramon Llull

Aquesta Tesi Doctoral ha estat defensada el dia ____ d_____ de 200

al Centre Escola Tcnica Superior d'Enginyeria Electrnica i Informtica La Salle

de la Universitat Ramon Llull

davant el Tribunal format pels Doctors sota signants, havent obtingut la qualificaci:

President/a

_____

Vocal

_____

Vocal

_____

Vocal

_____

Secretari/ria

_____

Doctorand/a
Xavier Gonzalvo Fructuoso

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es