ISSN 1870-4069

Comparison between LSP and MFCC parameterizations in a Spanish Speech Synthesis System

Carlos Franco[1], Abel Herrera[2], Boris Escalante[2], Fernando Del Río[2]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Artes, Puebla, Mexico

[2] Universidad Nacional Autónoma de México, Laboratorio de Tecnologías del Lenguaje, Mexico

francocarlos@gmail.com, abelhc@hotmail.com, boris@unam.mx,haitosan@hotmail.com

Abstract. Voice parameterization using Line Spectral Pair was implemented to a Mexican Spanish HMM-based Speech Synthesis System. Five phrases were synthesized and statistically validated by applying a MOS test to 30 listeners who analyzed the original voices versus synthetic voice. Results were compared with a synthesizer where MFCC was used as voice parameterization. Two aspects were evaluated on the voice: Naturalness and Intelligibility. The comparison shows that LSP parameterization is above the mean score and pointed better than MFCC.

Keywords. Speech synthesis, line spectral pair, MFCC, Spanish language synthesis.

1 Introduction

By the end of the 20th century, the Festival speech synthesis [1] system together with its variants, CLUNITS and CLUSTERGEN reached a remarkable naturalness in artificial speech. Festival-CLUNITS parameterizes sub-phonemes, it belongs to a kind of synthesis named parametric speech synthesis (SPSS). Festival-CLUSTERGEN [2] on the other hand, works with acoustic sub-phonemes it belongs to the unit selection speech synthesis types. Both Festival programs are concatenative synthesizers.

Some authors in this paper replicated these achievements by adjusting Festival to central Mexico Spanish [3]. Hidden Markov Models (HMM) were applied in speech synthesizers to search units using stochastic methods which meant some advancement in how natural the Synthesizer sounded. Such systems are known as Hidden Markov Models as Text to Speech Synthesis (HTS) [4] these are SPSS, in other words, they use parameterized speech.

Still today, the doubt remains wether the HTS systems overcome the Festival-CLUSTERGEN system [5]. Some of the authors adapted an HTS system to central Mexico spanish which showed certain improvements compared to Festival [3].

pp. 19–27; rec. 2017-09-05; acc. 2017-10-20

In Festival CLUNITS and HTS, the preferred parameterization was MFCC [6]. For the latter, the parameterization together with pitch and duration conform the speech signal, sometimes delta and delta-delta are included [7]. Unfortunately, no substantial improvement was shown.

Another essential aspect of HTS is the inclusion of a Vocoder filter to recreate the speech signal. This process occurs having pitch, duration and parameterization as inputs [6]. Unlike Festival-CLUSTERGEN where instead of synthesizing through a Vocoder, acoustical units are concatenated and smoothen out.

During the first decade of the current century, another mildly successful parameterization was experimented with. It was known as STRAIGHT. It was also validated in a Mexican spanish system. [8].

The success of MFCC and LSP was not due to their compression in terms of voice segments but because such parameterizations are solidly based on the acoustical characteristics of the human voice.

When LPC was first created, several variants of it came along. Among those Line Spectral Pair was very well received. LSP take into account the acoustical behavior of the speech signal within the vocal tract. At that point in time, this feature was not relevant, and the parameterization was no longer worked on for some time.

It has not been as widely used as MFCC, Nakatani and colleagues [9] hypothesized that MF-LSP is a little more efficient than LSP. The authors decided to continue that line of research with its respective experiments. After adjusting and statistically validating the system. The authors conclude that it efficiently produces speech synthesis in Spanish Language using LSP. Its naturalness and intelligibility were qualified above the mean and above previously validated MFCC based synthesis.

It has just happened during the last four years that major changes took place in Speech synthesis. From 2013, Deep Neural Networks [10] are used to synthesize speech. Different network architectures report improvements in the voice quality [11]. A major application of the current work is in hybrid systems where LSP are used in applications for low memory devices [12].

This document is organized as follows: Section 2 mentions related efforts towards validating LSP as a speech synthesis parameterization. Section 3 three briefs the reader on HTS Speech Synthesis. Section 4 summarizes the theory behind the parameterization. Section 5 describes LSP as speech parameterization. The experiments and its results are described in section 6 and the conclusions are given in section 7.

2 Related Work

LSP parameterization of a speech signal has been in the interest of speech synthesis and recognition for the last three decades. Nakatani [9] and colleagues evaluated LSP parameterized phrases, but their study was exclusively focused on analyzing isolated phonemes in japanese and not entire phrases. Arakawa and colleagues [13] applied LSP to improve certain features of the STRAIGHT synthesis system, but the principles of such system differ from those of the system the authors experimented with. Bäckstörm in his doctoral project [14], [15] makes a complete mathematical analysis of LSP but his work is theoretical and not focused exclusively on speech signals. Tokuda and his team [4] left the door open to experiment with Either LSP or MFCC but they focused on the HTS (Hidden Markov Models as Text to Speech Sythesis) system from a global perspective and do not report results on speech parameterization effectiveness.

3 Speech Synthesis Using HTS

HTS (Hidden Markov Models as Text to Speech Sythesis) is a proposal from the 2000´s. This type of synthesis decomposes a voice signal in three vectors which include its three main features: Mel General Cepstral coefficients MGC [16], F0 and duration. In practice, these vectors are obtained with a software named Signal Processing Tool Kit SPTK [4].

The vectors are accessed non-linearly to obtain the correct phoneme sequence in a spoken phrase. Therefore, the stochastic selection algorithm of Hidden Markov Models HMM is used in contrast with other synthesis systems, such as Festival [17] were phonemes are selected using a linear method named CART [18].

To compute the probability of the HMMs, the creators of HTS took advantage of a free distributed system developed by the university of Cambridge. The program is known as Hidden Markov Model Toolkit HTK [19].

HTK was originally designed for speech recognition.

Figure 1 shows a general scheme of HTS. More details can be found on the references [5] and the HTS website [4].

Before being able to synthesize a phrase, HTS need to be trained with the desired language specifications. Other characteristics are as well defined in the training stage (e.g. parameterization, number of coefficients, sampling frequency, etc.)

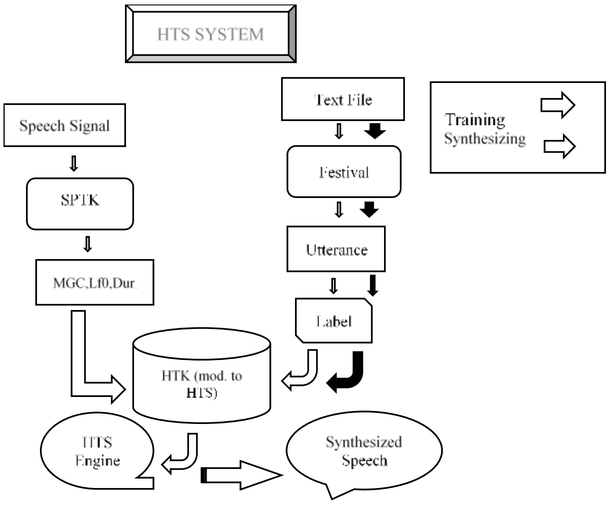

Fig. 1. HTS General Scheme.

The system is trained by inputting 300 audio files containing the recording of phonetically balanced phrases and text files with their respective transcription. The highest probabilities of occurrence of a phoneme sequence will be calculated within the HMMs to obtain the better combination. Text to phoneme conversion is done through Festival [17]. Since Festival was originally designed for english language synthesis, when a different language is used, the system must be adapted to process the grammatical features of such a language. All these grammatical features are coded in a software called lexicon. A lexicon in spanish indicates Festival the use of stressed vocals, letter “ñ”, differences between phonemes like /c/ or /z/ among others. The current system uses a lexicon created originally for Andalusian spanish named Junta de Andalucía. It was chosen because iberic spanish is grammatically identical to mexican Spanish, no further modifications were needed. Except for substituting “c” and “z” letters for an “s” when the desired synthesized phrase is being written. Text to phoneme conversion is performed in the following order: Sentence to phrase, phrase to word, word to syllable and syllable to phoneme [2].

Once the conversion is finished, Festival delivers a utterance (.utt) file. The actual synthesis process takes place in a software named HTS Engine, utterance files must be reorganized to be compatible with it. For that purpose, they are changed into label (.lab) files.

Input data to the system were used before in the spanish synthesis MFCC parameterization training. Such data consists of 300 phrases recorded as wave files in an anechoic chamber by a male professional radio speaker. The wave files were coded into RAW files which contain the same information of the wave file except for a header.

The other input data simultaneously processed are the label (.lab) files. These are text files which indicate HTS Engine the desired phoneme sequence (e.g. sentence, phrase, word, syllable) of the phrase to be synthesized.

The RAW files are decomposed in three vectors: One vector contains Mel General Cepstral Coefficients; the second vector contains the phrase LogF0 and the third one the phrase duration. These three elements are stored in a three-streamed HMM which is in practice a Gaussian matrix. Their delta and double delta Coefficients are also considered to smooth out the wave transitions within each other. A common practice in speech processing. This model is named hmm0. The calculations are done based on a previously given phoneme probability master label file MLF [19].

The model hmm0 should be divided into smaller models to separate the different phoneme values. For that matter, the mean of hmm0 is calculated generating a new three-streamed model named hmm1. The probabilities stated in the MLF are then condensed in a Master Macro File MMF. Based on this file probabilities, the process is repeated iteratively until several HMM models are formed. The number of HMM models is previously defined by the user.

Once the HMM models are completed, their probabilities are computed following a Viterbi algorithm and grouped into single phoneme gaussians. Thus, for example, all the /a/ phonemes are together in a same group. And the selection process will be linear.

The synthesis takes place in a piece of software named HTS Engine [5] which is a vocoder filter driven by two sound sources: Sinusoidal for voiced sounds and white noise for unvoiced sounds. The formers emulate those voice sounds produced by the vocal cord vibrations and the others are phonemes produced by air currents passing from the lungs to the mouth. The filter frequencies correspond to those of the phonemes that will be produced.

4 Mel General Cepstral

The concept of Mel General Cepstral MGC [16] includes two different voice parameterizations: Mel-Cepstral Analysis and Linear Predictive Coding LPC.

The Mel-Cepstral analysis is quite popular. It was the first effort of the authors when dealing with HTS based speech synthesis. The part which corresponds to LPC is the starting point to the authors proposal using Line Spectral Pair LSP.

Mel General Cepstral parts form the speech signal spectrum H(z) defined as follows:

$$H(z) = S_{\gamma}^{-1}(\textstyle\sum_{P=1}^{N} A_P z^{-P}), \qquad (1)$$

where Sγ is a generalization of the logarithmic function:

$$S_{\gamma} = \begin{cases} \frac{\omega^{\gamma}-1}{\gamma}, \ 0 < |\gamma| \leq 1 \\ \log\omega, \ \gamma = 0 \end{cases} . \qquad (2)$$

Applying this principle to H(z) in equation (1) provides the following information:

$$H(z) = \begin{cases} (1 + \gamma \sum_{P=1}^{N} A_P z^{-P})^{\frac{1}{\gamma}}, 0 < |\gamma| \leq 1 \\ \exp \sum_{P=1}^{N} A_P z^{-P}, \gamma = 0 \end{cases} , \qquad (3)$$

when γ=0, the speech parameterization corresponds to the Cepstrum definition, in which MFCC parameterization is based on. On the other hand, if γ=1 LPC parameterization is obtained.

To convert from LPC to LSP, we define the filter $H(z) = \ 1 + \sum_{P=1}^{N} A_P z^{-P}$ as the sum of two polynomials P(z) and Q(z) [20] each of them is defined as:

$$P(z) = A_p(z) - z^{-(p+1)} A_p(z^{-1}), \qquad (4)$$

$$P(z) = 1 + \textstyle\sum_{p=1}^{P}(a_{p+} a_{P+1-p}) z^{-p} + z^{-(p+1)}, \ (5)$$

$$Q(z) = A_p(z) + z^{-(p+1)} A_p(z^{-1}), \qquad (6)$$

$$Q(z) = 1 + \textstyle\sum_{p=1}^{P}(a_{p-} a_{P+1-p}) z^{-p} - z^{-(p+1)}. \qquad (7)$$

Every polynomial has P/2 pairs of complex conjugate roots for this reason, the above written equations can be represented the following way:

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

$$= (1 + z^{-1}) \prod_{i=1}^{\frac{P}{2}} (1 - 2cos\omega_i z^{-1} + z^{-2}), \qquad (8)$$

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

$$= (1 - z^{-1}) \prod_{i=1}^{\frac{P}{2}} (1 - 2cos\theta_i z^{-1} + z^{-2}). \qquad (9)$$

The values of ω and ϴ represent respectively in P(z) y Q(z) the formant frequencies of the format to synthesize. All of them contained in (0, π) they are known as Line Spectral Frequencies LSF.

5 Using LSP to Synthesize a Speech Signal

The authors decided to test this type of parameterization and adapt it to the current HTS Spanish system. The system by default decomposes the speech signal in Mel General Cepstral Coefficients. It is based on a mathematical concept that unifies MFCC and LSP based on the equation (3) mentioned in the previous section.

This process takes place using SPTK. A technical manual with coding details will be published by the authors.

The authors decided to test it for several reasons: First, LSP is based on Linear Predictive Coding (LPC) which parts form seeing the human vocal tract seen as a filter and the formant frequencies as the filter coefficients. The spectra obtained based on vocal tract models tend to resemble natural speech remarkably. Even more, LSP takes into account more data than LPC which results in a richer quantization of the original speech signal. An LSP voice filter is more stable in nature, the mathematical demonstration can be found in [15]. The size of the audio files is smaller than that of the files using MFCC. Finally, and most important: There are little documented on Speech synthesis using LSP and particularly in spanish, no documentation was found.

6 Evaluation and Comparison of Both Parameterizations

With the purpose of verifying the quality of the synthesized voiced, the authors tested both parameterization techniques: LSP and MFCC [8]. Two aspects were validated: Naturalness and Intelligibility. For naturalness, a MUSHRA Test was performed. To validate intelligibility, five phrases were played to an audience who had to write them down.

6.1 MUSHRA Test

The MUSHRA test [21] is a standard validation test recommended by the International Telecommunications Union. It was specifically designed for the evaluation of different audio codecs. It is organized in a way that the listener analyzes the same audio content codified in different forms including the original recording without compression and the original recording low-passed filtered to anchor the listener to the reference.

A population of 30 listeners was surveyed. All the listeners were either audio specialists or music technology students, since the ITU recommendation requests for experimented listeners. Each person listened to 5 phrases in five different versions: The Original Recording, The Original recording passed through a 100 Hz cut-off filter, a synthesized phrase using MFCC parameterization and a synthesized phrased with LSP parameterization. The subject sat in front of a computer and listened to the phrases through headphones with a SNR of 93 dB. Two aspects on the audio were validated, every phrase had to be qualified by the listener on a 0 to 100 scale according to the norm. At least one phrase had to be qualified with 100.

6.2 Intelligibility Test

Usually, intelligibility tests in speech coding, are focused on proving how easy is for a listener to understand a phrase when the speech signal is masked or filtered. In this case, the interest of the authors was to evaluate how easy the synthetic phrase was to understand depending on the parameterization used.

The best way to validate intelligibility is by dictation. Five LSP and five MFCC synthesized phrases were played to a group of 27 listeners who had to write them down. The listeners written dictations were the individually marked. The given marks to each phrase were correct or incorrect. The MFCC phrases averaged a score of 0.84 whereas the LSP phrases averaged 0.89.

Table 1 shows the obtained mean scores for both aspects. Note that the original reference and its anchor were only used to measure naturalness.

7 Conclusions

As we could learn from the results regarding naturalness and intelligibility - shown in Table I, there is an improvement in both aspects when LSP is chosen as voice parameterization. In terms of file size, LSP speech parameterization files are smaller than MFCC parameterization files. This reduction can be important in terms of data transferring and data storing economization.

The authors consider LSP speech parameterization as a new standard in future works related to speech synthesis in Laboratorio de Tecnologías del Lenguaje FI UNAM.

MFCC parameterization on the other hand is not much below LSP in qualifications. It is widely used in several recognition and synthesis systems. It will be hardly replaced by a speech parameterization which is only a few points ahead in acceptance.

The anchor in the MUSHRA test is used precisely to unconsciously remind the listener what the reference was. Surprisingly it was marked below both parameterizations.

The reference was unmistakably identified by all the listeners. This condition is a reminder that a synthesizer that sounds as natural as a human is still a relevant challenge in the field.

After conducting the experiments described in this document, to new voices were developed using male and female speakers. Both were parameterized with LSP. They have not been statistically validated but early tests showed certain success in intelligibility and naturalness in the authors opinion, their validation remains for future work. Experimenting with different speakers to be synthesized would shed certain light in determining which features should a human voice have to serve as a model for a synthetic voice.

Table 1. Evaluation Results.

Type	Naturalness	Intelligibility
Reference	100	N/A
Anchor	62.6	N/A
LSP Parameter	69.5	0.89
MFCC Parameter	61.4	0.84

Some of the possible failures in imitating human speech are related to the way the phonemes are chosen and concatenated. Adjustments in that stage may lead to an improvement in quality independently of the chosen parameterization.

Current studies on speech synthesis and recognition are walking away from HMM and searching the use of Deep Neural Networks DNN as the new phoneme selection system

Acknowledgements. The current work was sponsored by a CONACYT scholarship and PAPIIT IT 102314 of DGAPA-UNAM. The authors appreciate the suggestions of Dr Felipe Orduña regarding the statistical methods of evaluation.

References

1. Taylor, P. , Black, A.W., Caley R.: The Architecture of the Festival Speech Synthesis System. Proc. 3rd ESCA Work, Speech Synth, 147–151 (1995)
2. Black, A.: CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling. INTERSPEECH (2006)
3. Herrera-Camacho, A., Del Río-Ávila, F.: Development of a Mexican Spanish Synthetic Voice Using Synthesizer Modules of Festival Speech and HTSStraight. Int. J. Comput. Electr. Eng, 36–39 (2013)
4. Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., and Oura, K.: Speech Synthesis Based on Hidden Markov Models. Proc. IEEE, 101(5), 1234–1252 (2013)

5. Tokuda, K., Zen, H., Black, A.: An HMM-based speech synthesis system applied to English. IEEE Speech Synth. Work (2002)
6. Zen, H., Tokuda, K., Black, A.: Statistical parametric speech synthesis. Speech Commun. 1229–1232 (2002)
7. Tokuda, K., Kobayashi, T., Imai, S.: Speech parameter generation from HMM using dynamic features. 1995 Int. Conf. Acoust. Speech, Signal Process, 1(5), 660–663 (1995)
8. Franco, C., Del Rio, F., Herrera, A.: ATINER Conference Paper Series Speech Synthesis of Central Mexico Spanish using Hidden Markov Models. 1–12 (2016)
9. Nakatani, N., Yamamoto, K., Matsumoto, H.: Mel-LSP Parameterization for HMM-based Speech Synthesis. Eurasip Proc. SPECOM 2006 (2006)
10. Lu, H., King, S., Watts, O.: Combining a Vector Space Representation of Linguistic Context with a Deep Neural Network for Text-To-Speech Synthesis (2014)
11. Qian, Y., Yuchen, F., Wenping, H., Soong, F. K.: On the Training Aspects of Deep Neural Network (DNN) For Parametric Synthesis. Microsoft Reasearch (2014)
12. Soong, F., Juang, B.: Line spectrum pair (LSP) and speech data compression. In: ICASSP ’84. IEEE Int. Conf. Acoust. Speech, Signal Process, 9(9), 37–40 (2013)
13. Arakawa, A., Uchimura, Y., Banno, H., Itakura, F., Kawahara, H.: High quality voice manipulation method based on the vocal tract area function obtained from sub-band LSP of STRAIGHT spectrum. ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process - Proc, 2, 4834–4837 (2010)
14. Bäckström, T., Magi, C.: Properties of line spectrum pair polynomials-A review. Signal Processing, 86(11), 3286–3298 (2006)
15. Backstrom, T.: Linear predictive modelling of speech - constraints and line spectrum pair decomposition. Matrix (2004)
16. Tokuda, K., Kobayashi, T., Masuko, T., Imai, S.: Mel Generalized Cepstral Analysis - A Unified Approach to Speeh Spectral Estimation (1994)
17. Taylor, P., Black, A., Caley, R.: The architecture of the Festival speech synthesis system (1998)
18. Black, A., Taylor, P.: Automatically clustering similar units for unit selection in speech synthesis (1997)
19. Young, S.: The HTK Book. J. Chem. Inf. Model. 53(9), 1689–1699 (2013)
20. Zheng, F., Song, Z., Li, L., Yu, W., Wu, W.: The distance measure for line spectrum pairs applied to speech recognition. In: Proc. 5th Int. Conf. Spok. Lang. Process, 1998 (ICSLP ’98), 1123–1126 (1998)
21. Itu-BS.1534: Method for the subjective assessment of intermediate quality level of audio systems Policy on Intellectual Property Right (IPR). Series of ITU-R Recommendations, 1534–3 (2015)