Phonetic Vocoding

Carlos M. Ribeiro[1] Isabel M. Trancoso[2]
INESC, Rua Alves Redol 9, 1000 Lisboa, Portugal

Abstract

A segmental vocoder is a type of coder that explores the correlation between frames in order to achieve significant bit rate savings, in a variable bit rate framework. The coder proposed in this paper falls in the class of segmental vocoders known as phonetic vocoders. Like in other basic LPC vocoders, the transmitter stage performs LPC analysis, and estimates pitch, voicing and energy parameters, on a frame by frame basis. Here, however, the LPC coefficients are fed into a phone recogniser which segments the speech signal, producing a phone index that is transmitted together with the prosodic information. Speaker recognisability is one of the main problems faced by vocoders at the lowest bit rates, given the need to reduce speaker specific information. Hence, phonetic vocoders are very suitable to speaker dependent coding, and can achieve bit rates as low as 250 bit/s. For speaker independent coding, a speaker adaptation methodology may be adopted, although resulting in higher bit rates. The paper will thus discuss a set of trade-offs between synthetic speech quality, speaker recognisability and bit rate (250 to 600 bit/s).

I. INTRODUCTION

The speech coding research community has, for decades, invested a great deal of effort in reducing the bit rate, not only to be able to use communication channels with lower capacity, but also to multiplex more signals in the same channel. The TIA/EIA/IS-96 Standard allows for variable bit rate that can be as low as 1200 bit/s, based on LPC coding with 20 ms frame length. To reduce even more the bit rate, the correlation between frames must be explored, degenerating into a segmental vocoder [6]. This type of vocoder attempts to decompose speech into a sequence of segments. When these segments are identified with phones or phone-like units, the segmental coder is designated as phonetic vocoder.

In the proposed vocoder, like in other basic LPC vocoders, the transmitter stage performs LPC analysis, and estimates pitch, voicing and energy parameters, on a frame-by-frame basis. LPC coefficients are then fed into an HMM phone recogniser which segments the speech signal and produces a phone index. This index and the corresponding phone duration now replace the LPC information and, together with pitch, voicing and energy information constitute the parameters to be transmitted.

Phonetic vocoders can work as low as 250 bit/s if the pitch, voicing and energy are transmitted once per phone. The quality of the synthetic speech, however, is perceptually quite different from the original, being speaker recognisability one of the main problems faced by this type of vocoders, given the need to reduce speaker specific information. Hence, phonetic vocoders are very suitable to speaker dependent coding [6]. For speaker independent coding, some type of speaker adaptation may be performed. Gender dependent synthesis based on average pitch is a bit free option, but insufficient to characterise the speaker.

One possible method to overcome this problem is to choose the best codebook from a set of multiple-speaker codebooks [2]. Another one is to adapt the codebook to the input speaker [8]. The latter type of approach is the one used in the present work, and will be described in Section II.

The following sections will be devoted to the basic building blocks of our phonetic vocoder. Section III will describe the codebook design stage. Section IV will briefly introduce the HMM recogniser. Section V will present a set of alternative quantization schemes, from segmental to frame-by-frame based. Finally, Section VI will present our main conclusions and discuss further work.

In spite of being one of the basic building blocks of the coder, the synthesiser will not be described in this paper. In fact, we have implemented both conventional LPC vocoder synthesis and magnitude-only harmonic synthesis. Both can be used in this framework, as interoperability [5] between the two schemes may be obtained by deriving the amplitude of the harmonics from the LPC filter.

II. SPEAKER ADAPTATION

A. Speaker Modification

The adaptation strategy we have followed is based on the speaker modification work described in [9]. The authors introduced a method of altering the formant frequencies of vowel segments using LPC analysis/synthesis. The pole location modification is based on statistical references and provides individual control over formant frequencies and bandwidths. The method is based on collecting statistics of the radius and angle of the poles associated with formants, for each frame corresponding to the same vowel class, and for each speaker. LPC analysis is performed on the utterance from the source speaker that we want to modify, in order to make it sound as spoken by the target speaker. Each pole of the linear prediction polynomial (expressed in its polar form $re^{j\theta}$) is then moved toward the mean of the target speaker for that particular class by zscore normalisation:

[1] Also with ISEL-CEDET-DEEC
[2] Also with IST

The authors would like to thank their colleagues Drs. Céu Viana and Isabel Mascarenhas (CLUL), for their suggestions and their hard work in the manual correction of the EUROM.1 corpus labelling, to Rui Amaral (INESC) for the help in the HMM segmentation and also to Prof. Luís Oliveira (INESC/IST), for many fruitful discussions.

$$r' = (r - \bar{r}_{source}) / \sigma_{rsource} \quad (1a)$$

$$\theta' = (\theta - \bar{\theta}_{source}) / \sigma_{\theta source} \quad (1b)$$

where $\bar{r}_{source}$ and $\bar{\theta}_{source}$ are the mean values of the radius and angle, respectively, for the source speaker, computed along the phone segments, and $\sigma_{rsource}$ and $\sigma_{\theta source}$ are the corresponding standard deviations. The pole modification is achieved by introducing the target speaker statistics:

$$r_{\mathrm{mod}} = r' \times \sigma_{rtarg} + \bar{r}_{targ} \quad (2a)$$

$$\theta_{\mathrm{mod}} = \theta' \times \sigma_{\theta targ} + \bar{\theta}_{targ} \quad (2b)$$

For simplicity sake, we have omitted the pole index in the above expressions. After reconstructing the modified linear prediction polynomial, LPC synthesis is performed using a modified residual.

B. Proposed Speaker Adaptation Scheme

Our adaptation strategy is slightly different, being based on the modification of LSF (Line Spectrum Frequencies) coefficients, instead of directly modifying formant frequencies and bandwidths. The two sets of parameters are intimately related: as discussed in [1], the closer two consecutive LSF coefficients are together, the narrower the bandwidths of the corresponding pole of the vocal tract filter. Hence, formants are marked by two close LSF coefficients, whereas spectral tilt is primarily marked by LSF coefficients, which are further apart. The roots of P(z) have been named as position coefficients, because the closed glottis model is the best approximation for a lossless approximation of the vocal tract filter. Hence, whenever formants are present, one can find a correspondence between the roots of P(z) and the locations of the formant frequencies. The roots of Q(z), on the other hand, have been called difference coefficients, because of their role in marking the presence and absence of a formant by their closeness to a position coefficient.

In order to implement this speaker adaptation strategy, equations (1) and (2) are rewritten using LSF's coefficients:

$$LSF_{\mathrm{mod}} = \alpha + \beta\, LSF \quad (3)$$

where the scale factor β is the quotient between the target and source standard deviation values and α is the difference between the average target value and the average value of the scaled source. This modification, however, does not correspond to the minimisation of the mean squared error between the target and modified vector parameters, which can be shown to satisfy:

$$\beta = \frac{C_{source,target}}{C_{source}} \quad (4a)$$

$$\alpha = \overline{LSP}_{target} - \beta\, \overline{LSP}_{source} \quad (4b)$$

where $C_{source,target}$ corresponds to the covariance between the source and the target, and C_{source} corresponds to the autocovariance of the source.

We found that speaker characteristics are essentially preserved by matching the LSF average values, and that the scale factor β has a minor perceptually effect in the output speech, and can be set to one, reducing the transmitted parameters and consequently the bit rate [7]. In order to implement this speaker adaptation strategy, it is necessary to transmit the average values of each LSF coefficient over the whole duration of the phone, which act as target values.

This extra speaker specific information is only transmitted for vowel and glide phones, where the speaker characteristics are perceptually more important. In the receiver stage, the phone index together with the previous and next indexes, are used to retrieve a set of LSF coefficients from a context-dependent codebook. The restored LSF coefficients are then used, together with energy and pitch information, to produce the synthetic speech.

III. CODEBOOK DESIGN

The codebook was generated using speech data from 8 of the 10 speakers of the *few talkers* subset of the EUROM.1 corpus for European Portuguese, collected in the SAM_A ESPRIT project. This data (15 passages from each speaker) was hand-labelled using 53 phone labels, and corresponds to about 32 minutes of speech, after removing silences. The codebook includes one normalised codeword for each different context-dependent phone. Each codeword is computed as the 10^{th} order LSF centroid of the most likely duration (measured in number of frames, using a frame update period of 11.25 ms), for a given context-dependent phone. This averaging procedure avoids mixing phones with different duration.

This codebook includes one codeword for each index jth phone, of dimension L_j (p+2), where L_j is the duration of the stored j^{th} phone in terms of number of frames, p the LSF order and the additional terms are the RMS per frame and the average value for each LSF coefficient, stored for quantisation purposes, as explained latter. An example of this codeword is shown in Table 1.

Table 1
Example of codeword /i/
with left context /l/ and right context / J/

Frame→	1	2	3	4	5	Average
LSF1	279	261	263	265	268	267
LSF2	375	343	325	320	320	337
LSF3	918	794	875	953	967	901
LSF4	1652	1788	1711	1660	1488	1660
LSF5	1936	2100	2185	2251	2233	2141
LSF6	2207	2331	2395	2446	2498	2375
LSF7	2405	2465	2531	2635	2677	2543
LSF8	2731	2828	2902	2950	2979	2878
LSF9	2940	2963	3022	3088	3169	3036
LSF10	3529	3447	3459	3523	3583	3508
RMS	2531	2605	2565	2550	2297	2510

Time scale modification is adopted to adjust the duration of the normalised phone in the codebook to the transmitted

duration. For vowel and glide phones, the LSF coefficients are restored by matching the transmitted input values that carry speaker information, according to equation (3) but with $\beta=1$, and relying in the dynamic characteristics stored in the codebook. Instability in the resulting LPC filter is checked and filter is forced to be stable. For consonants, the retrieved codeword is used without modification.

Only 6089 different context-dependent phones were found in this training data. In European Portuguese, as in many other languages, the number of existent left and right context-dependent phones is much larger. When a context-dependent phone (of the test data) is not present in the codebook, it must be replaced by an existing context. However, replacing by a very different context (like replacing the /a/ in context /p-a+s/ by the same phone in the context /w-a+p/) can create artefacts in the output speech. A phonetically based procedure was, therefore, implemented to search the codebook for the closest phone context

Some artefacts can arise from phone concatenation, due to strong variations in the spectrum. This effect can be minimised by storing two additional frames for each codeword (not shown in Table 1), one in the left and another in the right context, that allow for an interpolation segment of two frames between adjacent phones.

The proposed adaptation strategy proved to be effective even for male to female voice adaptation or vice-versa. Nevertheless, strong prosodic variations due to changes in fundamental frequency can also introduce artefacts in the speech output signal. To minimise this problem, duplicate codebooks were built one for male and another for female speakers. The procedure for choosing the gender-dependent codebook is based on the average values of the pitch frequency, computed for each of the last three voiced phones. An average value greater than a given threshold is considered to be from a female voice, and smaller than that is from a male voice. A majority rule is then used for the decision. The threshold value (150 Hz) was computed based on the statistics of the training data. With this heuristic decision, the transmission of information about the speaker gender is therefore avoided.

IV. HMM SEGMENTATION

The segmentation procedure we have used is based on state-of-the-art HMM (Hidden Markov Models) phone recognition. The recogniser uses 3-state, 3-mixture-per-state models. Each input vector has 30 coefficients (14 mel-cepstra, 14 delta-mel-cepstra, energy and delta-energy).

53 HMM context-independent models were trained with the same hand-labelled data used to create the codebook, i.e. 32 minutes of speech from 4 female and 4 male speakers.

The 2 remaining speakers of the *few talkers* subset of the EUROM.1 corpus (totalling 8 minutes of speech) were used for testing the recogniser, yielding 37% recognition errors.

The amount of hand-labelled data we have is, in fact, too small, either for training HMM models or for creating LSF codebooks, indexed by triphones. Hence, a bootstrap procedure can be adopted to automatically align more data, using the context-independent models. This data can then be used to retrain models, create new codebooks or assess the perceptual quality of the coder.

In spite of the 37% recognition errors, a sufficiently good acoustic matching is generally obtained. This type of errors does not seriously degrade the subjective speech quality of phonetic vocoders [6]. This idea was confirmed by a perceptual comparison between synthetic speech produced using hand-labelled and automatically segmented speech. Nevertheless, the use of different parameters in recognition (mel-cepstra) and synthesis (LSF) may not be as efficient as "unified" approaches [3].

Another approach we have tried which significantly reduced the recognition errors is the use of context-dependent phone models. The use of the right context-dependent phone, however, may be prohibitive in terms of delay. This approach yielded a reduction of 25% (43 to 32 %) in recognition errors, in experiments using cepstral coefficients [7], and is currently under test using mel-ceptral coefficients.

These results have been obtained using ergodic models. A continuous speech recogniser trained for European Portuguese with corresponding pronunciation lexica and language models would improve the above mentioned recognition score, but has not been implemented yet.

V. QUANTIZATION

The set of parameters to be quantised and transmitted comprises: the index and duration of each phone; the LSF average values in vowel and glide regions; the energy, the voiced/unvoiced decision (V/UV) and the pitch period. The quantisation schemes to be described were tested with all the automatically aligned sentences of the *many talkers* subset of EUROM.1, corresponding to 50 speakers (originally 60 but 10 were common to the *few talkers* subset). This amounts to 18 minutes of speech, after removing silences. Each parameter is quantised as explained below:

A. Phone Index

The 53 phones are coded using the Shannon-Fano memoryless source coding. The entropy computed on the basis of the phone probabilities for the training corpus was 5.23 bits per phone. Our quantiser achieves 5.27 bits per phone, which corresponds to 99% efficiency. The most likely phones are coded with 4 bits and the least likely phones with 9 bits.

The minimum, average and maximum values of the phone rate and vowel-glide rate obtained in the test *corpus*, after removing silences, are listed in Table 2.

Table 2
Minimum, average and maximum values for phone and vowel-glide rate.

	Minimum	Average	Maximum
Phone/s	10	13	14
Vowel-glide/s	3	5	6

For an average of 13 phones per second, the phone index was quantised with 76 bit/s in the test *corpus*.

Further bit rate savings could be achieved using language modelling (e.g. phone bigrams or trigrams), to explore the phonotatics of the language.

B. Duration

The phone duration is measured in terms of number of frames (11.25 ms long). The entropy computed for this parameter in the training corpus was 3.64 bits per phone. 68% of the phones in this corpus have a duration of 1 to 6 frames. These duration values are coded with 3 bits, and larger values are coded using Huffman tree coding. The average number of bits per phone achieved with this quantization scheme is 3.67, which corresponds to 99% efficiency. For an average rate of 13 phones per second (excluding silences) in the test *corpus*, the value obtained is 53 bit/s.

The phone duration, as the energy, is correlated with the type of phone, but it is difficult to take advantage of this dependence, as a speaker may have a higher or lower speaking rate than average. The same can also be said about the loudness of the sentence.

C. Differential coefficients

An estimate of the expected value of each average i^{th} LSF coefficient for the j^{th} phone is the average values of the corresponding codeword (Table 1), assuming a stationary process. The difference between the computed LSF average values and those expected values carries the speaker information, and its dynamic range is smaller than the one of the computed values. It can, therefore, be transmitted more efficiently. In the receiver, of course, these values must be added to the expected values in order to restore the average LSF values.

This procedure can be interpreted as a vector-scalar quantiser, with as many vectors as vowel and glide phones. The index to the vector part of this quantiser is the phone index that must be transmitted anyway. The scalar part, that is, the variation relatively to the expected values, contains the speaker information needed to adapt the codeword.

Each LSF differential coefficient is quantised with 2 bits, using the LBG algorithm [4], resulting in a total of 20 bits for the scalar part of the quantiser. For an average rate of 5 vowel and glides per second, this corresponds to about 100 bit/s. The value obtained with the test *corpus* was 92 bit/s.

D. Energy

An optimal quantization table for this parameter was also derived using the LBG algorithm. The energy value is quantised every other frame, i.e., every 22.5 ms. Interpolation is used to reconstruct the frame by frame information. Energy information for the first and last frames of each phone is always transmitted, as amplitude variations are perceptually more important there. This scheme results in an update rate slightly higher than 44.4 double-frames per second. For a 4-bit quantiser, the average value achieved in the training *corpus* is 191 bit/s.

The transmission of this parameter effectively consumes the biggest slot of the bit rate, as can be observed in Table 6. In order to reduce this slot, the average value of the energy can be transmitted only once per phone, and a procedure equivalent to the one used for LSF quantization can be used to restore the values on a frame by frame basis. For a 5-bit quantiser and an average rate of 13 phone/s, this corresponds to about 65 bit/s. With different quantisation tables for consonants and vowels, it is possible to quantise the vowel energy with 4 bits and the consonant energy, perceptually less important, with only 3 bit per phone. For 8 consonants and 5 vowels per second, this result in 44 bit/s. The value obtained with the test *corpus* was 42 bit/s. These alternative segmental quantisation schemes, however, are less robust than the frame-by-frame quantization, even using interpolation between phones.

E. Voicing and pitch information

Voicing decision and pitch are typically quantised together, using 7 bits per frame (double-frame, in our coder). One of the 128 possibilities is used to indicate the presence of an unvoiced frame, and the remaining 127 to quantise pitch lags (20 to 146 sampling periods). With 44.4 double-frames per second, this quantisation scheme results in 311 bit/s.

A differential scheme was also adopted for transmitting the pitch information in each double-frame: if the previous frame is unvoiced, 1 bit is transmitted to inform if the present frame is voiced or unvoiced; if the present frame is voiced, 7 more bits are transmitted with the estimated pitch value. If the previous frame is voiced, the current pitch estimate is differentially encoded using only 3 bits. However, the quantisation levels are not selected as consecutive values, in order to increase the dynamic range. The 8 hypotheses correspond to the unvoiced code and differences of {-9, -6, -3, 0, 3, 6, 9}. This differential scheme results in 125 bit/s in the training *corpus*, less than half the value obtained with absolute quantization.

Further reduction can be achieved if this information is transmitted only once per phone. Although with less quality and robustness, for 13 phones per second and up to 9 voiced phones, results in 76 bit/s (9x8+4). The value obtained with the test *corpus* was 73 bit/s. A phone is considered voiced if the majority of its frames are voiced. The transmitted pitch is the average value. In the receiver, all the phone frames are considered either voiced or unvoiced, and pitch interpolation between voiced phones is adopted.

F. From 250 to 600 bit/s

The parameters that are transmitted once per phone, that is, on a segmental basis, resulted in the average bit rate values indicated in Table 3. Adaptation was performed in 37% of the phones. Table 4 summarises the different bit rates that can be achieved for the remaining parameters (energy and pitch plus voicing decision), if these parameters are quantised

on a frame-by-frame or on a segmental basis. The frame-by-frame scheme can be used for highest quality but also higher bit rate.

Table 3
Average values of bit rate for each quantisable parameter (segmental level).

	Average bit rate
Phone index	76
Phone duration	53
LSF differential values	92
Segmental parameters (total)	221

Table 4
Average values of bit rate for energy, and pitch + V/UV.

	Absolute	Differ.	Segmental
Energy	191	--	42
Pitch, V/UV	311	125	73

This was the scheme used in Table 5, totalling an average bit rate of 537 bit/s, corresponding to a minimum value of 474 bit/s and a maximum value of 579 bit/s.

Table 5
Average values of bit rate for each quantisable parameter - pitch+V/UV and energy on a frame-by-frame basis.

	Quantization	Average bit rate
Phone index	Segmental	76
Phone duration	Segmental	53
LSF differ. Values	Segm (Vowels)	92
Energy	Frame	191
V/UV and pitch	Differential	125
Total		537

The segmental scheme can be used for the lowest bit rates, as suggested in Table 6. In order to obtain this low average value of 244 bit/s, speaker adaptation was discarded. In between these two extremes, one may consider intermediate schemes, which provide a good trade-off between speech quality and speaker recognisability, on one hand, and bit rate on the other. For instance, if one adds the 92 bit/s corresponding to speaker adaptation to the quantisation scheme of Table 6, one obtains an intermediate average bit rate of 336 bit/s.

Table 6
Average values of bit rate for each parameter - full segmental quantization.

	Quantization	Average bit rate
Phone index	Segmental	76
Phone duration	Segmental	53
LSF differ. Values	No	0
Energy	Segmental	42
V/UV and pitch	Segmental	73
Total		244

VI. CONCLUSIONS AND FURTHER WORK

This paper presented a phonetic vocoder based on speaker adaptation with a set of quantization schemes which allows a trade-off between quality and bit rate, in a range of 250 to 600 bit/s.

One of the main limitations that we faced was the reduced size of the hand-labelled spoken material, namely for codebook design and coder assessment. As better alignment methods are presently being developed in the scope of other projects, this limitation will soon become less relevant.

We are currently testing methods to better control the LPC filter instability that can arise from the speaker adaptation procedure; as well as methods to explore the speaker specific information which has been previously transmitted in order to adapt present phones. Another area of work is the use of the same parameters in the recognition and synthesis stages, in order to improve robustness in the presence of recognition errors. Finally, a set of formal assessment tests including speaker recognisability is also planned for the near future.

V. REFERENCES

[1] J. R. Crosmer, T. P. Barnwell, “A Low Bit Rate Segment Vocoder Based on Line Spectrum Pairs”, Proc. Int. Conf. Acoust., Speech, Signal Proc., pp 240-243, 1985.

[2] P. Jeanrenaud, P. Peterson, “Segment Vocoder Based on Reconstruction With Natural Segments”, Int. Conf. Acoust., Speech, Signal Proc., pp 605-608, 1991.

[3] W. J. Holmes, “Towards a Unified Model for Low Bit-Rate Speech Coding Using a Recognition-Synthesis Aproach”, Proc. ICSLP’98.

[4] Y. Linde, A. Buzo, R.. Gray, “An Algorithm for Vector Quantization”, IEEE Trans. Comm., pp 84-95, 1980.

[5] R. J. McAuley, T. Champion, “Improved Interoperable 2.4 kb/s LPC using Sinusoidal Transform Coder Techniques”, Proc. Int. Conf. Acoust., Speech, Signal Proc., pp. 641-643, 1990.

[6] J. Picone, G. Doddington, “A Phonetic Vocoder”,- Proc. Int. Conf. Acoust., Speech, Signal Proc., pp 5809-583, 1989.

[7] C. Ribeiro, I. Trancoso, "Phonetic Vocoding with Speaker Adaptation", ", Proc. 5th European Conference on Speech Communication and Technology, 1997.

[8] S. Roucos, A. Wilgus, “Speaker Normalisation Algorithms For Very Low Speech Coding”, Proc. Int. Conf. Acoust., Speech, Signal Processing, pp 1.1.1-1.1.4, 1984.

[9] J. Slifka, T. Anderson, “Speaker Modification With LPC Pole Analysis”, Proc. Int. Conf. Acoust., Speech, Signal Processing, pp 644-646, 1995.