SPLIT TEMPORAL DECOMPOSITION AND QUANTISATION

C. H. Ritz, I. S. Burnett
Whisper Laboratories, TITR,
University of Wollongong, NSW, Australia
chritz@st.ele.uow.edu.au, i.burnett@elec.uow.edu.au

ABSTRACT

Standard temporal decomposition derives models for the speech spectral parameter vectors without considering the perceptual significance of the vector elements. To overcome this drawback, Split Temporal Decomposition divides the parameter vector (in this case Line Spectral Frequencies (LSFs)) into sub or split vectors for which separate event functions are determined. Hence, multiple event functions are derived for the overall parameter vectors and these are shown to provide a distinct improvement in the modeling of such vectors. We also show that by using a joint event codebook for the sub-vectors, improvements in LSF vector quantisation can be achieved. In particular, this allows the emphasis of perceptually important LSF vector elements within the quantisation scheme.

1. INTRODUCTION

Temporal decomposition (TD), originally proposed by Atal (1993), is a method for achieving very low bit rate speech compression. It relies on the idea of being able to represent speech as a series of time overlapping events, which can be related to the different phonemes produced during speech (Van Dijk-Kappers & Marcus, 1989, Deleglise, Bimbot, Montacie, & Chollet, 1988).

In TD, vectors of speech parameters (in this paper, Line Spectral Frequencies (LSFs)), are represented as a weighted sum of event functions and can be described by expression (1) for a segment of length N

$$\hat{y}_i(n) = \sum_{k=1}^{K} a_{ik}\phi_k(n), \quad 1 \le n \le N, 1 \le i \le p, \qquad (1)$$

Here, $\hat{y}_i(n)$ is the approximation of speech parameter $y_i(n)$ produced by the model, $\phi_k(n)$ is the *k*th event function at time n, a_{ik} is the *i*th target vector for *k*th event function, p is the number of LPC parameters per frame and *K* is the number of event functions found for the segment.

Standard TD analyses the entire LSF vector and derives a single event function that best represents all elements in the vector. Split TD applies the analysis separately to different sub-vectors of the total LSF vector, where the sub-vectors are chosen based on their perceptual significance. Hence separate event functions are derived for each sub-vector resulting in multiple event functions for the entire LSF vector. As with split vector quantisation, this technique attempts to model more perceptually important LSFs more accurately.

For analysing each sub-vector, the restricted temporal decomposition method (Kim & Oh, 1999) is used here. In this method, events and targets are located at stable points of the LSF vector trajectories (using the stability criterion described by Kim & Oh (1999)) and derived by minimising the mean squared error between the original vectors and the approximated vectors of expression (1).

The next section describes split TD with two sub-vectors. Section 3 describes quantisation of the event functions while Section 4 presents results of this method including a discussion on event quantisation. Conclusions drawn from these results are provided in Section 5.

2. SPLIT TD WITH TWO SUB-VECTORS

A number of methods can be used to split the LSF vector, but in this paper, a simple split into two sub-vectors is investigated. Expressions (2) and (3) describe such a split, where each represents the more and less perceptually important sub-vectors, respectively. Here K and L are the number of events occurring in each segment of length N.

$$\hat{y}_i(n) = \sum_{k=1}^{K} a_{ik}\phi_k(n), \quad 1 \le n \le N, 1 \le i \le I, \qquad (2)$$

$$\hat{y}_j(n) = \sum_{l=1}^{L} a_{jl}\phi_l(n), \quad 1 \le n \le N, 1 \le j \le J. \qquad (3)$$

For the simple 4-6 split investigated in this paper, the lower 4 LSFs are modeled by (2) and the upper 6 LSFs are modeled by (3), with I and J equal to 4 and 6, respectively. This split can be justified by the knowledge that the lower order LSFs are more perceptually important than the higher order LSFs (Paliwal & Atal, 1993). Hence this split ensures the lower order LSFs are quantised more accurately than the higher order LSFs. Note that other splits can be used, but for the purpose of this paper only the 4-6 split is investigated. In split vector quantisation, this was found to achieve the best results (Paliwal & Atal, 1993), and so was chosen for the initial investigation into split TD.

Two approaches are suggested – aligned Split Temporal Decomposition (aligned-STD) and nonaligned Split Temporal Decomposition (nonaligned-STD). In aligned-STD, events are located using the entire LSF vector while in nonaligned-STD, events are located separately for each sub-vector. In both these methods, events and targets are derived using the technique described in the previous section.

3. EVENT FUNCTION QUANTISATION

In the temporal decomposition technique used in this paper (Kim & Oh, 1999), event shapes are restricted to vary between 0 and 1. Quantisation of the event functions can be achieved by vector quantising the event shapes using a codebook containing representative event function vectors. In this paper the codebook of event vectors is obtained using the k-means algorithm.

3.1 Resulting Bit Rate

One of the drawbacks of using multiple event functions is the increase in bit rate required for quantisation. As an example, consider the case when the event rate is around 15 events/sec for standard temporal decomposition, aligned-STD, both sub-vectors of nonaligned-STD. If 7 bits are used to code event shapes, 4 bits for the widths and 33 bits for the targets, similar to that used by Kim & Oh (1999), the bit rates for transmission become 660 bps, 825 bps and 765 bps for standard temporal decomposition, nonaligned-STD and aligned-STD, respectively. Non-aligned-STD requires more bits than aligned-STD since, in the former case, the width for each event of the split is transmitted, while in the later case, only one width is transmitted for both events of the split.

3.2 Reducing the Bit Rate of Aligned-STD

One way to reduce the bit rate required for transmitting the event function shapes in aligned-STD is to vector quantise the event shapes for each sub-vector simultaneously using a single codebook. For the 4-6 split used here, each entry in the codebook is then the combination of two vectors

representing the lower and upper LSF event shapes. Due to the nonalignment of events in nonaligned-STD, this technique is only applicable to aligned-STD. Hence when quantising the event functions, aligned-STD is the preferred approach.

4. RESULTS

A 10^{th} order Linear Prediction (LP) analysis was performed on 25 ms frames of speech sampled at 8 kHz with a 5 ms overlap. The resulting LPCs were converted to LSFs and linearly interpolated to generate two LSF vectors per frame, prior to derivation of event functions and target vectors.

Figure 1 shows the event functions derived using standard TD, aligned-STD and non-aligned-STD for the LSFs corresponding to a short segment of speech. The shape of events for both nonaligned-STD and aligned-STD are quite different compared with those for standard TD. The difference in locations for nonaligned-STD concurs with the assertion (Naranjin & Fallside, 1989) that vocal tract parameters move with unequal velocities.

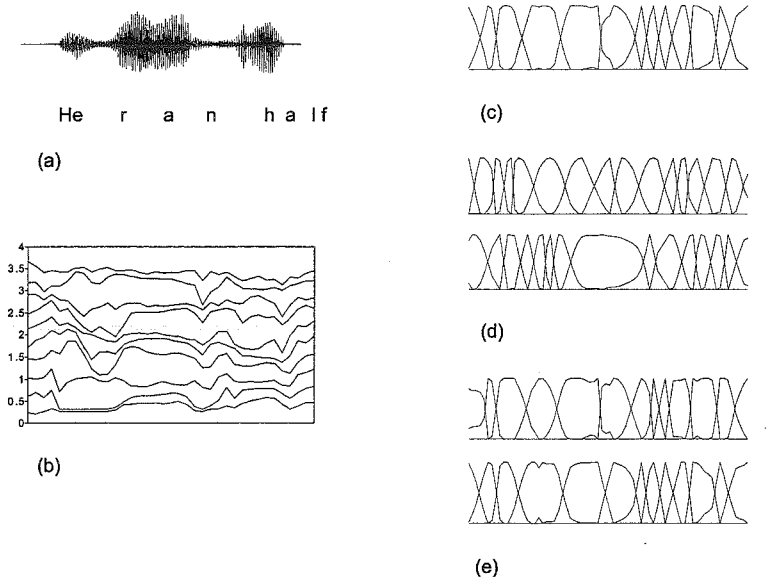

Figure 1. Event functions for a segment of speech from a male speaker a) speech waveform for the segment b) corresponding LSFs (in kHz) c) events for standard TD d) events for nonaligned-STD e) events for aligned-STD

4.1 Performance Evaluation

A wide, mixed set of male and female sentences was used for analysing the performance. An initial performance analysis was achieved via the average log spectral distortion (Kondoz, 1994). The log spectral distortion is defined in (4) while the average log spectral distortion was evaluated by averaging the log spectral distortion over M frames using expression (5). In (4) P and P' are the original and reconstructed LPC power spectrum corresponding to the LSFs for the current frame and evaluated at F/2 discrete frequencies.

$$sd = (F/2)\sum_{f=0}^{F/2-1}\left[10\log_{10}\frac{\left|P'(f)^2\right|}{\left|P(f)^2\right|}\right]^2 \qquad (4)$$

$$SD = (1/M)\sum_{m=1}^{M}\sqrt{sd(m)} \qquad (5)$$

The average spectral distortion between the original LSFs and the reconstructed LSFs for standard temporal decomposition, nonaligned-STD and aligned-STD was found to be 1.49 dB, 1.37 dB and 1.35 dB respectively. Further analysis was performed via the weighted mean squared error between the original and reconstructed LSFs, where the weights are described by Paliwal & Kleign (1995). Nonaligned and aligned-STD achieved, respectively, a 25% and 18% reduction in these errors per LSF, compared with standard TD. These results indicate an improvement in the modeling accuracy of both aligned-STD and non-aligned-STD compared with standard TD.

To further analyse the performance improvement for each sub-vector, Table 1 shows the percentage reductions in the mean squared reconstruction error for the relevant sub-vectors and the whole vector for both aligned-STD and nonaligned-STD. Note that spectral distortion or a weighted mean squared error cannot be used as a performance measurement since each sub-vector only contains part of the LSF vector. These results show a significant improvement in the reconstruction error for each sub-vector, with nonaligned-STD providing the most improvement.

	Percentage Reduction in the Mean Squared Error		
	LSF(1:4)	LSF(5:10)	LSF(1:10)
(nonaligned-STD)	30.1	19.9	24.5
(aligned-STD)	23.1	12.9	17.2

Table 1. Percentage Reductions in mean squared errors for the sub-vectors and vectors nonaligned-STD and aligned-STD.

4.2 Quantisation Of Event Functions

The event functions derived for aligned-STD with a 4-6 split were quantised using the technique described in Section 3.2, for which a 7-bit codebook was trained. For comparison purposes, a 7-bit codebook consisting of single event functions derived from standard TD was also trained on the same dataset.

Informal tests found that the spectral distortions over a small number of speech files between the original LSFs and those reconstructed using the quantised event functions for both aligned-STD and standard TD were similar. However, a more significant result is obtained by measuring the mean squared reconstruction error for the sub-vectors. It was found that these errors for the upper six LSFs were similar or slightly worse for aligned-STD using this new codebook, while for the lower four LSFs these errors showed an average 7.5% improvement. This gives the desired increase in modeling accuracy for the lower order LSFs without an increase in the bit rate compared to standard TD.

5. CONCLUSION

Both nonaligned-STD and aligned-STD produce different event functions compared with those of standard temporal decomposition. This indicates that standard TD is sub-optimal in modeling the perceptually important parameters. This is confirmed by reductions in the reconstruction errors for the whole LSF vector as well as for the sub-vectors when using split TD.

When quantising the event functions using standard vector quantisation of each event, the bit rate required for transmission of both split approaches increases. To reduce the bit rate in aligned-STD, the event functions were quantised jointly. This achieved similar spectral distortions for the entire LSF vector, but improved the modeling accuracy of the perceptually important lower order LSFs. Since this technique cannot be applied to nonaligned-STD, aligned-STD is the preferred approach.

Split Temporal Decomposition could also be applied to other forms of temporal decomposition. One example would be the technique suggested by Ghaemmaghami & Deriche (1996), which approximates events by known functions and thus requires zero bits for quantisation of the shapes. Using multiple event functions in this approach could also give improvements in perceptual quality without an increase in bit rate.

ACKNOWLEDGEMENTS

C.H. Ritz is in receipt of an Australian Postgraduate Award and a Motorola (Australia) Partnerships in Research Grant.

REFERENCES

Atal, B. S. (1983), "Efficient coding of LPC parameters by Temporal Decomposition", *Proc. ICASSP'83*, pp. 81-84, Boston.

Deleglise, P., Bimbot, F., Montacie, C. and Chollet, G. (1988), "Temporal Decomposition and Acoustic-Phonetic Decoding of Speech", *Proc. ICASSP'88*, vol. 1, pp. 445-448.

Ghaemmaghami, S. and Deriche, M. (1996), "A new approach to very low-rate speech coding using temporal decomposition", *Proc. ICASSP'96*, pp. 224-227, vol. 1.

Kim, S. J. and Oh, Y.H. (1999), "Efficient quantisation method for LSF parameters based on restricted temporal decomposition", *Electronics Letters*, vol. 35, issue 12, pp. 962-964.

Kondoz, A.M. (1994), *Digital Speech, Coding for low bit rate communication systems*, John Wiley and Sons, pp. 102-104.

Naranjin, M. and Fallside, F. (1989), "Temporal decomposition: a framework for enhanced speech recognition", *Proc. ICASSP'89*, vol. 1, pp 655-658.

Paliwal, K.K. and Atal, B.S. (1993), "Efficient Vector Quantisation of LPC Parameters at 24 Bits/Frame", IEEE Transactions on Speech and Audio Processing, Vol 1., No. 1, p.p. 3-14.

Paliwal, K.K. and Kleijn, W.B. (1995), "Quantization of LPC Parameters", *Speech Coding and* Synthesis, pp. 452-453, Elsevier Science.

Van Dijk-Kappers, A. M. L. and Marcus, S. S. (1989), "Temporal decomposition of speech", *Speech Communications*, vol. 8, no. 2, pp. 125-135.

HIERARCHICAL SPEECH COMPRESSION FOR STORAGE - A TWO-LEVEL APPROACH

Peter Veprek and Alan B. Bradley
Department of Communication and Electronic Engineering
RMIT University, Melbourne, Australia

Abstract – In this paper, we investigate speech compression for storage. We propose a compression method operating on several levels of the speech signal hierarchy. The method is suitable for off-line compression of large closed corpora with known transcription. A realisation of the method using two levels of hierarchy is presented and evaluated. Results show that the method can provide high quality of reconstructed speech at low data rate for sufficiently large corpora.

INTRODUCTION

Structure of Speech

Speech signals, when recorded and digitised, can be analysed at various levels within an overall speech production hierarchy. First, there are samples of the signal itself. As such, individual samples are not of particular interest with the exception of waveform coding strategies (Deller et al., 1993).

The next level up in the hierarchy is the pitch epoch in the case of voiced speech and a frame in the case of unvoiced speech. Voiced speech is characterised by periodically vibrating vocal cords. This vibration modulates the flow of air passing through the vocal tract. The pitch period is the time between successive vibrations of the chords and the corresponding segment of the speech waveform is often referred to as a pitch epoch. Unvoiced speech can be segmented into (usually fixed-size) frames of similar duration. Due to the quasi-stationary nature of speech, each epoch can be described using the source-filter model by an excitation pulse and a set of formants or simply taken as a short waveform segment. It is clear that the epoch and frame repertoire of a single speaker is defined by the combination of different excitation pulses and vocal tract configurations and, although made up of a large set of possible states, it remains a limited set. Epochs and frames are often used as the basic units of speech production within speech synthesisers, coders, and voice converters (Hamon et al., 1989; Valbret et al., 1992; Veprek & Bradley, 1999).

Beyond the epoch and frame level analysis is the defined sequence of epochs and/or frames. While arbitrary sequences could be selected, diphones are commonly chosen as the unit at this level. By definition, each diphone represents the transition from one phoneme to another. The minimum number of diphones (in English) required for speech production is in the order of 1,700-1,800 and more if different acoustic and/or prosodic realisations of diphones are considered. The direct use of phonemes as a building block for speech production is an alternative, of course, with the benefit of having significantly fewer units (less than 100). However, due to strong coarticulation (particularly in English), phonemes are not adequate as a unit for speech generation and are consequently of little interest in the light of this research work.

Moving further up in the speech production hierarchy, a number of often overlapping, higher level units exist. To name a few, demisyllables and other sub-word units are frequently used in speech synthesisers (Pearson et al., 1998). Occasionally, entire words and/or phrases and their portions are considered as acoustic and/or prosodic units in some systems (Holm & Hata, 1998; Donovan et al., 1999; Lewis & Tatham, 1999; Stober, 1999). In addition, the speech signal can be examined from a number of different perspectives: acoustic, prosodic, linguistic, and so forth.

Traditional Speech Coding

Traditional speech coding techniques divide the digitised speech signal into short segments and encode these segments using diverse parameterisation techniques. The segment size varies from a single sample (in many waveform coders), through a fixed-size frame (in numerous linear-prediction based vocoders), to longer, often variable-length units such as phonemes (in several segment and phonetic vocoders). In most cases, a single type of segment is used although some techniques exploit redundancies at more than one level (e.g. adjacent sample and adjacent pitch cycle predictors utilised in some coders).

It is evident that typical speech coding techniques do not fully utilise the rich structural content of speech.