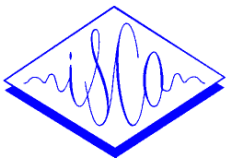

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

LOW-RATE SPEECH CODING BASED ON TIME-FREQUENCY INTERPOLATION

Yair Shoham

Signal Processing Research Department
AT&T Bell Laboratories
600 Mountain Ave.
Murray Hill, NJ 07974

ABSTRACT

This paper presents a new algorithm for high-quality speech coding and demonstrates the advantage of the proposed coder over the conventional CELP algorithm for low rate coding. The paper proposes an empirical but perceptually advantageous framework for voiced speech processing, called Time-Frequency Interpolation (TFI). The general formulation of the TFI technique is given first. Then, a 4.2 Kbps speech coder, based on TFI, is described. The performance of this coder is demonstrated in terms of formal MOS scores. It is shown that the 4.2 Kbps TFI coder is comparable in performance to the 8 Kbps North-American cellular standard IS54 coder and to the 13 Kbps European standard GSM coder.

I. INTRODUCTION

Low rate speech coding research has recently gained a new momentum due to the increased national and global interest in digital voice transmission for mobile and personal communication. The Telecommunication Industry Association (TIA) is actively pushing towards establishing a new "half-rate" digital mobile communication standard with twice the capacity of the current North-American "full rate" digital system (IS54). Similar activities are taking place in Europe and Japan. The demand, in general, is to advance the technology to a point of achieving or exceeding the performance of the current standard systems while cutting the transmission rate by half.

The voice coders of the current digital cellular standards are all based on the Code-Excited Linear Prediction (CELP) [1,2] or closely related algorithms. CELP coders deliver fairly high-quality coded speech at rates of about 8 Kbps and above. However, the performance deteriorates quickly as the rate goes down to around 4 Kbps and below, and it is unlikely that the new demands on quality vs. rate can be met by conventional CELP algorithms. New approaches have to be investigated either for enhancing the CELP algorithm or for creating whole new "beyond-CELP" platforms.

This paper discusses a new algorithm for high-quality coding of voiced speech and demonstrates the advantage of the proposed method over CELP for low rate coding. The algorithm is based on representing and interpolating the speech signal in the time-frequency domain. The paper proposes an empirical but perceptually advantageous framework for coding voiced speech by Time-Frequency Interpolation (TFI). In the next section the TFI method is formulated. In section 3, the proposed coder is described in detail. Section 4 presents the results of MOS[3] tests which show that the proposed 4.2 Kbps coder is comparable in performance to the 8-Kbps North-American cellular standard IS54 coder and to the 13 Kbps European standard GSM coder. Section 5 concludes the paper.

II. TIME-FREQUENCY INTERPOLATION

2.1 The TFI Framework

Time-Frequency Interpolation (TFI), as defined in this paper, is based on the concept of short-time *per-sample* discrete spectrum sequence. Each time n on a discrete-time axis is associated with an $M(n)$-point discrete spectrum. In a simple case, each spectrum is a discrete Fourier transform (DFT) of a time series $x(n)$, taken over a contiguous time segment $[n_1(n), n_2(n)]$, with $M(n)=n_2(n)-n_1(n)+1$. Note that the segments may not be equal in size and may overlap. Although not strictly necessary, we assume that n lies in its segment, namely, $n_1(n) \le n \le n_2(n)$. In this case, the $n-th$ spectrum is conventionally given by:

$$X(n,K) = \sum_{m=n_1(n)}^{n_2(n)} x(m)\, e^{-j\frac{2\pi}{M(n)}Km} \qquad (1)$$

The time series $x(n)$ may be over-specified by the sequence $X(n,K)$ since, depending on the amount of segment overlapping there may be several different ways of reconstructing $x(n)$ from $X(n,K)$. Exact reconstruction, however, is not the main objective in using TFI. Depending on application, the "over-specifying" feature may, in fact, be important and instrumental in synthesizing signals with certain desired properties.

In a more general case, the spectrum assigned to time n may be generated in various ways to achieve various desired effects. The general-case spectrum sequence is denoted by $Y(n,K)$ to distinguish between the straightforward case of Eq. (1) and more general transform operations that may utilize linear and non-linear techniques like decimation, interpolation, shifts, time (frequency) scale modification, phase manipulations and others.

We denote by $y(n,m)=F_n^{-1}\{Y(n,K)\}$ the *extended* inverse transform of $Y(n,K)$, obtained by the operator F_n^{-1}. If $Y(n,K)=X(n,K)$, then, by definition, $y(n,m)=x(m)$ for $n_1(n) \le m \le n_2(n)$. Outside this segment, $y(n,m)$ is a *periodic extension* of that segment and, in general, is not equal to $x(m)$. A new signal $z(m)$ is now derived from the signal set $y(n,m)$.

Various transforms can be used for this purpose. A simple one used here is given by:

$$z(m) = y(m,m) = F_m^{-1}\{Y(m,K)\} \qquad (2)$$

As a convention, we assume that the TFI process takes place in a time frame $[0,..,N-1]$. and that no data is available for $n \geq N$. This makes the system causal. In coding applications, only a decimated version of the sequence $Y(n,K)$ along the time axis n is available. The missing spectra are interpolated from the survivor ones. For convenience, we align the decimated sequence with frame boundaries. Specifically, all spectra but $Y(N-1,K)$ are set to zero. The nulled spectra are then interpolated from $Y(N-1,K)$ and $Y(-1,K)$ the latter being the spectrum of the previous frame. In general we have:

$$Y(n,K) = I_n(Y(-1,K), Y(N-1,K)) \ , \ n=0,..,N-1 \qquad (3)$$

where the I_n operator denotes an interpolation function along the n-axis. Various interpolation functions can be applied. In this paper, linear interpolation is used. The corresponding signals $y(n,m)$ are, then,

$$y(n,m) = F_n^{-1}\{I_n(Y(-1,K), Y(N-1,K))\} \ , \ n=0,..,N-1 \quad (4)$$

where the F_n^{-1} operator indicates inverse DFT, taken at time n, from frequency axis K to the time axis m. The final signal is therefore given by

$$z(m) = F_m^{-1}\{I_m(Y(-1,K), Y(N-1,K))\} \ , \ m=0,..,N-1 \quad (5)$$

Note that, in general, the operators F_n^{-1}, I_n *do not* commute, namely, interchanging their order alters the result. However, in some special cases they may partially or totally commute. For each special case, it is important to identify whether or not commutativity holds since the complexity of the entire procedure may be significantly reduced by changing the order of operations.

2.2 Linear TFI

As mentioned above, *Linear TFI* is used in this work. In this case, I_n is a linear operation on its two arguments and has the form $I_n(u,v) = \alpha(n)\,u + \beta(n)\,v$. the operators F_n^{-1} and I_n, now commute, and may be interchanged. Note that, although I_n is a linear operator, the interpolation functions $\alpha(n)$ and $\beta(n)$ are not necessarily linear in n and linear TFI is not a linear interpolation in that sense. The linearity implies that

$$z(m) = \alpha(m)\,y(-1,m) + \beta(m)\,y(N-1,m) \qquad (6)$$

which shows that linear TFI can be performed directly on two waveforms corresponding to the two spectra at the frame boundaries.

Linear TFI with *linear* interpolation functions $\alpha(m)$, $\beta(m)$ is simple and attractive from implementation point of view and has previously been used in similar forms [5,6]. In this case, the interpolation functions are typically defined as $\beta(m) = m/N$ and $\alpha(m) = 1-\beta(m)$, which means that $z(m)$ is simply a gradual change-over from one waveform to the other.

2.3 High vs. Low Rate TFI

The rate of the TFI is defined as the frequency of sampling the spectrum sequence, which is clearly $1/N$. The discrete spectrum $Y(n,K)$ corresponds to one $M(n)$-size period of $y(n,m)$. If $N>M(n)$, the periodically-extended parts of $y(n,m)$ take part in the TFI process. This case is referred to as Low-Rate TFI (LR-TFI). LR-TFI is mostly useful for generating near-periodic signals, particularly in low-rate speech coding.

When $N < M(n)$, the extended part of $y(n,m)$ does not take part in the TFI process. This High-Rate TFI (HR-TFI) can be used, in principle, to process any signal. However, it is most efficient for near-periodic signals because of the smooth evolution of the spectrum. Usually, in HR-TFI, the spectra are taken over overlapping time segments. Note that there are no fundamental restrictions on the TFI rate other than $1/N > 0$.

In speech coding, the TFI rate is a very important factor. There are conflicting requirements on the bit rate and the TFI rate. HR-TFI provide smooth and accurate description of the signal, but a high bit rate is needed to code the data. LR-TFI is less accurate more prone to interpolation artifacts but a lower bit rate is required for coding the data. It seems that a good tradeoff can only be found experimentally by measuring the coder performance for different TFI rates.

2.4 Time Scale Modification in TFI

An important aspect of the TFI process is the implicit Time Scale Modification (TSM) that is used during the inverse-DFT operation. TSM amounts to dilation or contraction of a continuous-time signal $x(t)$ along the time axis. The operation may be time-variable as in $z(t) = x(c(t)\,t)$. On a discrete-time axis, the similar operation $z(m) = x(c(m)\,m)$ is, in general, undefined. To get $z(m)$, one has to first transform $x(m)$ back to its continuous-time version, time-scale, and finally resample it. This procedure may be very costly. Using DFT (or other sinusoidal representations), TSM can be easily *approximated* as

$$z(m) \approx \sum_{K=0}^{M-1} X(K)\, e^{j\frac{2\pi K}{M} c(m) m} \qquad (7)$$

It is emphasized that Eq. (7) is *not* a true TSM but only an approximation thereof. It, however, works fairly well for periodic signals, with a modest amount of dilation or contraction. This pseudo-TSM method is very useful in voiced speech processing since it allows for very fine alignment with the changing pitch period. Indeed, we make this method an integral part of the TFI algorithm by defining F_n^{-1} in Eq. (4) to be

$$F_n^{-1}\{Y(n,K)\} = \sum_{K=0}^{M(n)-1} Y(n,K)\, e^{j\frac{2\pi K}{M(n)} c(m) m} = y(n,m) \qquad (8)$$

The function $c(m)$ is usually indirectly defined by choosing a particular interpolation strategy in the *fundamental phase* domain $\Psi(n,m) = 2\pi\, c(m)\, m / M(n)$. The phase interpolation is performed along the m-axis and, as implied by the above notation, it may be different for each of the waveforms $y(n,m)$. Various interpolation strategies may be employed [5,6]. The one used in the proposed coder will be described later.

In most cases, it is possible and useful to make the operator F_n completely independent of n. In this case, the phase is arbitrarily disassociated from the DFT size and is said to depend on m only. It is then determined by the chosen interpolation strategy, along with two boundary conditions at $m=0$ and $m=N-1$. For speech processing, the boundary conditions are usually given in terms of two fundamental frequencies (pitch values). The DFT size is made independent of n by simply using one common size $M = \max_n M(n)$ and appending zeros to all spectra shorter than M. Since the phase is now independent of the DFT size, namely, of the original frequency spacing, one has to make sure that the actual spacing made by the phase $\Psi(m)$ does not cause spectral aliasing. This is very much dependent upon how $Y(n,K)$ is interpolated from the boundary spectra and on how the actual size of $Y(n,k)$ is determined. One advantage of the TFI system, as formulated here, is that spectral aliasing, due to excessive time-scaling, can be controlled during spectral interpolation. This is hard to do directly in the time domain.

The time-invariant operator F^{-1} is now given by:

$$F^{-1}\{Y(n,K)\} = \sum_{K=0}^{M-1} Y(n,K)\, e^{j\Psi(m)K} = y(n,m) \qquad (9)$$

For voiced speech coding, TFI provides a useful domain in which coding distortions can be made less objectionable. This is due to the fact that the spectrum of voiced speech, especially when synchronized to the speech periodicity, changes slowly and smoothly. The TFI approach seems to be a natural way of utilizing these speech characteristics. The next section describes the proposed TFI-based speech coder.

III. LOW-RATE SPEECH CODING BASED ON TFI

3.1 Main Block Diagram of the Coder

A high-level block diagram of the proposed TFI-based coder is shown in figure 1. The coder uses the classical Linear-Predictive-Coding (LPC) decomposition of spectral envelope information, represented by an all-pole filter (LPC) parameters, and an LPC excitation signal. LPC analysis is first performed by the receiver and the parameters are quantized. The quantized LPC's are then used in processing the excitation. The coder processes and codes the excitation in one of two distinctly different modes, determined by the voicing and pitch detection unit. In the *voiced* mode, the excitation is processed by the TFI unit. In the *unvoiced* mode, the algorithm employs the Code-Excited Linear-Predictive (CELP) technique [1,2] which, essentially, means optimizing the coded excitation by monitoring the output coded speech. This is represented in the figure by the dotted feedback line. The coded excitation from either the TFI or the CELP units drives the coded all-pole LPC filter whose output is the coded speech.

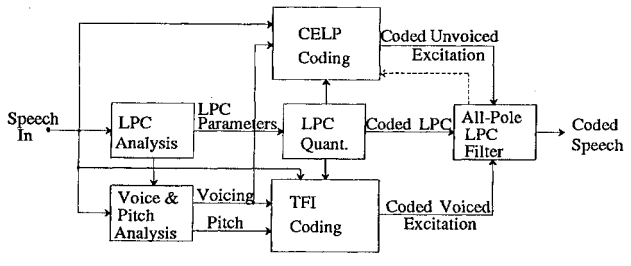

Figure 1. Block Diagram of the TFI Coder

The coder processes the input speech in 20 msec. blocks (160 samples) using 84 bits per block, for a total rate of 4.2 Kbps. A more detailed description of the system is given below.

3.2 LPC Analysis and Quantization

Standard LPC analysis is used for deriving a 10th-order all-pole stable filter. The filter parameters are then transformed to line-spectral frequencies (LSF). The LSF vectors are vector-quantized once per 40 msec. using 30-bit split vector quantization. The LSF vector is split into 3 subvectors of sizes 3,3 and 4. Each subvector is then coded by a 10-bit vector quantizer. The rate of the LPC quantizer is therefore, 15 bits per frame.

The quantized LSF's are block-interpolated from the previous to the current vectors with a resolution of 8 vectors per frame. The interpolated vectors are then converted back to LPC filter parameters. Using 8 filters per frame, the LPC filter is therefore switched every 20 samples, which provides for a very smooth update of the spectral envelope.

3.3 The TFI unit

The TFI unit is active in the voiced mode and it comprises the sequence of operations shown in figure 2. An LPC residual signal is first obtained by inverse-filtering the input speech, using the quantized interpolated LPC parameters. Once per subframe of 40 samples (5 msec.), an initial spectrum $X(K)$ is derived pitch-synchronously by applying DFT to a pitch-size segment of the residual.

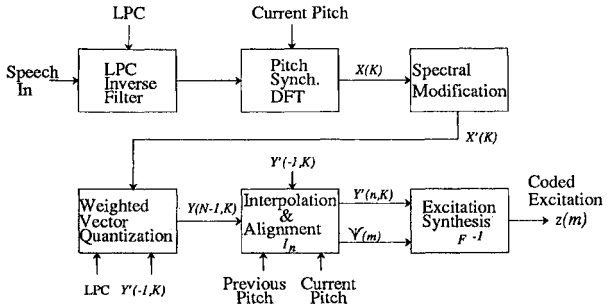

Figure 2. Time Frequency Interpolation and Coding of the Excitation

The spectrum of the current subframe is modified and quantized by a weighted, variable-size, predictive vector quantizer. The quantizer is predictive in the sense of using the previous spectrum as an initial estimate and optimizing an additive innovation spectrum $V(K)$ to get

$$Y(N-1,K) = g_0\, Y(-1,K) + g_1 V(K) \qquad (10)$$

where the vector-prediction coefficient g_0 is 0.7 if the previous subframe belongs to a voiced main frame and is zero otherwise. The VQ operation, which involves spectral weighting, is given by:

$$\min_{V(k),\, g_1} \| H(K)\,(X'(K) - Y(N-1,K)) \| \qquad (11)$$

where $\| \cdot \|$ means sum of squared magnitudes. $H(K)$ is the DFT of the impulse response of a modified all-pole LPC filter

[1,2]. The best innovation spectrum is drawn from an 11-bit codebook and is scaled by the gain g_1 drawn from a 4-bit table. This gain-shape VQ uses 15 bits per subframe, for a total of 60 bits per main frame for coding the excitation.

This system uses linear TFI as defined in section 2.2 with the linear interpolation functions $\alpha(m) = 1 - m/N$, $\beta(m) = m/N$. The inverse DFT phase $\Psi(m)$ (Eq. (9)) is interpolated assuming linear trajectory of the *pitch frequency*. If the previous and current pitch angular frequencies are ω_p and ω_c, respectively, then, the phase is given simply by

$$\Psi(m) = [\ \omega_p\ \alpha(m) + \omega_c\ \beta(m)\ ]\, m + \Psi(-1) \quad ; \quad m=0,..,N-1 \quad (14)$$

Notice that the system is a high-rate TFI since the pitch period may be greater than the subframe size. This allows for a close tracking of the inter-pitch variations which is important for preserving the naturalness of the voiced speech. Reducing the TFI rate still preserves the smoothness of the TFI process but may cause objectionable artifacts due to excessive periodicity.

In the TFI mode of operation, the pitch period is coded and transmitted using 7 bits. Adding one bit for voicing, 60 bits for the excitation and 15 bits for the LPC, sums up to 83 bits per 20 msec. The total rate in this mode is, therefore, 4.15 Kbps.

3.4 The CELP Unit

The CELP unit is active in the unvoiced mode. A common CELP structure is used in this this coder, except for the fact that a pitch loop or similar techniques for exploiting long-term redundancies are *not* used here. The process is applied to subframes of 40 speech sample, that is, 4 times per main frame. A standard closed-loop minimization procedure is used [1,2] for selecting an excitation vector from a 12-bit codebook and a scale factor from a 5-bit table. The total number of bits per frame, assigned to the excitation is 4(5+12)=68 bits. Adding one voicing bit and 15 LPC bits gives a total of 84 bits, corresponding to a rate of 4.2 Kbps.

IV. PERFORMANCE

The performance of the TFI-based coder was formally assessed by a Mean-Opinion-Score (MOS) test [3]. The MOS test rates the speech quality on a scale of 1 to 5, based on an average response of many listeners. The coder was tested along with the European GSM [7] and the North-American IS54 [4] full-rate coders to calibrate its performance against the current standards. Uncoded speech, referred to as PCM, was also included in the test for reference. The test material included filtered (IRS) and unfiltered (non-IRS) speech. The filtered speech was processed by an IRS filter [8] to simulate a typical commercial telephone line. The scores are given in table 1.

Coder	Rate (Kbps)	IRS	Non-IRS
PCM	--	4.21	4.17
GSM	13.00	3.61	4.04
IS54	7.95	3.86	4.16
TFI	4.20	3.96	3.99

Table 1. Mean Opinion Scores [3] for the standard full-rate coders and the TFI coder.

The MOS scores of the 4.2 Kbps TFI coder are very close to 4.0 which is considered very high at this bit rate. This level of performance is unlikely to be achieved by conventional CELP coders. TFI coder performs somewhat better than the full-rate coders for the IRS condition and somewhat worse for the non-IRS condition. These results indicate that TFI is a viable framework for a new generation of speech coders capable of achieving higher performance at half the bit rate of the current systems.

V. CONCLUSIONS

The paper presents a time-frequency interpolation framework for coding voiced speech. A general formulation of the TFI method is given and then, a 4.2 Kbps coder based on TFI is proposed. It is shown via formal MOS scores that this coder performs very close to the current digital cellular standard coders while operating at about half the bit rate. TFI coding is therefore a promising technique for the next-generation half-rate digital-cellular systems.

REFERENCES

[1] M.R. Schroeder, B.S. Atal, "Code-Excited Linear Predictive (CELP): High Quality Speech at Very Low Bit Rates", Proc. IEEE Int. Conf. ASSP., 1985, pp. 937-940.

[2] P. Kroon, E.F. Deprettere "A Class of Analysis-by-Synthesis Predictive Coders for High-Quality Speech Coding at Rate Between 4.8 and 16 Kb/s.", IEEE J. on Sel. Area in Comm. SAC-6(2), Feb. 1988, pp. 353-363.

[3] "Subjective performance assessment of digital encoders Using degradation category rating procedure". CCITT Blue Book, Vol. V, Supp. 14

[4] TR45 "Full Rate Speech Codec Compatibility Standard", PN-2972 EIA, 1990.

[5] W.B. Kleijn, "Continuous Representations in Linear Predictive Coding", Proc. ICASSP-91, May 1991, Vol. S1, pp. 201, 204.

[6] W.B. Kleijn, "Methods for Waveform Interpolation in Speech Coding", Digital Signal Processing, Vol 1, No. 4, 1991, pp. 215-230. Academic Press.

[7] GSM 6.10 "GSM Full Rate Speech Transcoding", ETSI, Jan. 1990

[8] "Specification for an Intermediate Reference System", CCITT Blue Book Vol. V, Rec. P.48, p. 81