Adaptive Transform Coding (ATC) of Speech

Phase II

Peter Kabal   Rafi Rabipour

INRS-Télécommunications
3, Place du Commerce
Île des Soeurs, Québec H3E 1H6
(514) 768-6691

INRS-Télécommunications Report No. 83–07

April 1983

Final Report to Communications Research Center

Contract OST82–00085

1. Introduction

1.1 Background

This is a report on work done under contract to the Communications Research Center, Department of Communications. The previous phase of the work on Adaptive Transform Coding (ATC) of speech concerned itself with the ATC algorithm and developing a computer simulation for assessing the quality of this speech coding technique [1][2]. In this phase of the work, implementational aspects of the ATC algorithm are emphasized.

ATC is a relatively complicated scheme which produces good quality speech at low data rates (9.6 kb/s). These low data rates are important for the efficient utilization of channel capacity, especially on circuits such as satellite channels. Phase I of the investigation compared different algorithmic options for ATC coders. One important result was the demonstration of an ATC system using short block lengths. This scheme has a significantly reduced complexity without unduly sacrificing speech quality. It also pointed to the possibility that an ATC coder could be implemented in the near future in real-time hardware.

Phase II, the current work, covers modifications to the algorithm with a view to examining the options opened up by the availability of special purpose digital signal processing integrated circuits. The objectives of this study are as follows.

a) Investigate modifications to the algorithm to reduce the complexity. The objective is to pinpoint computational hurdles in the algorithm which can be then be alleviated to reduce the processing burden.

b) Produce a version of the algorithm using fixed point arithmetic. The aim of this aspect of the work is to determine the feasibility of a hardware implementation which would of necessity use fixed point hardware. This aspect of the investigation represents the major part of the work reported.

c) Quantify the performance of the final algorithm. An important aspect of this part of the work is to assess the sensitivity of the speech quality with respect to acoustical background noise.

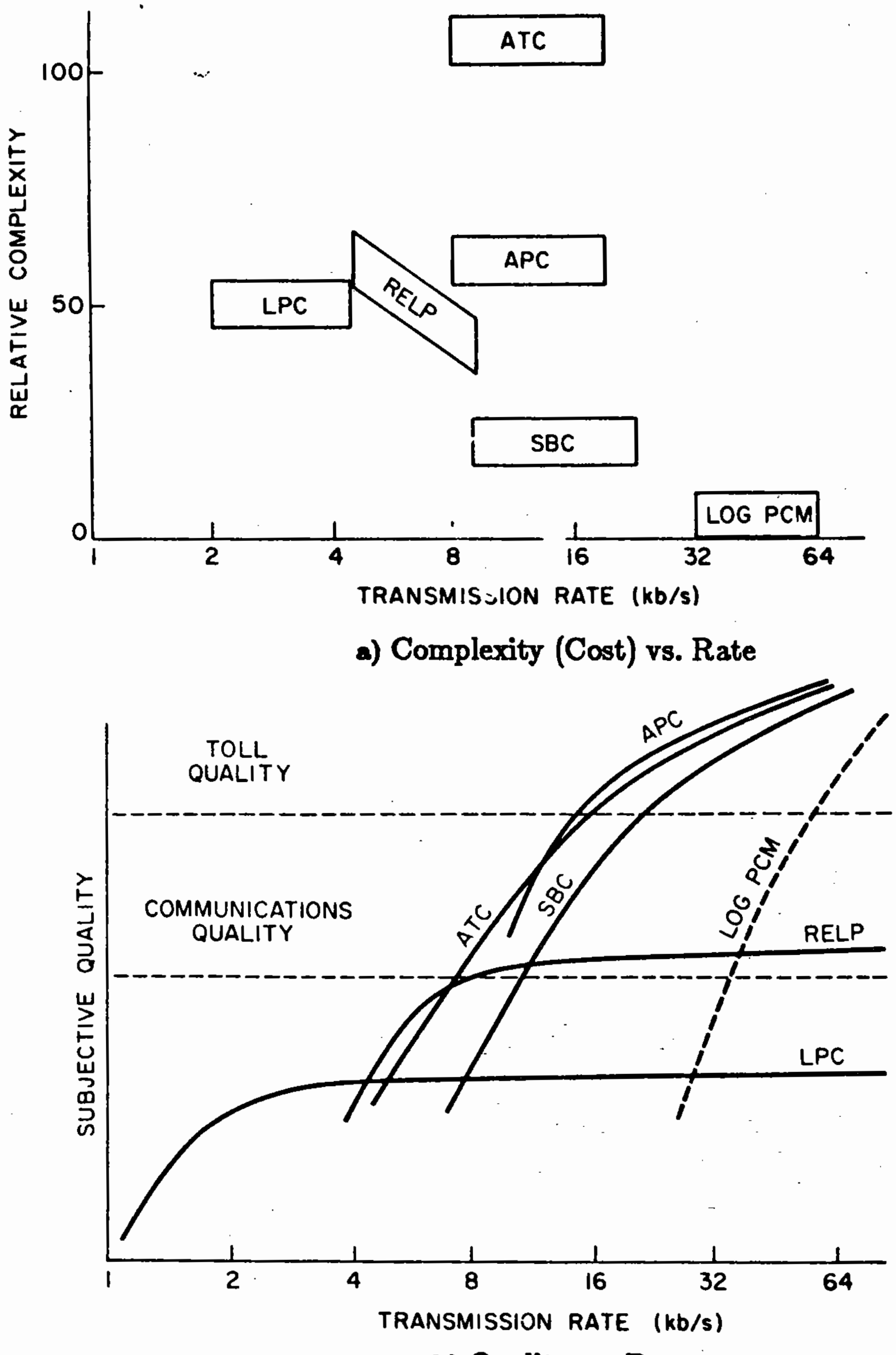
In the remaining part of this section, we briefly discuss speech coding algorithms in an attempt to give the reader an understanding of the alternatives available.

1.2 Digital Speech Coding

Digital transmission facilities are the option of choice for new systems since they are more reliable, flexible and economical than analog systems. Digital speech coders serve as the interface between end-user and digital transmission links. The choice of a coding algorithm is dependent on many factors. In particular, the tradeoffs available can be visualized as spanning 3 dimensions: complexity (or cost), transmission rate and speech quality (see Fig. 1-1) [3]. At data rates above 10 kb/s, bit rate reductions can be achieved by increasing the complexity of the coding schemes while maintaining relatively good speech quality. At rates below 10 kb/s, some speech quality degradation is inevitable. Often the use of low bit rate coders is dictated by the need for a bit rate compatible with existing analog transmission facilities or the need to efficiently use scarce bandwidth resources. To achieve reasonably high quality in the reproduced speech at these low bit rates, a relatively complex algorithm must be employed.

Speech coding schemes can be classified into two main types: waveform coders and analysis/synthesis coders.

APC	- Adaptive Predictive Coding
ATC	- Adaptive Transform Coding
LPC	- Linear Predictive Coding
RELP	- Residual-Excited Linear Predictive Coding
PCM	- Pulse-Code Modulation
SBC	- Sub-Band Coding



a) Complexity (Cost) vs. Rate

b) Quality vs. Rate

Figure 1-1 Complexity (Cost) / Rate / Quality Trade-offs for Speech Coders

1.3 Waveform Coders

Waveform coders attempt to replicate the input waveform at the receiving end. The complexity of waveform coding runs the gamut from pulse code modulation (PCM) up to sophisticated techniques such as ATC or Adaptive Predictive Coding (APC). High quality speech can be achieved with transmission rates as low as 10 kb/s using the more complex types of waveform coders; whereas the simpler schemes require higher rates.

As the bit rate is lowered, each waveform sample is of necessity more coarsely represented. In order to improve the quality of reproduction, the inherent redundancy and local properties of the input waveform must be utilized. The correlation between adjacent samples is used to advantage by schemes which digitize the difference between a value predicted on the basis of the past samples and the input signal. The difference signal has a smaller dynamic range than the original signal and hence is easier to quantize. The simplest forms of differential PCM have a compromise fixed linear predictor.

The input waveform has different statistical properties for different speakers and indeed for the same speaker within an utterance. Modifications to the basic scheme which adapt to the local (in time) properties of the speech waveform are thus beneficial. The digitizer can be made adaptive in that it will expand and contract its range to follow the short term energy trends in a signal. In addition, the predictor can be made adaptive so that it learns the local correlation properties of the input signal. These additions lead to robust coders that follow the changing characteristics of the input signal. The voiced segments of speech display a quasi-periodicity which can also be exploited by using an adaptive pitch predictor. A further improvement results if advantage is taken of imbedded silence in speech. In variable rate coding, the high amplitude segments of a speech waveform are more precisely quantized (by being assigned more bits) than the low amplitude portions.

The quantization noise spectrum of the reconstructed signal can be shaped in such a way as to utilize the perceptual masking to reduce the subjective effect of the quantization distortion. Adaptive Predictive Coding is a time domain technique which uses all of these techniques in order to efficiently code speech at low rates.

Many of the same ends can be achieved by coding in the frequency domain. In its simplest form, a frequency domain coding technique such as sub-band coding, divides the input spectrum into a small number of disjoint bands and codes each band separately, multiplexing the bit streams together. The digitizer for each subband adapts to the short term statistics of that band. Additional benefits accrue to frequency domain coders since quantization noise produced tends to be less bothersome than time domain coding noise.

Adaptive Transform Coding (ATC) carries this process further by using a large number of frequency bands and by dynamically allocating bits to the bands. The available bits are allocated in such a manner as to reduce the overall distortion, generally favouring the most energetic portions of the signal.

1.4 Analysis/Synthesis Coders

Analysis/synthesis coders attempt to model the production of speech. The transmitter analyzes the input speech to produce a set of parameters each of which is digitized and sent to the receiver. At the other end the speech signal is synthesized using the received parameters. By employing a model appropriate to speech production, highly intelligible speech can be reproduced from a very compact parameterization of the input speech. Unfortunately, problems occur when the input speech deviates from the assumed model. Generally, these coding schemes are not robust to atypical speakers (e.g., high pitched individuals) or to speech corrupted by background acoustical noise (e.g., office noise or background noise in vehicles). Amongst analysis/synthesis coders, the best known technique is that of linear predictive coding (LPC) [4]. LPC uses an all-pole model of speech production to produce highly intelligible speech at low transmission rates (2.4 kb/s and below).

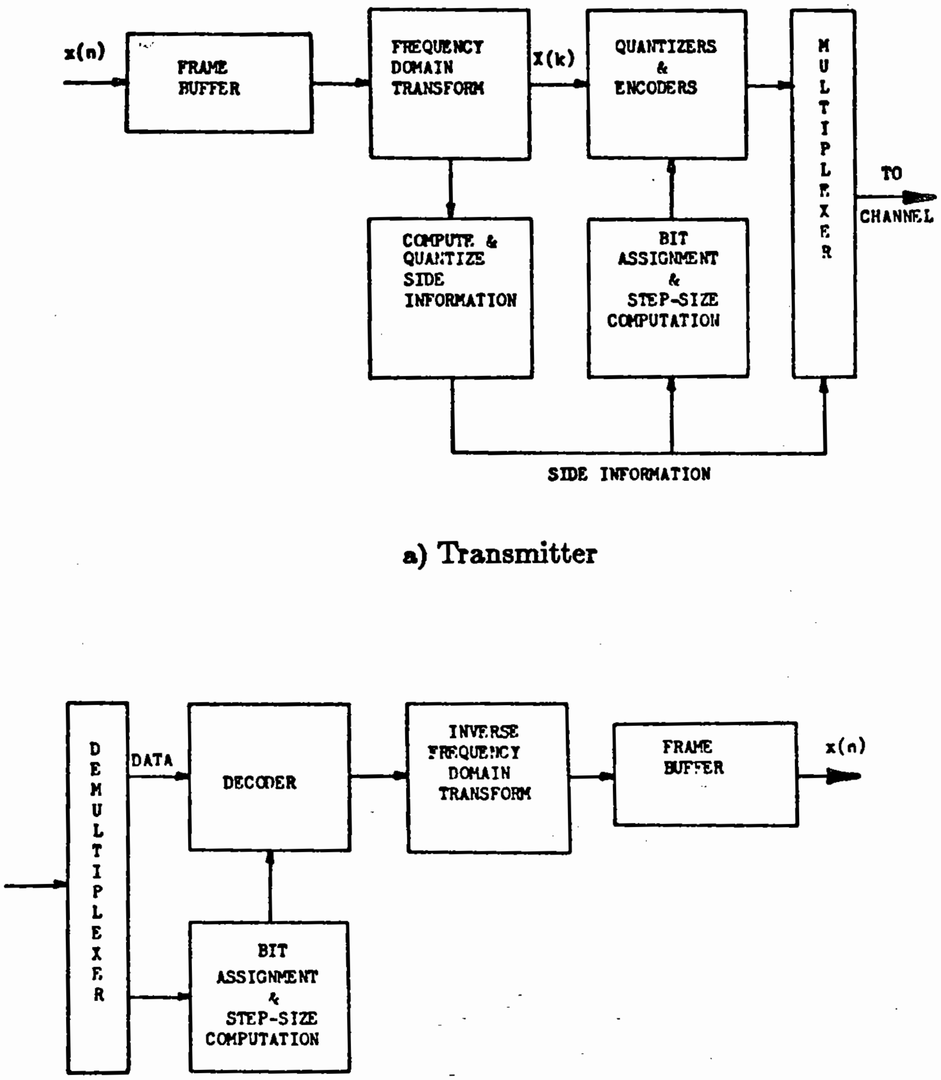
2. Adaptive Transform Coding

In transform coding, the input speech samples are processed in blocks (frames). Each block of samples undergoes a linear transformation. The transformed coefficients are then quantized sample by sample and coded for transmission. At the receiver, the received data is decoded and passed through an inverse transformation. The resulting samples represent the coded speech signal. As described, this constitutes a transform coder. The optimal linear transformation for the above configuration is the Karhunen-Loève transform (KLT) which decorrelates the data. However, the KLT requires an exact knowledge of the correlation for the block of samples, making it data dependent.

For speech coding, it is desired to use a fixed transformation that approximately decorrelates the data. In addition, the transformation should be simple to apply. The Discrete Cosine Transform (DCT) is the prime contender. This transformation decorrelates a first order Markov sequence or equivalently a sequence with an exponentially decaying correlation function [5]. In fact, the correlation function for speech roughly fits this function. An important advantage of the DCT is that fast computational algorithms based on the fast Fourier transform (FFT) algorithm can be used to calculate the DCT.

In an ATC coder, it is the DCT coefficients that are transmitted, not the actual speech samples themselves. Simple PCM coding of the transformed coefficients results in an increase in signal-to-noise ratio of about 6 dB over that of straight PCM coding of the speech samples [6]. However, the major quality improvement for ATC occurs when adaptive bit assignment is used.

The basic steps in an ATC coder as developed in Phase I of this project are as follows (Fig. 2-1).


x(n)
FRAME BUFFER
FREQUENCY DOMAIN TRANSFORM
X(k)
QUANTIZERS & ENCODERS
MULTIPLEXER
TO CHANNEL
COMPUTE & QUANTIZE SIDE INFORMATION
BIT ASSIGNMENT & STEP-SIZE COMPUTATION
SIDE INFORMATION
DEMULTIPLEXER
DATA
DECODER
INVERSE FREQUENCY DOMAIN TRANSFORM
FRAME BUFFER
x(n)
BIT ASSIGNMENT & STEP-SIZE COMPUTATION

a) Transmitter

b) Receiver

Figure 2-1 Block Diagram of an ATC Coder

a) The input sampled speech is pre-emphasized to accentuate the higher frequencies.

b) The speech samples are grouped into overlapping frames of from 32 to 256 samples for speech sampled at 8 kHz.

c) The discrete Cosine transform (DCT) is taken of the frame of data. The envelope of the DCT coefficients has the same amplitude as the conventional discrete Fourier transform (DFT) [7]. This means that the DCT coefficients have a frequency domain interpretation that is useful in interpreting the operation of the coder.

d) The transform coefficients are quantized with a variable number of bits. The bit assignment is determined so as to affect the best quality of the reproduced speech.

e) The quantized coefficients along with side information necessary for the decoder to determine the bit assignment are sent to the decoder.

f) The decoder reconstructs the transform coefficients from the received information.

g) The inverse DCT is taken of the transform coefficients.

h) The resultant data is deemphasized (an inverse operation to pre-emphasis) and windowed to give the output speech.

2.1 Spectral Modelling

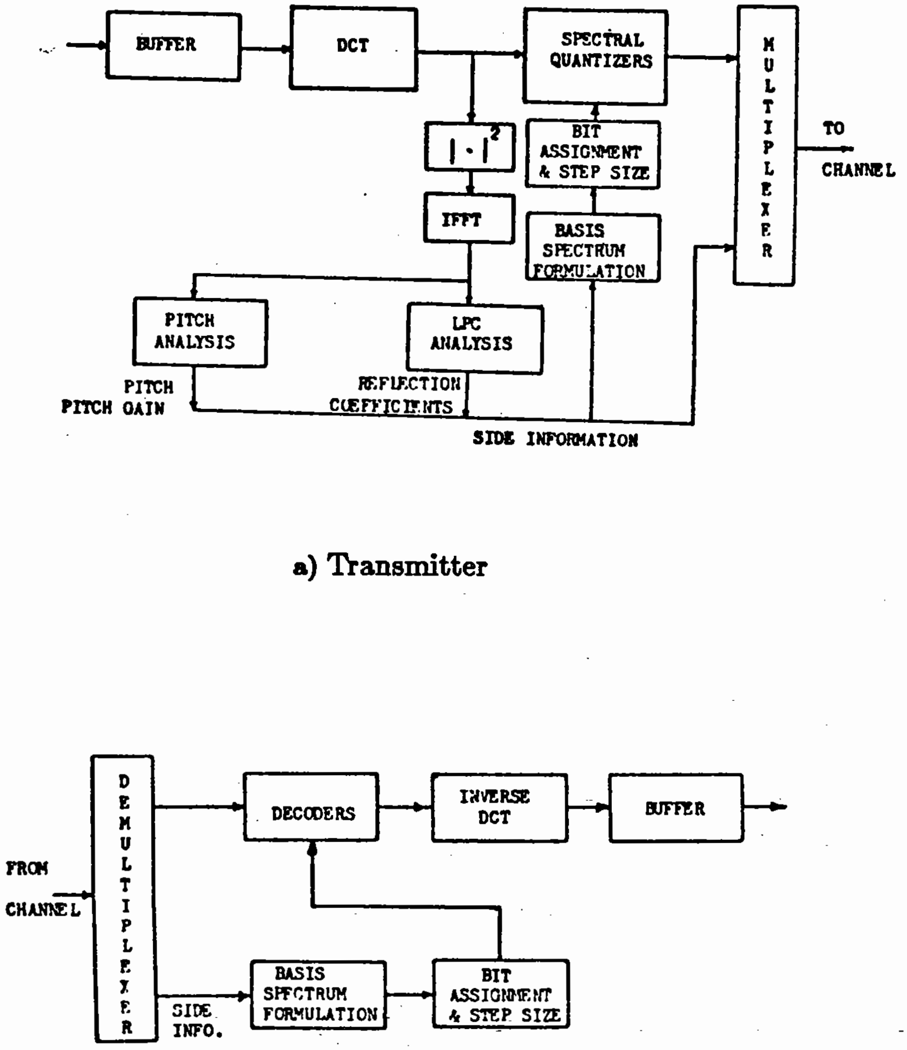
The major influence on the quality of the reconstructed speech is the bit assignment strategy. Based on the frequency interpretation of the DCT coefficients, a spectral fit to the formant structure of the coefficients provides an estimate of the energy of the coefficients. This energy estimate serves a dual role. First, it is used to determine the number of bits to be assigned to each coefficient in such a way as to minimize the total mean-square quantization error. This means that the higher energy coefficients will be assigned more bits than the lower energy coefficients. The spectral estimate is also used to scale the quantizer, in effect adapting the quantizer range to the coefficient being quantized.

The DCT spectrum retains the formant structure of the speech. A similar problem of estimating the spectrum of speech occurs in linear predictive coding of speech. An all-pole spectral model is appropriate for modelling speech production since it tends to highlight the spectral features known to be perceptually important. The frequency interpretation of the DCT values indicates that such a method of determining an average energy contour is also appropriate in this case.

Fig. 2-2 shows the steps involved. The optimal mean-square error fit is determined most conveniently from auto-correlation coefficients. In this case, a pseudo auto-correlation function is determined from the DCT energy spectrum by inverse transforming the energy spectrum using a DFT [7][8]. The Levinson algorithm can then be used to efficiently calculate the all-pole fit. This model can be described by a number of different parameters, for instance pole positions or filter parameters. The reflection coefficient form has been used since it is a particularly convenient form to code for efficient transmission of the side information.

2.2 Bit Allocation Strategy

The bits alloted for the transmission of a frame of transform coefficients are assigned so as to minimize the mean-square error in the reconstructed speech. A second round of optimization is used later to modify the basic strategy to better reflect psycho-acoustical phenomena. The bit allocation process allocates one bit at



a) Transmitter

b) Receiver

Figure 2-2 Basis Spectrum Generation

a time from the pool of available bits for the frame of data. At each step of the process, a bit is allocated to the coefficient which will give the largest decrease in mean-square error with the addition of that bit [2].

2.3 Quantization

The coefficients are quantized using a quantizer with the number of bits specified by the bit allocation. The spectral estimate is also used to adapt the quantizer range, or equivalently to normalize the transform coefficients.

2.4 Decoding

The decoder regenerates a spectral estimate of the DCT energy spectrum using the side information transmitted to it. The bit allocation, using the identical algorithm to that used in the transmitter, is recalculated at the decoder. With this knowledge, the decoder can parse the incoming bit stream knowing which bits belong to which coefficient. The restored quantized DCT coefficients can then be inverse transformed to form the output speech signal.

3. Algorithm Experimention

The basic algorithm as outlined in Section 2 produces very good quality speech for rates down to about 16 kb/s. Below this rate, the paucity of bits becomes evident. At 9.6 kb/s, only an average of 1 bit per sample is available for coding the coefficients. Some coefficients will not be coded at all, while others will be quantized coarsely. However, within the framework of the basic algorithm described in the previous sections, modifications to take into account perceptual effects can be employed. Modifications to the ATC algorithm to reduce the computational complexity are also discussed. Finally the performance of the algorithm with respect to background acoustical interference is reported.

3.1 Basis Spectrum Modification

The basic route taken to alter the bit assignment has been to modify the spectral estimate (here referred to as the basis spectrum). This leads to convenient interpretations of the steps involved.

3.1.1 Pitch Modelling

The spectral distribution of the bits can also be improved for speech signals by taking into account the quasi-periodicity of the speech signal during voiced segments. This manifests itself in the frequency domain as a comb spectrum. The smooth energy contour generated by a linear predictive model models the speech formant structure but does not adequately describe the fine pitch detail. An improvement in performance can be expected with pitch modelling because for frequencies between the pitch harmonics, no bits need be assigned. The equivalent time domain interpretation shows that the the formant model takes into account only the low order auto-correlation coefficients. The role of the pitch model is to given some weight to successive coefficients. Two methods for the modelling of the pitch structure are considered.

3.1.2 Pulse Train Pitch Model

In this method, the pitch period is estimated from the pseudo auto-correlation function as the location of the second peak in the function. The degree of periodicity is ascertained from the ratio of the second peak height to the first peak height (signal energy).

The steps involved in this process can be outlined as:

a) Given the N DCT coefficients, the energy spectrum is computed by squaring these terms. The correlation function is calculated as the inverse discrete Fourier transform (DFT) of the energy spectrum. The first 8 to 12 coefficients are used to determine the formant model of the energy spectrum using a minimum mean-square error fit. The basis spectrum fit to the energy spectrum is then calculated from the model parameters.

b) The location of the second peak of the correlation function beyond the lag corresponding to the coefficients used to determine the formant spectrum is deemed to be the pitch period. The ratio of the main peak (at lag zero) to the pitch peak is used to define the amplitude of a pulse train. This pulse train has pulses with exponentially decaying amplitudes at multiples of the pitch period. The frequency response of this pulse train is used to multiply the basis spectrum derived from the low order correlation terms by the LPC procedure. The pitch model assumes that the modified correlation function is the convolution of the low order correlation terms with the pitch pulse train. The modified correlation function is a superposition of delayed and attenuated replicas of the correlation function.

The implementation used allowed the pitch model to be gradually added to the basic formant spectrum. A pitch gain of 0.4 (0 corresponding to no pitch model and 1 corresponding to a full pitch model) gives good results. This represents a compromise between no pitch model and the inadequacies of the model. The model assumes a particular form for the correlation function. If the correlation fails to match, the pitch component may actually make the spectral fit worse. The fit has been observed to be poor at high frequencies if the pitch is not an integral number of samples.

3.1.3 Minimum Mean-Square Error Pitch Model

The pitch model of the previous scheme is applied in such a way as to modify the smooth formant spectrum. An alternate scheme in which a combined formant/pitch model is derived simultaneously was also used. This method uses the low order correlation terms (as in the usual LPC fit) but also uses correlation terms centred around the pitch peak. A combined set of formant/pitch terms is formed to simultaneously solve for the optimal spectral fit. The advantage of this method is that the secondary (pitch) correlation peak always improves the spectral fit compared to not using the pitch information. In fact, during unvoiced segments the secondary peak slides down to effectively increase the order of the formant analysis.

Let the auto-correlation sequence be $R(i)$. The optimal predictor coefficients, $\{c_1,\ldots,c_N\}$, at time delays $D_1,\ldots,D_N$ are given by the solution to the matrix equation,

$$\mathbf{A}c = \alpha\,,$$

where A is the matrix of auto-correlation terms with entries $A_{ij} = R(D_i - D_j)$, c is the vector of predictor coefficients, and α is the vector of cross-correlation terms with entries $\alpha_i = R(D_i)$. In the case at hand, the formant part of the predictor has $D_i = i$, while the pitch part of the predictor has terms centred around the delay corresponding to the pitch period, i_P, e.g. for three pitch coefficients, $D_{N-2} = i_P - 1$, $D_{N-1} = i_P$, $D_N = i_P + 1$. The combined equations that must be solved are no longer Toeplitz as in the previous method. This means that the Levinson recursion is not applicable. However, a Cholesky decomposition can be used to get a solution reasonably efficiently. The total amount of computation is slightly reduced since in the previous method the pitch modelling adds considerable complexity to the basic LPC solution.

Experiments indicate that three additional correlation terms located near the pitch peak are a good compromise between performance and complexity, although even a single term is beneficial. Using more than three terms gives diminishing additional benefit in signal-to-noise ratio.

The predictor coefficients corresponding to the formant part of the predictor can be transformed to reflection coefficients for efficient coding for transmission. The pitch part of the predictor must be handled separately. In the context of pitch prediction in Adaptive Predictive Coding (APC), Atal [9] suggests the following quantization approach for three pitch predictor coefficients. If the coefficients are c_{N-2}, c_{N-1} and c_N, the following bit quantization is suggested,

$\log(c_{N-2} + c_{N-1} + c_N)$	5 bits
$c_1 - c_3$	4 bits
$c_1 + c_3$	4 bits
pitch value	7 bits

3.1.4 Comparison of the Pitch Modelling Techniques

The modifications to the basis spectrum were compared using a floating point simulation of the ATC coder. Listening tests show that when the pitch structure is modelled with a pulse-train there is a slight improvement in the quality of the coder output. Fig. 3-1 shows the plots of the energy and the pulse-train modified basis spectrum for the vowel /a/. It is seen that adding the pitch modelling improves the estimate. However, since the pitch period can only be estimated with finite precision, the estimated pitch structure may not match the true pitch harmonics at the high frequency end of the spectrum.

The minimum mean-square error pitch model was examined using the first 10 samples of the correlation function and 3 samples centered at the second peak of the correlation function. Subjective tests show that there is a subtle improvement in the coder performance over the pulse-train pitch modelling; there is less burbling noise and some high frequencies reappear. The plots of the energy and the modified basis spectrum for the utterance /a/ (given in Fig. 3-2) show that indeed a better spectral fit is achieved, especially in the "valleys" of the spectrum.

3.2 Basis Spectrum Calculation

A new algorithm for taking FFT's of real, even sequences has been developed. This scheme uses a $N/4$ point complex FFT to be used to evaluate the DFT of a N point real, even sequence [10]. The novelty of this algorithm is that it uses no divisions. In the ATC coder, this new algorithm has application in finding the correlation sequence from the energy spectrum and in finding the energy spectrum from the spectral model parameters.

The use of the FFT in calculating the pseudo auto-correlation function was eventually abandoned in the integer version of the ATC coder in favour of a technique that is numerically more stable (see Section 4.3).

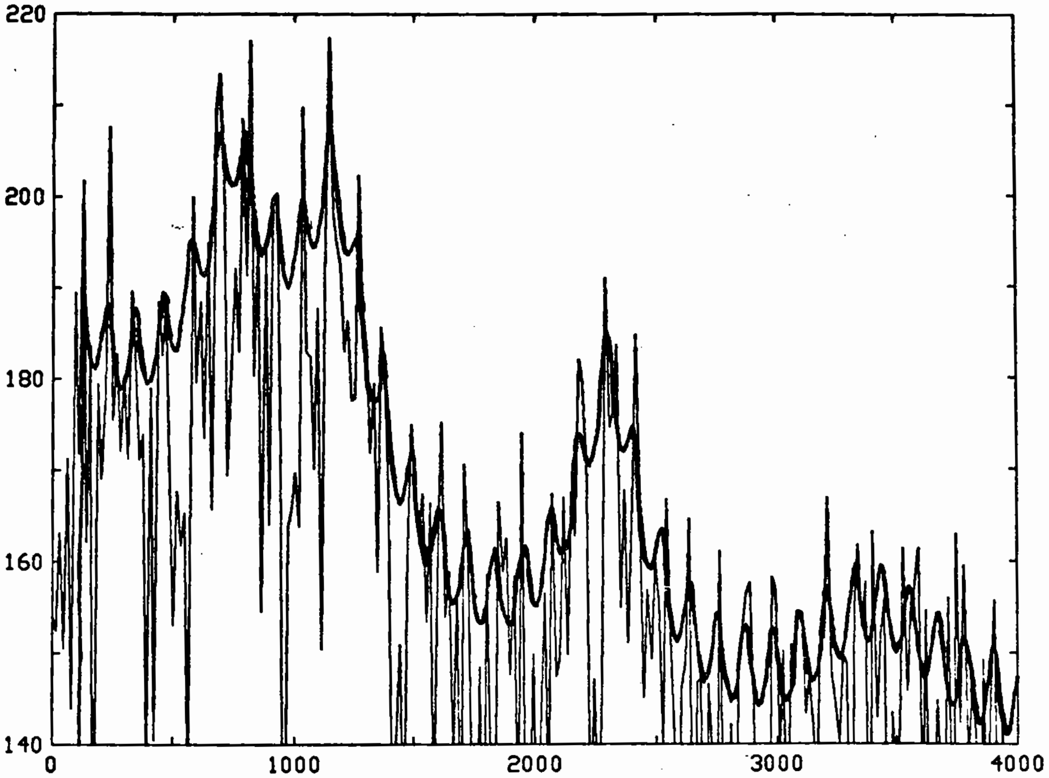


Figure 3-1 Modified Basis Spectrum (First Pitch Model)

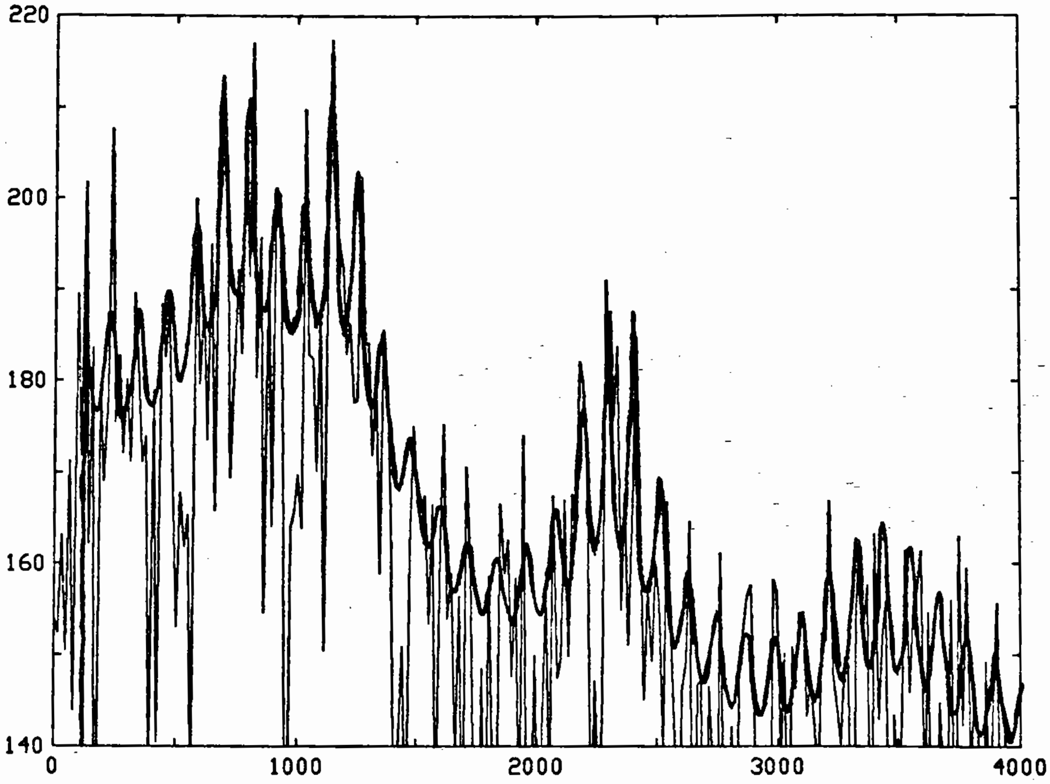


Figure 3-2 Modified Basis Spectrum (Second Pitch Model)

3.3 Bit Assignment

The procedure for bit assignment follows the optimal allocation procedure proposed by Segall [11]. Since the DCT is a unitary transformation, minimizing the mean-square error of the coded transform coefficients is equivalent to minimizing the mean-square error of the time waveform. The total mean-square error is the sum of the mean-square error for each DCT coefficient. The bit allocation proceeds by assigning bits in such a way as to maximally decrease the mean-square error with each additional bit that is assigned.

The basis spectrum is used as an estimate of the variance of the DCT coefficients. It is used to estimate the error incurred in coding the DCT coefficients and in a later stage, during the actual quantization, to normalize the DCT coefficients before quantization.

The marginal decrease in the mean-square error incurred when an additional bit is assigned to coefficient k to take the allocation from i bits to $i+1$ bits is

$$\delta_{ik} = \sigma_k^2 \, (d_i - d_{i+1}) \, ,$$

where d_i is the mean-square error for a i bit quantizer with a unit variance input and σ_k^2 is the estimate of the variance of coefficient k (in our case the basis spectrum value). The normalized marginal distortion $d_i - d_{i+1}$ is assumed to be a monotonically decreasing function of i. The values of d_i are determined from the statistics of the coefficients. In the case of the ATC coder, the normalized DCT coefficients are quantized using a Gaussian quantizer. The performance of this quantizer is essentially the same as that of one optimally designed for the probability density function of the coefficients [2].

The array δ_{ik}, $i = 1, \ldots, N_{max}$, $k = 1, \ldots, N_{cof}$ is formed where N_{max} is the maximum number of bits to be assigned to any one coefficient (typically 5 for a 9.6 kb/s coder) and N_{cof} is the transform length. The coefficients corresponding to the largest elements in this array are assigned bits. Because of the monotonicity assumption about the marginal distortions, this is an optimal assignment. The number of elements assigned bits is determined by N_{bit}, the total number of bits allocated for coding the DCT coefficients in each frame.

Timing measurements indicate that the sorting takes a considerable fraction of the computation time for the ATC algorithm with a frame length of 256. (For shorter frame lengths, the fraction of time taken reduces proportionally to the reduction in frame length). The sorting algorithm used in the original implementation is of order $NlogN$ on the average, but of order N^2 in the worst case (see the Quicksort algorithm [12]), where N is $N_{max}N_{cof}$. In a real-time coder, the worst case must be allowed for. In addition, this method requires N storage locations.

An alternate sorting algorithm was implemented which reduces the computational load by taking into account structure in the problem at hand. The new algorithm takes advantage of the fact that the marginal distortions for a given coefficient decrease monotonically as the number of bits increases. Initially we form the N_{cof} values $\{\delta_{1k}\}$. This requires N_{cof} multiplications. The largest of these is assigned a bit. The value of δ_{1k} for the coefficient in question is then replaced by δ_{2k} for that coefficient and the selection procedure is repeated for the remaining bits to be allocated. Only one additional multiplications is needed for each bit assigned. Thus instead of $N_{cof}N_{max}$ multiplications, we need at most $N_{cof} + N_{bit}$. This reduces the number of multiplications to scale the distortion terms by a factor of over two for typical situations in a 9.6 kb/s coder with a frame length of 256. This scheme requires only N_{cof} storage locations. This procedure reduces the worst case sorting time to slightly less than the average time for Quicksort.

3.4 Basis Spectrum Weighting

A method suggested in [7] to control the distribution of bits assigned to the basis spectrum is to raise the basis spectrum to the power $\gamma + 1$. With $\gamma = 0$, the distribution of bits remains such as to minimize the mean-square coding error. With $\gamma = -1$, the distribution of bits is uniform across frequencies. A compromise value will tend to cause bits to be assigned to some of the low energy high frequency components without "stealing" too many bits from other important high energy components.

An important observation is that the bit allocation is determined entirely by the relative ordering of the scaled marginal distortion,

$$\delta_{ik} = \sigma_k^2 \left(d_i - d_{i+1}\right) .$$

This means that we can achieve the weighting effect either by raising σ_k^2 to the power $\gamma + 1$, or by raising the marginal distortion to the power $\frac{1}{\gamma+1}$. Since the marginal distortions are fixed quantities, the weighting effect may be implemented without incurring computational overhead.

3.5 Background Acoustical Interference

Experiments were conducted to determine the robustness of the ATC algorithm to background acoustical interference. The question is whether a coded signal suffers in quality when a background interfering signal is superimposed on the desired speech signal. Three types of background interference were used: multiple tones (narrow band interference), white noise (wide band interference) and a background talker (speech-like interference).

The experiment consisted of coding an utterances A and an interfering signal N. In addition the sum of A with various proportions of N (symbollically $A + kN$)

was also coded. Let the coding operation for the utterance be denoted by $C(A)$. The effective linearity of the system was determined by comparing $C(A+kN)$ with $C(A)+kC(N)$ at a coding rate of 9.6 kb/s. In addition, the quality of the coded signal and the coded interference were separately evaluated. The quality assessments were made by the authors— naïve listeners would probably not be able to discern the small differences that we note in some cases.

3.5.1 Tone Interference

In this case the interfering signal consists of two equal level tones at frequencies of 697 and 1477 Hz, two of the touch tone frequencies used in telephony. Three different levels of interference were tested— each tone having an rms level of 1%, 10% or 25% of the signal rms value. The tones coded by themselves were reproduced very well by the coder. For the low level case, the tone portion of the coded combined signal tended to be turned on and off by the speech, i.e. the tone seemed to pulsate somewhat. This is explained by the fact that the speech being of higher level, when present, gets most of the bits assigned to a frame resulting in the tone frequencies not being coded. We note that even when the coded tone is added to the coded speech, there is a perceived pulsation in the tone level due to masking effects. Overall, there is little perceived degradation to the speech quality. The difference between the coded composite signal and the sum of the coded signals is small. At the intermediate level of interference, the pulsating effect on the tone in the coded composite signal is diminished. The speech part of the coded composite signal is a little more muffled than when the tone is not present. At the highest interference level, the speech is again a little clearer without the tone present.

3.5.2 White Noise Interference

In this case the interfering signal is Gaussian white noise with an rms value of 1%, 10% or 25% of the signal rms value. The low noise level was almost completely suppressed in the coded composite signal when speech was present. The effect is that the low level noise seems to start at the end of the utterance. For the intermediate level noise, the noise component of the coded composite signal changes from a "hiss" to various "swishing" noises and artifacts. At the highest noise level, the artifacts become a little more like the original noise. The speech quality does not seem to be seriously degraded but the introduced changes in the characteristics of the noise are more disturbing than the original hiss-like noise.

3.5.3 Speech Interference

In this case the interfering signal consists of another speaker. The experiment was done using a foreground male speaker and a background female speaker and vice versa, in the expectation that speakers of different sexes represent a worst case situation since fewer spectral components overlap. Each speaker was reproduced well by the coding. For various levels of interfering speech, no degradation due to coding the composite signal could be noticed. The coded composite signal was indistinguishable from the sum of the coded signals. The fact that both the signals themselves have changing frequency components seems to render any cross effects negligible.

4. Integer Computation in the ATC algorithm

In this section the practical issues involved in the implementation of fixed point ATC coder are discussed. The recent appearance of single chip digital signal processing (DSP) elements has substantially broadened the scope of speech coding algorithms which can be implemented in real-time hardware. Even the relatively complex coding schemes such as the one under consideration are amenable to practical implementation. The first step in this direction is demonstration of an ATC coding algorithm using fixed point computations with performance similar to that exhibited by the floating point simulations of ATC coding. The present generation of DSP chips is also characterized by having limited on-chip storage capability. For this reason, the overall algorithm is divided into "chunks" that can be handled by individual processors.

4.1 Fixed point arithmetic

The arithmetic in DSP hardware is for the most part implemented as *fixed point* computations. By this it is meant that the location of the binary point is fixed. The implementation of fixed point arithmetic using finite length registers can result in two types of errors. *Round-off* or *truncation* error is caused by the finite precision of the finite length registers. *Register overflow* occurs when the dynamic range of a register is exceeded. Register overflow is particularly undesirable since it causes full scale errors unless saturation arithmetic is used. This latter feature is not always available in hardware. The overflow problem can be avoided by pre-scaling the input data to the proper range.

It is convenient to regard the contents of the registers as fractions. In this form 16 bit registers can represent numbers from −1 to +32767/32768 in two's complement form. Products of two such numbers remain fractions. A common feature in DSP hardware is the ability to form a double precision product from two single precision numbers. The product appears in a double precision register which can be used to accumulate terms. Thus intermediate results are available in double precision. With the fractional representation, a single precision result can be stored by truncating or rounding the least significant bits in the accumulator. Adding numbers in fractional notation can still result in register overflow. However, if the sum of a number of terms is known to result in a representable number, intermediate overflows (wraparound) can be ignored (assuming two's complement representation for negative numbers) [13].

In the present study the register length is assumed to be 16 bits and data are represented as two's complement fractions. Such a configuration enables a realistic simulation of the coder based on the capabilities of the DSP hardware currently available.

4.2 The Discrete Cosine Transformation

The N-point discrete cosine transformation (DCT) required in the ATC coder is calculated by means of an algorithm which uses an $N/2$-point fast Fourier transform (FFT). The computation is done in place, that is the output data replace the input data [10]. The choice of a particular FFT algorithm depends on the speed of execution and the amount of storage space (for instruction code) required. In general, there exists a trade-off between these two factors. The algorithm selected for this work is a variant of the Radix-2 FFT [13], which requires the block length to be an integer power of 2. This algorithm is suitable for implementation on the current generation of DSP chips which are characterized by limited storage capabilities, but which are fast enough to compute moderate length FFT's at speech data rates.

A major concern in the fixed point computation of FFTs is the problem of register overflow. For the fixed point DFT of a complex sequence $x(n)$ of length N, defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} ,$$

output register overflow is prevented if

$$| x(n) | < \frac{F_{max}}{N} \qquad 0 \leq n < N ,$$

where F_{max} is the largest representable number. An obvious way to prevent overflow is to pre-scale the input data by $1/N$. This method, however, results in poor signal-to-noise ratio (SNR) due to the fact that $\log_2 N$ bits of precision have been discarded at the begining of the computation. A better approach is to distribute the scaling over the various stages of the FFT computation. At each of the $\log_2 N$ stages of the FFT, "butterfly" terms of the form

$$X_{m+1}(p) = X_m(p) + e^{-j2\pi r/N} X_m(q)$$
$$X_{m+1}(q) = X_m(p) - e^{-j2\pi r/N} X_m(q)$$

are formed [13]. Scaling the input data to this stage by two will prevent overflow from occuring. The overall scaling factor remains the same as that due to dividing by N at the begining but more precision is available in the early stages of the FFT process.

Yet another technique is to "normalize" the data only if an overflow occurs. The SNR values for the two latter methods as well as a floating point FFT are plotted in Fig. 4-1 as a function of the FFT length. The input data are uniformly distributed random numbers. Fig. 4-2 is the similar plot for a DCT. Note that before applying to transformation the input sequence was scaled up by a power of two to the full 16-bit range to minimize the truncation error. These plots show that overflow-normalization is the best of the techniques. However, for this method to be

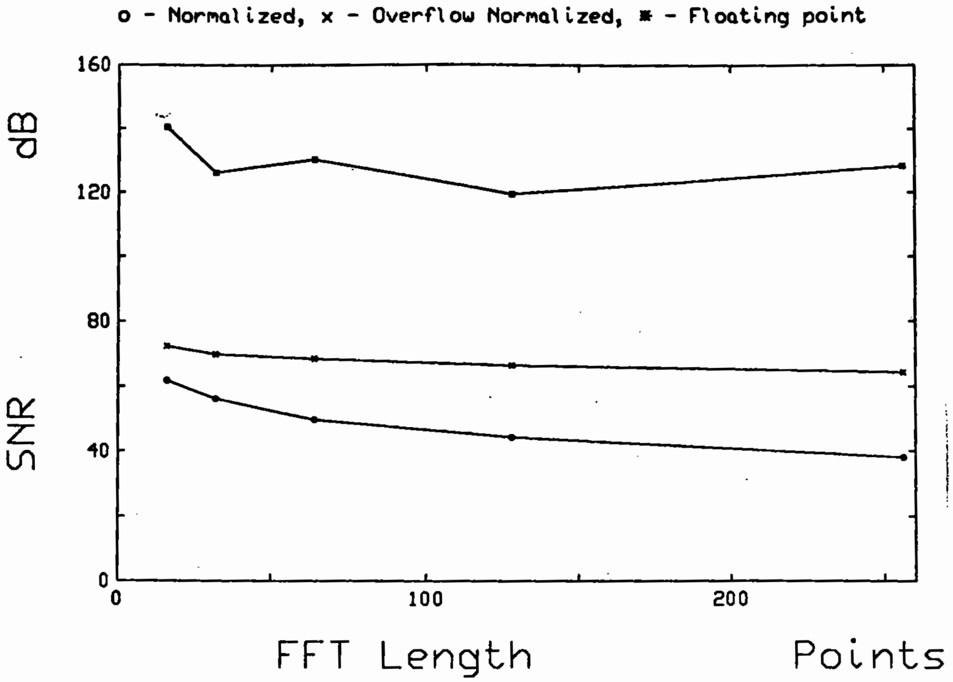


Figure 4-1 **Fixed Point FFT Performance Comparisons**

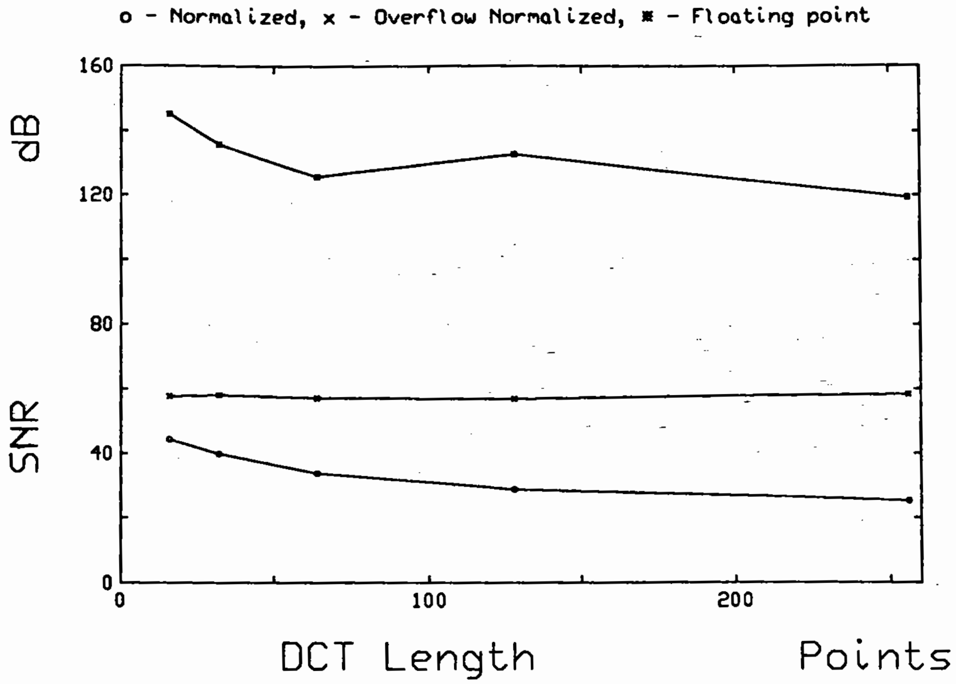


Figure 4-2 **Fixed Point DCT Performance Comparisons**

efficient, hardware traps have to be available to detect overflow conditions. These mechanisms are not available on the current generation of DSP chips.

In our implementation of the ATC, the DCT's were protected against overflow by the distributed pre-scaling method. In order to reduce the loss of precision due to normalization and truncation, it is necessary to scale the input vector up to the full 16-bit range. Equivalently one may postpone the normalization for k stages in the transform where k is the largest power of two by which the input vector can be multiplied without causing overflow. The latter approach was selected since it requires fewer scaling operations. Note that in all cases the scale factor is a power of two, thus reducing the normalization to a simple arithmetic shift operation..

The procedure for scaling the input to the transforms produces a scale factor appying to a block of data. In this way, the number of shifts incurred can be considered to be the exponent in a block floating point representation.

4.3 Determination of the Spectral Parameters

4.3.1 Auto-correlation Function

The all-pole model of the spectrum used in the bit allocation procedure was originally computed as follows [8].

a) The DCT spectrum is squared to get the DCT power spectrum.

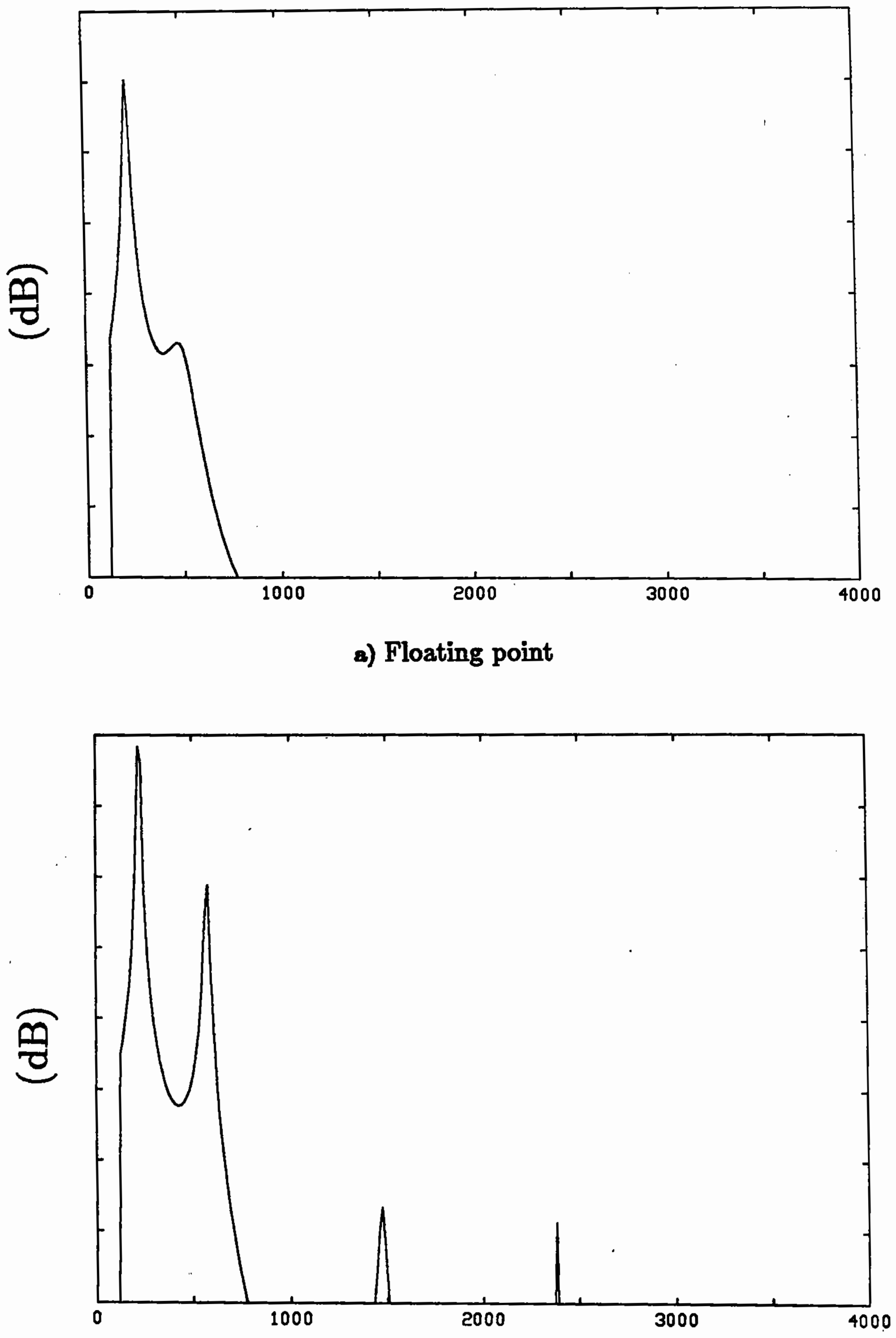
b) The DCT power spectrum is inverse transformed with an inverse DFT. This yields an auto-correlation-like function known as the pseudo auto-correlation function.

c) The first few samples of this function are applied to the Levinson algorithm to obtain the optimal mean-square error all-pole fit. The fit is parameterized by a set of reflection coefficients and a gain factor.

The straightforward application of fixed point arithmetic to the algorithm described above results in serious deterioration of the output speech quality. The poor performance of the coder in this configuration is caused primarily by the fixed point computation of the DCT power spectrum. This operation, requiring the computation of the square of the DCT coefficients, results in the doubling (in dB) of the dynamic range. Therefore, since in fixed point multiplication the least significant 16-bits are lost, it is not possible to get a faithful estimate of the power spectrum. The frequency components lost to such truncation errors result in ill-conditioned correlation equations. The loss of precision due to normalization in the following stage (inverse DFT of the power spectrum) further complicates the problem. The degradation is particularly severe in frames of data in which the energy is concentrated in narrow bands that span a large dynamic range even before the squaring operation. An example of such a frame is displayed in Fig. 4-3. This figure gives the plots of the basis spectrum calculated from coefficients determined using floating point and fixed point DCT power spectrums for the nasal sound "n" in the word "and". The resultant bit assignment vectors are displayed in Fig. 4-4, showing the errors caused by a poor determination of the spectral fit parameters.

One solution suggested by work in LPC coding is to pre-emphasize the input sequence. This serves to reduce the spectral dynamic range by accentuating the high frequencies which correspond to the low energy regions of the speech spectrum. Also, pre-emphasis is often used in coding schemes at low bit rates both to bias the bit assignment in favour of the high frequency bands (which would otherwise be deprived because of their low energy content) and as an effort in noise shaping.

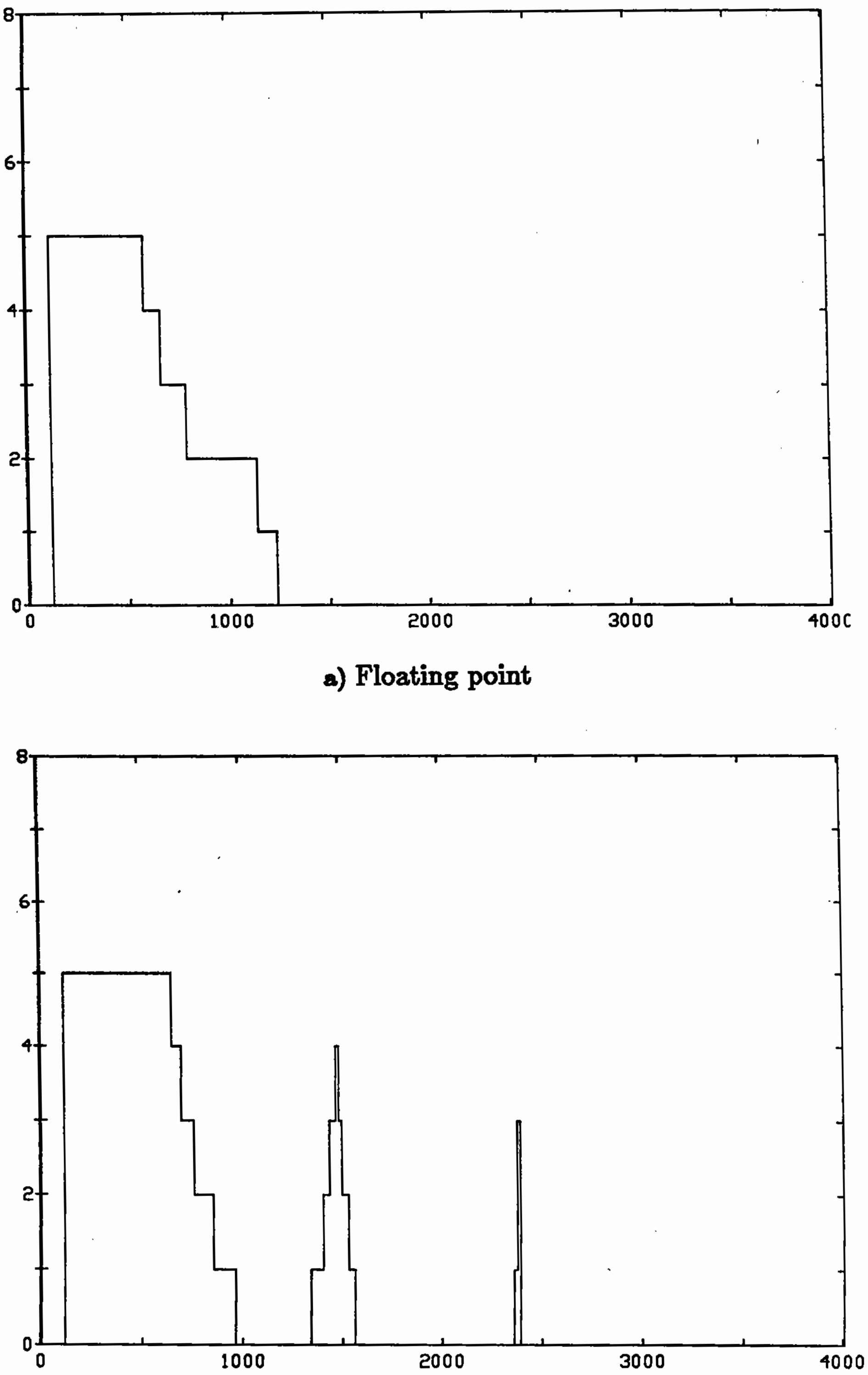
Experiments show that a large pre-emphasis factor (around 0.95) is required to alleviate the dynamic range problem reliably. However, this may not be desirable



a) Floating point

b) Fixed point

Figure 4-3 Basis Spectrum Comparisons



a) Floating point

b) Fixed point

Figure 4-4 Bit Assignment Comparisons

for the following reasons.

a) In some frames a large pre-emphasis factor may cause the bit assignment algorithm to allocate too few bits to the subjectively important low frequency coefficients.

b) It can be shown that for large pre-emphasis factors the DCT ($Y_p(k)$) is related to the DCT of the input sequence before pre-emphasis ($Y(k)$) according to the following expression.

$$Y_p(k) \approx Y(k)|H(k)|\cos\left(\frac{k\pi}{2N}\right)\left\{\tan\left(\frac{k\pi}{2N}\right) + \tan\left(\frac{k\pi}{2N} - Arg[X(k)]\right)\right\}$$

where $H(k) =$ DFT of the pre-emphasis filter

$X(k) =$ DFT of the input sequence

From the discussion above, it can be seen that a large pre-emphasis factor does not affect the DCT spectrum in a simple way, thus drastically reducing its value as a tool in noise-shaping and bit assignment.

An alternative approach is to compute the pseudo auto-correlation function directly. It can be shown that the N-point DCT of the sequence $x(n)$ of length N is equal to the first N samples of the $2N$-point DFT of the sequence $h(n)$ of length $2N$ defined as:

$$h(n) = x(n) + \tfrac{\sqrt{2}-2}{2}\bar{x} \qquad 0 \le n < N$$

$$h(n) = h(2N-1-n) \qquad N \le n < 2N$$

$$\text{where} \qquad \bar{x} = \frac{1}{N}\sum_{i=0}^{N-1} x(i)$$

So the pseudo auto-correlation $R(n)$ corresponding to the inverse DFT of the power spectrum can be obtained by computing the (circular) autocorrelation function of the sequence $h(n)$. Furthermore it can be shown that

$$R(n) = R_{xx}(n) + \frac{1}{2}\sum_{j=0}^{n-1}\left[x(j)x(n-1-j) + x(N-n+j)x(N-1-j)\right] - \frac{N}{2}\bar{x}^2$$

$$0 \le n < N,$$

where $R_{xx}(n)$ is the non-circular autocorrelation function of $x(n)$.

Since only the first few samples of $R(n)$ are needed to compute the basis spectrum, there are several practical advantages in the choice of the direct computation of the pseudo auto-correlation function. Perhaps the most important benefit to be gained from the direct method is a substantial increase in accuracy; the auto-correlation computation can take full advantage of the double-precision product and accumulate feature available on DSP chips. Needless to say the input sequence must be scaled properly to avoid register overflow during such calculations. Another advantage of this technique is a modest reduction of the arithmetic operations (especially if the complexity of the data access and control structure of an FFT algorithm is taken into account). Also, for a multi-chip implementation, the DCT and the basis spectrum computation may be carried out independently and perhaps concurrently.

The main disadvantage of the direct method is that the calculation of the pitch period requires extra effort. However, the benefits of pitch modelling, though noticeable at the longer block lengths, are not so great as to substantially increase the perceived speech quality. With short frame lengths, the pitch information is smeared to the point where pitch modelling is ineffective. For these reasons, the integer implementation does not include pitch modelling to modify the basis spectrum.

The direct auto-correlation method was implemented and tested. The results were significantly improved over a fixed point implementation of the inverse transform of the power spectrum algorithm. The comparison of the two techniques was based on plots of the basis spectrum and the signal-to-noise ratio of the basis spectrum with respect to the power spectrum. In most cases the all-pole fit is almost identical to the one obtained with floating point arithmetic. However, the fit is still poor in frames with large dynamic range, the problem again stemming from ill-conditioned correlation equations. The following solutions were considered.

a) Pre-emphasis of the input sequence: Pre-emphasis does alleviate the problem in most cases, but because of the reasons mentioned earlier it is not a desirable solution.

b) Addition of noise: Low level noise can be added to the DCT spectrum to condition the correlation matrices [9]. The noise level is set to a small fixed fraction of the frame energy.

Experiments were carried out using two types of noise: high-pass noise and white noise. High-pass noise is expected to be more effective since in the case of speech signals the ill-conditioning of the equations is caused by the absence of high frequency components of the spectrum. The high-pass noise filter used was taken from [9] and its Z-transform is:

$$H(z) = \frac{1}{4}(1 - z^{-1})^2$$

The effect of adding corrective noise to the DCT spectrum of the input signal is achieved by adding its autocorrelation function to the pseudo auto-correlation of the signal. The correlation function of noise produced by the high-pass filter given above is:

$$R_{hh}(n) = \frac{3}{8}\delta(n) - \frac{1}{4}\delta(n-1) + \frac{1}{16}\delta(n-2) \qquad 0 \le n < 2N$$

where N is the frame-length.

The addition of high-pass noise requires the addition of the terms above, scaled by the desired noise level, to the first three samples of the pseudo auto-correlation. Once the basis spectrum is computed, the frequency response of the high-pass noise can be subtracted from it. With the noise power set to 0.1% of the signal power the performance of the above scheme was excellent, with SNR values of the basis spectrum (compared to the DCT power spectrum) being generally superior to that of the floating point results based on the power spectrum approach.

Compared to high-pass noise, the addition of white noise is much easier to implement since only one term is involved. Tests show that white noise can be just as effective as high-pass noise, although for a given noise-level the performance of the latter is slightly better. It was also observed that the performance of the algorithm improves with decreasing noise power. Obviously there is a limit below which the noise addition is rendered ineffective by truncation errors. A suitable power level is about 0.05% of the signal energy. For such a small fraction it is not even necessary to subtract the white noise from the basis spectrum. This seems to be a good solution because it is effective yet extremely simple.

4.3.2 Basis Spectrum Parameters

The algorithm for the solution of the linear equations which determine the reflection coefficients, uses a set of intermediate variables (corresponding to the normalized prediction error), which have magnitude less than unity [14]. This normalization is fully compatible with fractional integer notation. This algorithm, as in the usual Levinson recursion used to solve the equations, uses a divide at each stage of the recursive solution. This divide can be implemented using a loop with a conditional subtract instruction available on signal processing chips. All computations for this step use single precision arithmetic with no noticable precision problems.

4.4 Basis Spectrum Parameter Quantization

The basis spectrum is described by a gain parameter and a set of reflection coefficients. For the moment we disregard pitch modelling parameters. The gain is quantized using a one-sided logarithmic quantizer. The reflection coefficients are quantized using log-area companded quantizers. The integer version of the quantization procedure is straightforward for the reflection coefficients because of their inherent normalization to the range $(-1,+1)$. The gain that is derived from the determination of the reflection coefficients is normalized relative to the signal energy. The normalization factor is just the number of bit shifts involved in the pseudo auto-correlation normalization. This scale factor is treated as a separate piece of side information.

4.5 Basis Spectrum Computation

4.5.1 Reflection to Predictor Coefficient Transformation

The reflection coefficients produced by the algorithm of the previous step must be transformed to a new set of predictor or error filter coefficients, in preparation for the evaluation of the predictor filter or inverse filter response. The transformation operation needs careful attention to normalization of the data values. The predictor coefficients, in theory, have an unlimited range of values. Previous work in Residually excited LPC (RELP) coding of speech [15] has shown that a range of -4 to +4 is sufficient to represent the predictor coefficient values. Other workers have indicated that a larger range (say -8 to +8) is necessary if the spectrum of the signal has a large dynamic range [9]. This larger range can be traced to an ill-conditioned set of equations, in which a range of filter coefficients all give nearly equal performance. The resolution of this problem is in part the "noise" added to the spectrum to better condition the solution. With this noise, the range of -4 to +4 seems to be adequate in all cases.

4.5.2 Computation of the Basis Spectrum

Once the all-pole model is determined, the filter parameters are used to compute the basis spectrum which is required for the purposes of bit assignment and the normalization of the range of the quantizers.

The algorithm for the computation of the basis spectrum, developed and tested in floating point arithmetic consists of the following steps.

a) First the reflection coefficients are converted to the analysis filter coefficients.

b) The frequency response of the analysis filter is calculated by computing the magnitude of the $2N$-point (N is the frame length) DFT of the filter coefficients.

c) The basis spectrum is obtained by inverting the magnitude of the analysis filter.

There are two major problems involved with the implementation of this procedure in fixed point arithmetic. One is that during the computation of the $2N$-point FFT up to $\log_2 2N$ bits of precision may be lost to the normalization stages involved. The other problem is the squaring operation required to calculate the magnitude of the DFT, which leads to the doubling (in dB) of the dynamic range. The losses in performance due to this type of computation also occurred as discussed previously in connection with the all-pole fit. Obviously the numerical errors mentioned here have the greatest impact on the regions of the spectrum where the magnitude of the analysis filter is small. But such regions correspond to areas on the basis spectrum which have the largest magnitudes, thereby receiving most of the total bit allocation (or all of it, depending on the transmission rate). So such errors are particularly devastating and must be avoided.

As alternatives to the procedure described above the following three methods were considered.

4.5.2.1 Synthesis Filter Correlation Function Transform

The auto-correlation function of the synthesis filter was calculated, making use of the fact that its first M (number of poles of the LPC model) samples are equal to those of the (pseudo) auto-correlation of the input signal, and that the rest are obtained by filtering the first M values by the synthesis filter. Note that since the synthesis filter has an infinite impulse response (IIR) then the desired correlation function is also an infinite sequence. However, in most cases the correlation function dies out rapidly, so it was hoped that it could be truncated and transformed by a DFT to give the basis spectrum directly.

The results were unacceptable. Only the large magnitude portion of the basis spectrum was obtained with any accuracy.

4.5.2.2 Synthesis Filter Pulse Response Transform

The parameters of the all-pole model were used to obtain the synthesis filter coefficients. Then the impulse response of the filter was directly calculated and truncated (since the synthesis filter is IIR). The basis spectrum was obtained by computing the magnitude of the DFT transform of the truncated impulse response. The truncation errors incurred in these computations are not of major concern since the regions of interest have the largest magnitudes and are not seriously affected by such errors.

The results of this scheme were generally satisfactory. However, in terms of the number of arithmetic operations involved it is rather expensive.

4.5.2.3 Analysis Filter Pulse Response Transform

The coefficients of the analysis filter were obtained from the all-pole model parameters. The auto-correlation function of the analysis filter was then calculated and DFT transformed to get the frequency response of the analysis filter, i.e. the inverse of the basis spectrum. This method offers important advantages; the analysis filter has only $M+1$ coefficients (M is the number of poles in the model of the spectrum) and so its correlation function is short and inexpensive to compute. Also, since the correlation function is a symmetric sequence with only $2M+1$ nonzero elements, its DFT can be written in the following form:

$$X(k) = x(0) + (-1)^k x(\frac{2N}{2}) + 2\sum_{n=0}^{M} x(n)\cos(\frac{2\pi}{2N}nk) \qquad 0 \le k < 2N$$

$$\text{where} \qquad N = \text{frame length}^*$$

Or, for $M+1 < N$ (as is the case, usually):

$$X(k) = x(0) + 2\sum_{n=0}^{M} x(n)\cos(\frac{2\pi}{2N}nk) \qquad 0 \le k < 2N$$

This formulation lends itself to inexpensive and accurate computation on a DSP chip; inexpensive because M is usually small, and accurate since the *double-precision product and accumulate* feature of the DSP chip may be utilized. The gain in precision becomes more appreciable in comparison to an FFT, during the computation of which up to $\log_2 2N$ bits may be lost in normalization stages.

The increase in accuracy is invaluable because when dealing with the inverse of the basis spectrum the regions of utmost importance have the smallest magnitudes and are thus most vulnerable to round-off and trucnation errors.

The simulation of this method shows that single-precision arithmetic operations do not yield sufficiently accurate results and one has to resort at least to "semi-double precision" computations. Semi-double precision (SDP) is used to mean that in the DFT computation the $x(.)$ values (the correlation function of the analysis filter)

[*] A $2N$-point DFT is required to obtain an N-point basis spectrum. Note that since the DFT of a real sequence is symmetric, only the first N points have to be computed.

are represented by double-precision numbers while the cosine terms are regular 16-bit scaled integer values. SDP is easy to implement; first the component of the DFT contributed by the lower word of the double length correlation samples is computed. It is then summed up (after shifting to the appropriate position) with the contribution of the higher words. Thus in terms of the number of arithmetic operations, SDP is roughly equivalent to two single-precision computations. Only $M+1$ double precision values need be stored— the $M+1$ samples of the auto-correlation sequence.

The double-precision results are converted to 16-bit values by extracting the most significant word of the result shifted to the left so as to represent the smallest (hence the most important) values with at least two bits. The samples corresponding to larger components which overflow the 16-bit registers as the result of the left shift operation are set to the largest positive 16-bit value. Alternatively, the performance of the coder can be improved by making use of the fact that values less than 2 (including negative numbers) can not occur. These values could be used to increase the dynamic range.

Experiments were carried out implementing the above scheme combined with the SDP calculation of the DFT. The dynamic range was extended by utilizing the "impossible" values in the following manner:

$$\begin{array}{lll} 2^{15} \le x < 2^{16} & \text{was stored as} & -\frac{x}{2} \\ 2^{16} \le x < 2^{17} & \text{was stored as} & 0 \\ 2^{17} \le x & \text{was stored as} & 1 \end{array}$$

The results of the test show that the basis spectrum obtained according to this procedure is sufficiently accurate. There is little or no degradation of the quality over the floating point algorithm described at the begining of this section. In view of the satisfactory performance of this technique and its simplicity (in spite of the SDP computations the overall operation count is well below all other methods discussed here), it was selected for the computation of the basis spectrum (note that in fact the inverse of the basis spectrum is calculated).

Once the inverse of the basis spectrum is obtained it is necessary to compute its square root for the purpose of the normalization of the range of the quantizers.

4.6 Bit Assignment

The bit assignment procedure of Section 3.3 was modified for the integer version of ATC to utilize the inverted basis spectrum. The procedure differs from that originally implemented in requiring the inverse of the weighted distortion values and in having the sorting to arrive at the bit assignment done in increasing rather than decreasing order.

4.7 Side Information

In an ATC coder the side-information consists of the LPC filter parameters. In the fixed point realization of the coder it is also necessary to transmit scale factors used in various stages of the coding algorithm in order to increase the dynamic range. These data can be combined into a single value not requiring more than 7 bits per frame.

4.8 Quantization of the DCT coefficients

The DCT coefficients are coded with quantizers determined by the the bit assignment. Before quantization, the coefficients are normalized by the square root of the basis spectrum value. This square root operation is implemented using a Newton-Raphson iteration. Before quantization, the DCT coefficients are multiplied by the square root of the inverse basis spectrum. In the reconstruction procedure (at the receiver), the quantized values is restored by dividing by the square root of the inverse basis spectrum.

4.9 Experimental results

In order to assess the fixed point algorithm, the ATC coder was simulated on a DEC VAX 11/780 computer using the techniques developed in this work. In the implementation of the fixed point coder the register length was taken to be 16 bits and data were represented as two's complement scaled fractions.

The simulations were carried out using frame sizes of 64 and 256 points. In each case the data frames were selected in such a way as to include an overlap region of $\frac{1}{16}$ their length in order to reduce the noise caused by discontinuities at the frame boundaries.

The main objective of this project was to develop a fixed point algorithm for ATC coding of speech. Since fixed point arithmetic is prone to a host of numerical problems and errors, the major concern was to find a technique with as little degradation over the floating point algorithm as possible. Therefore it is reasonable to base judgements of the quality of the fixed point coder on comparisons with the floating point one.

In order to make the comparisons as fair as possible features such as pre-emphasis, pitch modelling and spectral weighting were disabled. Furthermore, one piece of side information, namely the LPC gain parameter, was not quantized due to the fact that in floating point representation the gain is unbounded and therefore more "difficult" to quantize.

Two sentences each uttered by a group of speakers were coded by the two ATC algorithms operating at the rate of 9.6 kb/s. Each group consisted of a male and a female speaker. Table 4-1 gives the signal-to-noise ratio (SNR) and the segmental signal-to-noise ratio (SSNR) values for the frame length of 256 points. The SSNR figure is the average signal-to-noise ratio, in dB, calculated for 16 msec segments of signal. Experience has shown that changes in SSNR are generally better correlated with changes in speech quality than changes in SNR. The coder parameters for this configuration are given in Appendix A.

Note that according to Table 4-1 the fixed point algorithm performs slightly better than the floating point one. This is probably due to the better spectral fit observed earlier. Listening tests reveal that while the output of the fixed point coder sounds less muffled than the output of the floating point one, a little quantization noise is audible in the former. The quantization noise is caused by the truncation errors in the transformation routines and can be reduced or eliminated by either shortening the frame length or scaling the input to the DCT routines up to the full range. In any case, the difference between the outputs of the two coders is extremely small. Furthermore, full quantization of the side information does not decrease the performance of the fixed point algorithm.

The SNR/SSNR figures for the frame size of 64 points are given in Table 4-2 followed by the coder parameters in Appendix B. In this configuration the side information is obtained by an averaging process and is transmitted only once every four frames in order to keep the transmission rate of the side information below 2 kb/s.

Subjective tests indicate a drop in the quality of the ATC coder at the frame length of 64 points. This is attributed to the interpolation of the side information as well as the fact that the theoretical performance of the DCT decreases with the size of the frame*. However, listening tests also show that with 64 point frames the

* Recall that DCT converges in mean-square to KLT only in the limit of large block length

Speaker	Sentence	Floating Point	Fixed Point
F1	A	9.4 / 7.1	11.3 / 7.9
M1	A	10.4 / 8.7	12.0 / 9.1
F2	B	9.7 / 9.9	11.5 /10.2
M2	B	12.6 /10.5	13.9 /10.8

SNR (dB) / Seg. SNR (dB)

Table 4-1 Coder Performance for a Frame Length of 256

Speaker	Sentence	Floating Point	Fixed Point
F1	A	12.3 / 8.0	12.0 / 7.7
M1	A	12.1 / 9.1	11.3 / 8.7
F2	B	12.0 /10.1	11.4 / 9.3
M2	B	13.2 / 9.7	12.5 / 9.3

SNR (dB) / Seg. SNR (dB)

Table 4-2 Coder Performance for a Frame Length of 64

Speakers:

F1 :	Female
M1 :	Male
F2 :	Female
M2 :	Male

Sentences:

A : Open the crate, but don't break the glass.
B : The birch canoe slid on the smooth planks.

outputs of the fixed point and the floating point coders are indistinguishable despite the apparent differences in the SNR values of Table 4-2.

Finally it is worth noting that the figures given in the tables were produced using a set of coder parameters which suppressed the transmission of the first 125 Hz of the DCT spectrum. The elimination of this band, which is done to enhance the perceptual quality of the ATC coders, results in an increase in the mean-square error of the output. This is to say that without this operation it is possible to increase the SNR/SSNR values quoted in this report by up to 6 dB, depending on the speaker and the utterance.

5. Summary and Conclusions

This report has examined several aspect of ATC coding of speech. The first part of the work was directed at improving the quality and decreasing the computational burden of the coding algorithm. A second aspect of this study, was the assessment of the performance of the algorithm with background acoustical interference superimposed on the speech to be coded. The third and major part of the work reported relates to the production of a fixed point version of the ATC algorithm.

The main result of this work is the demonstration of an ATC coder with all computations carried out in fixed point arithmetic. The computational steps employed are fully compatible with known DSP chip architectures. Based on the experience gained in implementing the fixed point operations, it is felt that an ATC coder and decoder can be built with a relatively simple hardware configuration built around readily available digital signal processing chips.

The specific work performed for this study can be summarized as follows.

a) A new pitch modelling technique was investigated. This method results in a significantly better fit of the basis spectrum to the DCT data. However, the overall speech quality at a bit rate of 9.6 kb/s is only slightly improved over the case in which no pitch modelling is used. At this bit rate, the added computational complexity is probably not warranted for a real time coder. The promise of this technique is probably at still lower bit rates where the coder is getting "starved" for bits.

b) A revised bit allocation was devised. It produces the same bit assignment as the technique previously used but with significantly fewer computations and considerably less storage. A technique to efficiently apply a spectral weighting to the bit allocation procedure by modifying the table of marginal distortions was also discussed.

c) Experimental results show that the ATC method is relatively insensitive to background acoustical noise. Three types of interference were tested—narrow band, wide band and speech-like. Of these, the wide band interference was the most disturbing. The effect on the speech quality of the other types of interference was minimal.

d) Scaling techiques appropriate to the calculation of the FFT and DCT using fixed point arithmetic were implemented. A distributed scaling of the values along with an initial normalization was found to be useful. This normalization procedure applied in each frame can be thought of as comprising a block floating point technique.

e) It was found that considerable care had to be exercised in the computation of the spectral parameters and from them, the basis spectrum, in fixed point arithmetic. Alternate derivations of the various quantities were necessary in some cases in order to maintain sufficient precision in the final results. A direct computation of the pseudo auto-correlation coefficients from the time waveform rather than from the DCT spectrum was found to be both efficient and accurate. Numerical conditioning of the correlation equations was improved by compensating for the bandlimiting of the input signal with the simple but effective expedient of adding a corrective noise term.

f) The solution for the spectral parameters themselves posed no problem in fixed point arithmetic. The calculation of the basis spectrum from these parameters had to be modified. A transformation directly from the correlation of the predictor filter correlation to the inverted basis spectrum was used to help retain precision. This transformation was implemented directly instead of using an FFT algorithm. In addition, the computation has to be carried out in extended precision arithmetic in order to maintain accuracy. With this extended precision, the resulting spectral match is often better than the match with floating point arithmetic.

g) The coder using exclusively fixed point arithmetic has essentially the same performance as the one using floating point arithmetic.

Appendix

This section contains a listing of the parameter files used in the coder simulations of Section 4.9. It is meant to be used in conjunction with a listing of the simulation program. It is reproduced here as a record of the coder configuration used in the tests.

A. Frame Length 256

The following is the parameter file for an ATC coder with the frame size of 256 points running at 9.6 kb/s.

```
!
!  ATC coder 9.6 kb/s, 256 point window, side information
!  transmitted every frame
!
Bits/Frame=232
Max_Bits/Coef=5
Super-Frame_Size=256
Window=(Length=256,Type=Trapezoidal,Shape_Factor=Ø.Ø)
Frame_Advance=24Ø
Pre-emphasis=Ø.Ø
Gamma_Factor= Ø.Ø
Noise_Factor=Ø.Ø
Pitch_Factor=Ø.Ø
Freq_Cutoff=125
Correction_Noise=Ø.ØØØ5
Distortion_Table=( - !Marginal distortions for a MMSE Gaussian quantizer
  Values=(6.366E-1,2.459E-1,8.295E-2,2.5Ø5E-2,6.996E-3, -
          1.861E-3,4.8Ø6E-4,1.221E-4))
Dct_Quantizer=(Type=Gaussian)
Energy_Quantizer=(Number_Levels=Ø, Parameter=1ØØ, -
                  Min_Value=Ø, Max_Value=32767)
Reflection_Coef =(Number_Coef=8) ! Number of LPC poles
Reflection_Coef =( -
   K1=(Number_Levels=64, -
       Min_Value=-Ø.9728, Max_Value=Ø.7677), -
   K2=(Number_Levels=64, -
       Min_Value=-Ø.3, Max_Value=Ø.89Ø2), -
   K3=(Number_Levels=64, -
       Min_Value=-Ø.8184, Max_Value=Ø.5677), -
   K4=(Number_Levels=64, -
       Min_Value=-Ø.3ØØ6, Max_Value=Ø.7647))
Reflection_Coef =( -
   K5=(Number_Levels=64, -
       Min_Value=-Ø.4668, Max_Value=Ø.3822), -
   K6=(Number_Levels=32, -
       Min_Value=-Ø.2Ø22, Max_Value=Ø.6353), -
   K7=(Number_Levels=16, -
       Min_Value=-Ø.2586, Max_Value=Ø.4Ø79), -
   K8=(Number_Levels=16, -
       Min_Value=-Ø.1651, Max_Value=Ø.58Ø6))
Fix_opt=15
PARAMETER_FILE=SYS$INPUT
```

B. Frame Length 64

The following is the parameter file for an ATC coder with the frame size of 64 points running at 9.6 kb/s.

```
!
!  ATC coder 9.6 kb/s, 64 point window, side information transmitted
!  once every 4 frames
!
Bits/Frame=58
Max_Bits/Coef=5
Super-Frame_Size=256
Window=(Length=64,Type=Trapezoidal,Shape_Factor=Ø.Ø)
Frame_Advance=6Ø
Pre-emphasis=Ø.Ø
Gamma_Factor= Ø.Ø
Noise_Factor=Ø.Ø
Pitch_Factor=Ø.Ø
Freq_Cutoff=125
Correction_Noise=Ø.ØØØ5
Distortion_Table=( - !Marginal distortions for a MMSE Gaussian quantizer
  Values=(6.366E-1,2.459E-1,8.295E-2,2.5Ø5E-2,6.996E-3, -
          1.861E-3,4.8Ø6E-4,1.221E-4))
Dct_Quantizer=(Type=Gaussian)
Energy_Quantizer=(Number_Levels=Ø, Parameter=1ØØ, -
                  Min_Value=Ø, Max_Value=32767)
Reflection_Coef =(Number_Coef=8) ! Number of LPC poles
Reflection_Coef =( -
   K1=(Number_Levels=64, -
       Min_Value=-Ø.9728, Max_Value=Ø.7677), -
   K2=(Number_Levels=64, -
       Min_Value=-Ø.3, Max_Value=Ø.89Ø2), -
   K3=(Number_Levels=64, -
       Min_Value=-Ø.8184, Max_Value=Ø.5677), -
   K4=(Number_Levels=64, -
       Min_Value=-Ø.3ØØ6, Max_Value=Ø.7647))
Reflection_Coef =( -
   K5=(Number_Levels=64, -
       Min_Value=-Ø.4668, Max_Value=Ø.3822), -
   K6=(Number_Levels=32, -
       Min_Value=-Ø.2Ø22, Max_Value=Ø.6353), -
   K7=(Number_Levels=16, -
       Min_Value=-Ø.2586, Max_Value=Ø.4Ø79), -
   K8=(Number_Levels=16, -
       Min_Value=-Ø.1651, Max_Value=Ø.58Ø6))
Fix_opt=15
PARAMETER_FILE=SYS$INPUT
```

References

1. P. Kabal, "Adaptive Transform Coding of Speech at 9.6 kb/s", INRS-Télécommunications, Technical Report 82-06, May 1982.

2. R. Pinnell, "Adaptive Transform Coding of Speech Signals", INRS-Télécommunications, Technical Report 82-07, May 1982.

3. J. Turner, D. C. Stevenson, P. Kabal and P. Mermelstein, "A comparative study of digital coding techniques at 16 kb/s and below", IEEE Canadian Communications and Power Conference, October 1980.

4. J. Markel and A. H. Gray, *Linear Prediction of Speech*, Springer-Verlag, 1976.

5. A. K. Jain, "A sinusoidal family of unitary transforms", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. PAMI-1, pp. 356–365, Oct. 1979.

6. R. Zelinski and P. Noll, "Adaptive transform coding of speech signals", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 299–309, Aug. 1977.

7. J. M. Tribolet and R. E. Crochiere, "Frequency domain coding of speech", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-27, pp. 512–530, Oct. 1979.

8. D. Sloan, "Adaptive Transform Coding of Speech", M.Eng. Thesis, Department of Electrical Engineering, McGill University, July 1979 (also INRS-Télécommunications, Technical Report 79-05, July 1979).

9. B. S. Atal, "Predictive Coding of Speech at Low Bit Rates", *IEEE Trans. Commun.*, vol. COM-30, pp. 600–614, April 1982.

10. P. Kabal, "In place computation of the discrete Fourier and Discrete Cosine Transforms", paper in preparation.

11. A. Segall, "Bit allocation and encoding for vector sources", *IEEE Trans. Inform. Theory*, vol. IT-6, pp. 162–169, March 1976.

12. D. E. Knuth, *The Art of Computer Programming, Volume 3/ Sorting and Searching*, Addison-Wesley, 1973.

13. A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice-Hall, 1975.

14. J. LeRoux and C. J. Gueguen, "A fixed point computation of partial correlation coefficients", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 257–259, June 1977.

15. P. Kabal, "Feasibility Study of a Hardware Implementation of a 4.8 kb/s RELP Speech Coder", INRS-Télécommunications, Technical Report 81-08, May 1981.