Voice-excited predictive coder (VEPC) implementation on a high-performance signal processor

by C. Galand
C. Couturier
G. Platel
R. Vermot-Gauchy

In this paper, we discuss the implementation of a medium-bit-rate linear prediction baseband coder on an IBM bipolar signal processor prototype having a high processing capacity. We show that the implementation of our algorithm requires a processing load of 5 MIPS, with a program size of 5K instructions. We then discuss the application of our coder in a normal telephone environment, which requires mu-law to linear PCM conversion and other signal processing functions such as voice activity detection, automatic gain control, echo control, and error recovery. Quality evaluation tests are also reported which show that this type of coder, operating at 7.2 kbps, allows the transmission of telephone speech with communications quality. Moreover, obtained intelligibility scores and speaker recognition levels are high enough to demonstrate that this coder is a good candidate for telephony applications such as digital trunk transmissions, satellite speech communications, secure voice communications, and audio distribution systems.

Introduction

The availability of high-performance LSI microprocessors in conjunction with recent digital speech processing developments allows low-cost real-time speech compression to be considered for many applications such as digital trunk transmissions, satellite speech communications, secure voice communications, and audio distribution systems. In this paper, we discuss the implementation of a medium-bit-rate coder on a new powerful IBM signal processor prototype described in a companion paper [1]. We show that this coder can provide communications-quality speech and is therefore a good candidate for the applications mentioned above.

In the first part of the paper, we briefly recall the speech compression algorithm presented in [2]. The Voice-Excited Predictive Coder (VEPC) is a baseband residual-excited linear prediction vocoder which includes an original encoding of the baseband signal by a subband coder.

In the second section of the paper, we discuss the application of the digital speech coder in a telephone environment, and we show that it requires additional signal processing functions such as Voice Activity Detection (VAD), Automatic Gain Control (AGC), echo control, and

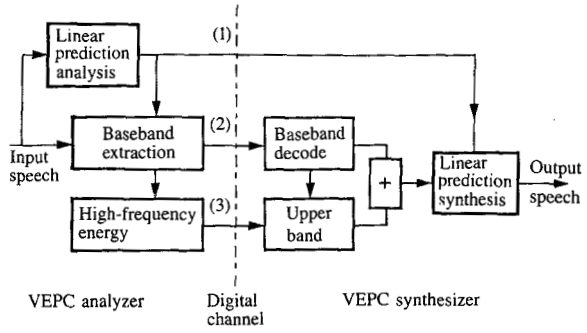

Figure 1

VEPC general block diagram.

error recovery. We then describe an efficient VAD algorithm, based on signal tracking in both time and frequency domains, which permits detection of speech signals in any high-level background noise, provided this noise remains quasi-stationary for short time periods. We subsequently propose an AGC algorithm for level control of telephonic input signals and describe an original echo-canceling algorithm which features a fast start-up operating mode. We then discuss error recovery procedures: In a digital speech transmission network, the speech is first compressed and packetized at the emitter; it is then transmitted to the receiver over digital lines. Transmission errors may occur, but standard error recovery procedures cannot be used because of the real-time constraint inherent in speech transmission. However, we show that one can take advantage of the native redundancy of the speech signal to interpolate missing speech segments from previously received segments.

In the third section, quality evaluation tests are reported. These show that the VEPC coder, operating at 7.2 kbps, provides communications-quality speech with a degree of intelligibility sufficient for telephony applications and good speaker recognition.

In the last section of the paper, we discuss VEPC implementation on the new IBM signal processor prototype. We note the size of the microcode and the corresponding cycle figures for each signal processing task, and we show that the implementation of the VEPC algorithm imposes a processing load of 5 millions of instructions per second (MIPS) and a program size of 5K instructions.

VEPC basic algorithm

The Voice-Excited Predictive Coder, originally proposed in [2], was later adapted so that it could be operated at 7.2 kbps and was implemented on the IBM signal processor prototype [1]. The basic architecture of the VEPC, shown in **Figure 1**, consists of two parts, the analyzer and the synthesizer.

- In the *analyzer*, three types of information are extracted:
  1. Spectral information contained in the speech wave which is characterized by linear prediction analysis, resulting in a set of predictor parameters.
  2. Baseband information obtained by band-limiting (300–1000 Hz) and subsequently subsampling at 2 kHz the residual (or excitation) signal. This residual signal, which results from the inverse filtering of the speech signal by its optimum predictor, carries the quasi-periodicity due to glottal excitation.
  3. High-frequency energy information corresponding to the part of the spectrum (1000–3400 Hz) which has been removed from the excitation signal by low-pass filtering.

These three types of informations are quantized, multiplexed, and transmitted to the synthesizer.

- In the *synthesizer*, the received information is demultiplexed. Then four basic operations are performed:
  1. Decoding the baseband signal and upsampling to 8 kHz.
  2. Upper-band generation, in which a high-frequency signal (1–4 kHz) is synthesized by nonlinear distortion and bandpass filtering of the baseband signal.
  3. Generation of a "pseudo-excitation" signal by combining the baseband signal and the high-frequency signal.
  4. Filtering of this pseudo-excitation signal by the all-pole filter corresponding to the vocal tract predictor.

The functions of these various processing blocks are discussed below. More details can be found in [2, 3].

- *Analyzer*

A detailed block diagram of the analyzer is shown in **Figure 2**. After low-pass Nyquist filtering, the speech signal is sampled at 8 kHz and quantized to 8 bits, according to mu-law compression. Although it introduces more distortion and has a lower dynamic range than linear 12-bit analog-to-digital converters, this type of A/D converter was chosen because of cost and size. Moreover, the additional distortion is negligible when compared to the distortion introduced by the VEPC coder. Finally, as shown later, dynamic range can be extended by using automatic gain control (AGC). The first digital processing task consists in decoding the samples and getting their 12-bit values. Then a spectral analysis is performed by the autocorrelation method with adaptive pre-emphasis. The analysis is performed on a block-by-block basis. Due to the nonstationary nature of the speech signal, the block length must be taken in the range 10–40 ms. Because of the slow mechanical changes in the vocal tract,

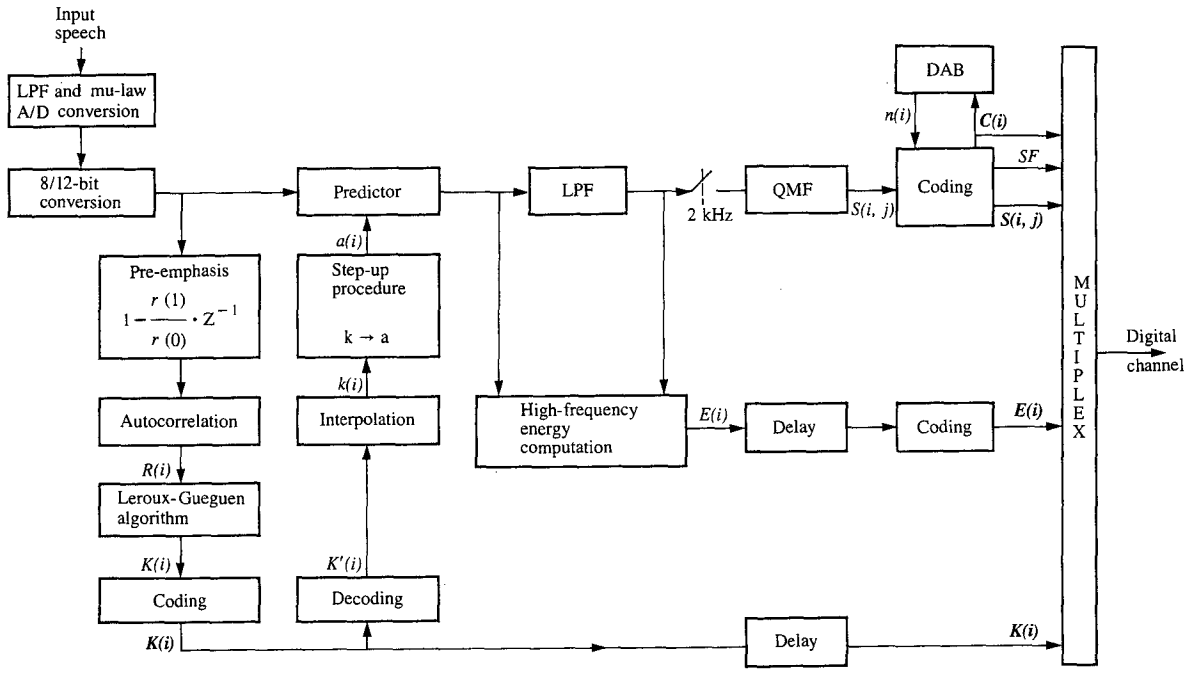

Figure 2

VEPC analyser.

during each block the signal can be considered to be quasi-stationary, and the linear prediction model is therefore correct. A block length of 20 ms, corresponding to 160 speech samples, has been selected. If we assume a 7.2-kbps bit rate, each block results in a 144-bit frame while providing a good trade-off between overhead cost and analysis accuracy. For each block of 20 ms, the analysis provides a set of eight partial correlation (PARCOR) coefficients which are encoded and multiplexed within the transmitted frame. In fact, this time-stepwise updating of the model filter may produce transient resonances which can be canceled by linearly interpolating the PARCOR coefficients $K(i)$ ten times per block. The partial correlation coefficients are obtained as shown in Fig. 2. The speech signal is first pre-emphasized with the optimum pre-emphasis factor $r(1)/r(0)$ [4], where $r(0)$ and $r(1)$ denote the first two autocorrelation coefficients of the input signal. Then the autocorrelation coefficients $R(i)$ of the pre-emphasized signal are computed for $i = 0, \cdots, 8$ and are used to derive the PARCOR coefficients according to the Leroux-Gueguen algorithm [5]. Note here that we use this algorithm instead of the classical Levinson algorithm [4] because we agree with the authors' claim that its implementation requires a lower dynamic range and can therefore easily accommodate a fixed-point computation.

After encoding, local decoding, and interpolation, the PARCOR coefficients are used to derive the direct form coefficients $a(i)$ of the predictor by a step-up procedure [4], and furthermore the residual signal is derived by inverse filtering using these coefficients. Note that we use the direct form instead of the lattice form predictor because, as can be seen in Fig. 7, shown later, it better fits the architecture of our signal processor. Note also in Fig. 2 that $\boldsymbol{K(i)}$, $K'(i)$, and $k(i)$ respectively denote the coded, decoded, and interpolated values of the coefficients $K(i)$. The residual signal is then low-pass filtered to 1000 Hz and subsequently downsampled at 2 kHz, resulting in the residual baseband signal. For each 20-ms block, the baseband signal contains 40 samples. Simultaneously the energy of the removed high-frequency spectrum, which has been discarded by low-pass filtering, is determined. It has been found necessary to determine this energy every 10 ms, that is, twice per block. These energy values $[E(i);\ i = 1, 2]$ are then encoded, resulting in two values $[\boldsymbol{E(i)},\ i = 1, 2]$, which are multiplexed within the transmitted frame.

The baseband is then split into eight subbands by a bank of quadrature mirror filters (QMFs) [3]. After subsequent 1/8 decimation, each subband signal contains five samples. Subbands 1 and 8 are not significant and are discarded, allowing us to reduce both the baseband bandwidth and the

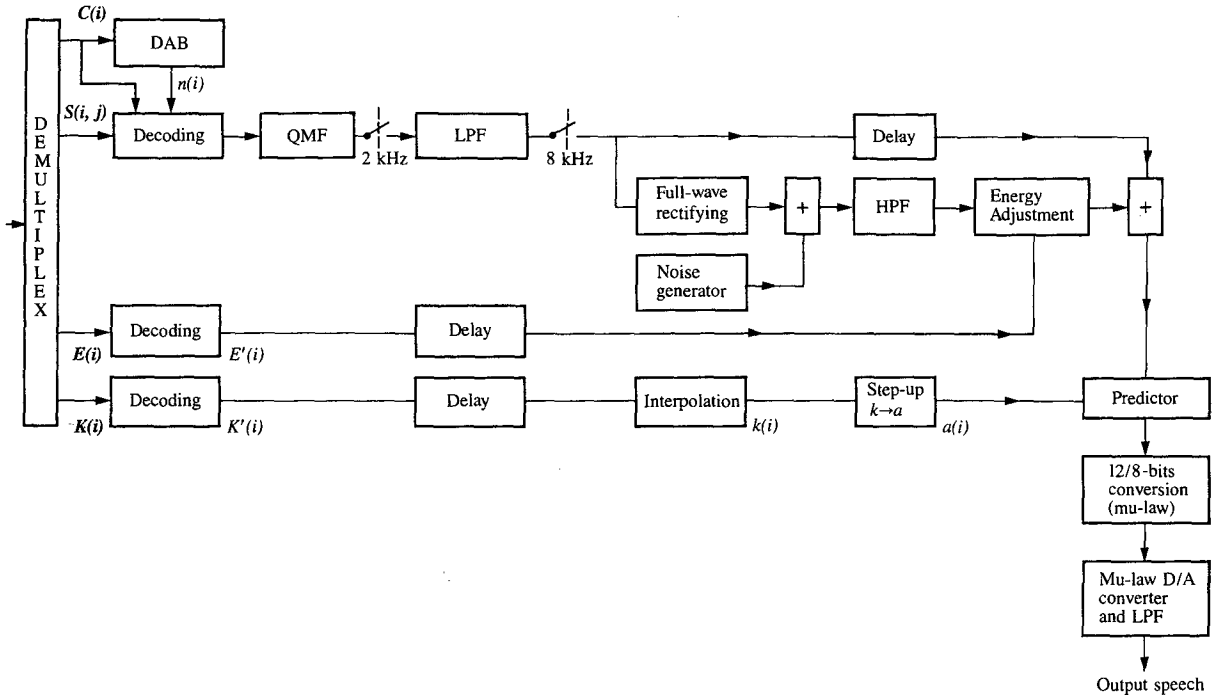

Figure 3

VEPC synthesizer.

filtering processing. At this step, the baseband signal contains only 30 samples. Each of the six subband signals [$S(i, j)$; $i = 1, \cdots, 6$; $j = 1, \cdots, 5$] is then quantized using block companding (BCPCM) techniques [3]. The number of bits [$n(i)$; $i = 1, \cdots, 6$] for the subband signals is adaptively allocated on a block-by-block basis. The bit allocation algorithm specifies the optimum number of bits to be used in each subband. This dynamic allocation of the bit resources (DAB) is in fact very efficient, since the coding gain over adaptive PCM can be as high as 16 dB [3]. In other words, the coding of the baseband signal by a subband coder results in a 16-dB signal-to-noise ratio improvement over that obtained with adaptive PCM coding. This high gain can be explained by the fact that the bandwidth of each subband is narrow (125 Hz). Thus, even for low-frequency-pitch speakers some of the subbands contain pitch harmonics, whereas other subbands have less energy and can therefore be encoded with fewer bits. The resulting baseband information consists of six subband characteristics and six encoded subband signals organized as macrosamples. For each block, the subband characteristics [$C(i)$; $i = 1, \cdots, 6$] are defined as the maximum absolute-valued sample in each subband of this block. A macrosample is defined as a set of six samples occurring simultaneously in each of the six subbands. The total number of bits per macrosample determines the information rate used for the baseband signal. It must be noted that the bit allocation is not transmitted, since it is determined in the same way at the emitter and at the receiver from the transmitted subband characteristics. Due to the limited dynamic range of our fixed-point implementation, a 1-bit-scale factor is introduced for the rescaling of low-amplitude subband samples prior to quantization. The coded characteristics [$\boldsymbol{C(i)}$; $i = 1, \cdots, 6$], the quantized subband samples [$\boldsymbol{S(i, j)}$; $i = 1, \cdots, 6$; $j = 1, \cdots, 5$], and the scale factor SF are multiplexed within the transmitted frame.

- *Synthesizer*

The synthesizer is described with reference to **Figure 3**. The transmitted baseband information is demultiplexed, decoded, and reconstructed through the bank of QMFs. Then its sampling frequency is increased by interpolation and filtering from 2 kHz to the original 8-kHz rate.

The baseband spectrum is then spread by means of a nonlinear distortion technique (full-wave rectifying) which expands to 4 kHz the harmonic structure due to the pitch periodicity. In the case of unvoiced sounds and especially for the fricative sounds (dominant high frequency), the baseband spectrum may lack definition to generate accurately such high-frequency signals. Consequently, it is

necessary to introduce a noise generator, at a very low relative level (say −50 dB when compared to typical voiced sounds). The output noise signal is then added to the full-wave-rectified baseband signal. In the case of voiced sounds, the noise signal is masked and does not corrupt the baseband signal, but in the case of unvoiced sounds its energy is sufficient to generate accurately the missing high-frequency part of the signal. The resulting signal is then high-pass filtered at 1000–4000 Hz, and its energy is adjusted twice per block to match the original high-frequency spectrum based on the decoded values $E'(i)$ of the transmitted energy values $E(i)$. The baseband signal is then added to the high-frequency generated signal to obtain a pseudo-excitation signal.

After decoding, the PARCOR coefficients are interpolated in the same way as at the analyzer. They are then used to control the vocal tract model filter. This filter is excited by the generated excitation signal, and its output gives the synthesized speech samples which are mu-law encoded in 8 bits and sent to the D/A converter which produces the output speech signal.

It must be noted that both the PARCORs and the high-frequency energies must be delayed to compensate for the delays which have been introduced in the baseband signal by the various filters. These compensating delays have been introduced partly during analysis and partly during synthesis in order to ensure that each transmitted frame contains information relative to the same analyzed block of speech samples.

- *Bit allocation*

The 7.2-kbps bit rate corresponds to 144 bits available for each 20-ms block. These bits are allocated to the different transmitted information according to the partitioning shown in **Table 1**.

Table 1 VEPC bit allocation (20-ms blocks).

Transmitted information		*Bits*
Spectral information		
PARCOR coefficients (8)	$K(i)$	28
Baseband information		
Subband characteristics (6 × 4 bits)	$c(i)$	24
Subband samples (5 × 15 bits)	$S(i, j)$	75
Scale factor	SF	1
High-frequency information		
High-frequency energy (2 × 4 bits)	$E(i)$	8
Programmable gain amplifier gain	G_1	3
Spare (synchro, signaling, etc.)		5
Total		144

Bit rate = 144/20 = 7.2 kbps.

Additional signal processing functions

In this section, we describe the application of our coder in a telephony environment. **Figure 4** shows that the insertion of a digital speech coder in an A-to-B full-duplex telephone conversation on two wires first requires a two-to-four-wire (2W/4W) conversion which is supposed to be implemented at the PBX level. We subsequently discuss the problem of controlling the echo introduced by this conversion, as well as some other functions, such as voice activity detection, automatic gain control, and error recovery, which are required to achieve an efficient digital transmission of the speech signal in full duplex. **Figure 5** shows a detailed

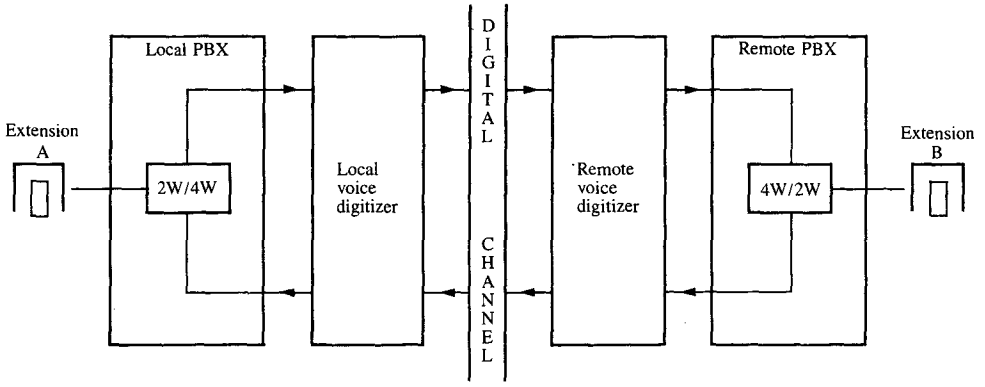

Figure 4 Insertion of a speech digitizer in a telephone environment.

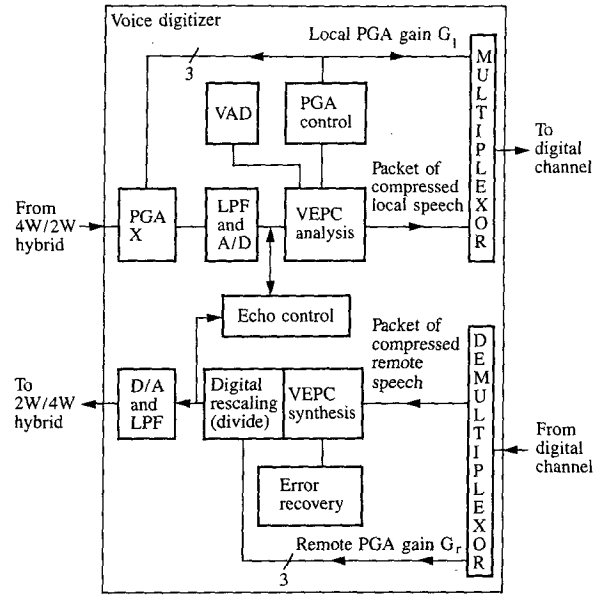

Figure 5

Detailed functional block diagram of a two-way speech digitizer operating in a telephone environment, as shown in Fig. 4.

functional block diagram of a VEPC-based speech digitizer as inserted in a telephony environment; it is discussed in the following section.

• *Voice activity detection*

As shown later, the VEPC algorithm can achieve the compression of the original 64-kbps speech signal to 7.2 kbps with communications quality. However, additional gain in the transmission can be achieved by taking advantage of the half-duplex effect of any normal telephone conversation, where each subscriber speaks less than half of the time. The remainder of the time is composed of listening, gaps between words and syllables, and pauses. This idle time can be used to interpolate additional talkers to achieve a limit of twice the overall channel capacity. To approach this limit, however, a considerable number of channels must be available.

One of the problems of a digital speech interpolation (DSI) system is the detection of the voice on a given channel. In general, the method used is based on the measurement of the signal energy for short periods of time. If this energy is lower than a prespecified threshold, the period is declared to be a silence, while speech is detected in the other case. The difficulty is to specify the threshold. If it is too high, the speech talkspurts are clipped, resulting in poor-quality speech; if it is too low, the activity of the circuit rapidly increases due to the idle noise, and the overall efficiency of the DSI system decreases.

In our speech detector, the activity decision is naturally based on an evaluation, for each block of input speech samples, of the signal energy, and on the comparison of this energy with an activity threshold. However, so as to take into account the long-term nonstationary quality of the background noise in any normal environment, the power spectrum of this noise is continuously evaluated and compared to the power spectrum of the signal currently being analyzed. Thus, the algorithm is designed to eliminate quasi-stationary noise. Another interesting feature of the algorithm is that it does not require a significant processing load when it is associated with speech coders based on linear prediction. Indeed, the short-term power spectrum of the signal in a block of samples can be determined from the autocorrelation function of the signal, and the energy of the signal is well approximated by the maximum magnitude of the samples within the block. These values are determined elsewhere for the encoding task, the first one for the computation of the predictor coefficients, the second one for intermediate signal scaling in our fixed-point implementation.

This algorithm has been found to provide robust detection of speech, even for low-energy fricatives in a high-level background noise.

• *Automatic gain control*

In a digital voice network, a critical problem appears at the analog-to-digital (A/D) conversion stage. Indeed, the signal level at the coder input may vary widely, depending on the type of telephone instrument and on the telephone line lengths. These amplitude variations create problems for A/D conversion, since in general the signal level does not fit the aperture of the mu-law A/D converter. For low-level signals, the ratio of signal to quantization noise is very low [3], while for high-level signals the A/D converter may be in overflow, thus producing harmonics by clipping.

Our experiments showed that these effects are amplified when a low-to-medium-bit-rate voice compression algorithm is used after A/D conversion. In such cases, both granular noise and clipping noise introduce a bias on the spectral analysis of the input signal, resulting in degradation of the decoded speech. To prevent such degradation, an A/D converter with a wider dynamic range can be used, e.g., a 12-bit linear A/D converter. Unfortunately, currently available linear converters are much more expensive and larger than mu-law coders. Therefore, we use a programmable gain amplifier (PGA) which is hard-wired in front of the A/D converter, as shown in Fig. 5. The VEPC analysis comprises a PGA control function in which the input signal level is tracked so as to derive a PGA gain which is further quantized with 3 bits in 6-dB steps. This 3-bit gain G_1 is locally used to control the front-end amplifier. It is also multiplexed in the transmitted frame, then used at the receiver end to digitally rescale the synthesized signal. Note

that this rescaling is necessary to avoid oscillation due to a loop gain in the 4W/2W conversions (see Fig. 4).

- *Echo control*

In most cases, a phone conversation is transmitted partly over a bidirectional two-wire line and partly over a couple of unidirectional two-wire lines. The junctions between the two types of lines are established by hybrid transformers (Fig. 4). Since the hybrid transformers are balanced for an averaged line impedance, they do not perfectly separate the two unidirectional paths from each other, and therefore they create echo signals. For local calls, these echoes do not disturb the conversation. However, if a delay is introduced in the four-wire path, for example by a satellite transmission or by digital encoding, the echoes must be reduced to ensure better speech quality.

Usually echoes are canceled in the following way: The signal of one of the unidirectional paths is analyzed in order to produce a replica of the echo signal; the replica is then subtracted from the unprocessed signal on the other unidirectional path. As a general rule, the greater the distance traveled by the echo, the longer should be the duration of the signal to be analyzed. Finite-impulse-response (FIR) digital filters are often used to approximate the hybrid transfer function. The filter coefficients are adapted by using a classical gradient method to determine the cross-correlation between the outgoing signal and the incoming signal after echo cancellation. Since typical echo paths can be as long as 32 ms, the filter delay line should have 256 taps, assuming an 8-kHz sampling rate. Considerable processing power could therefore be necessary to implement the echo canceler.

However, we have found that the echo path can be approximated by a flat delay followed by a short impulse response. Once the flat delay has been determined, the hybrid impulse response can be approximated with a low-order (16-to-32-tap) FIR filter, which drastically reduces the processing load required for the echo cancellation. The approximation is efficient; our experiments have shown that the relative level of the echo with respect to that of the signal is reduced to −40 dB in the case of hybrids having low echo return loss (−9 to −24 dB), and to −35 dB in the case of hybrids with high echo return loss (0 to −9 dB). For accurate determination of the flat delay, we use a fast data-transmission-type start-up procedure which is based on a training sequence [6]. As a result, the flat delay is evaluated and the echo canceler filter coefficients are initialized at the beginning of the conversation in less than 200 ms. Then, to prevent sudden variations of the hybrid balancing in the course of the conversation, the coefficients are continuously updated by using the standard gradient method.

- *Error recovery*

In a digital speech transmission system, the compressed digital voice signals are transmitted in the form of packets of bits. During transmission the voice packet may get lost. For example, it may be frozen-out in a network node due to a queueing overload. Transmission errors may also introduce bit errors within the packet. Due to the real-time constraint of speech signals, standard error recovery procedures involving parity checking and retransmission cannot apply, and in both cases the packet is considered lost. However, one can take advantage of the native redundancy of the speech signal to approximate the lost packet on the basis of knowledge of the previously received packets.

Due to the jitter in the delay of transmission (queues in communication controllers), the received speech packets are enqueued in a retention buffer to be used for synthesis at a constant 20-ms rate. When the synthesis requires a packet (N) which has not been received, the next packet ($N + 1$) may or may not have been received. In the first case (more than 95% of cases in practice), one can interpolate the missing packet, and in the second case one can only extrapolate it. The basic operation of our error recovery algorithm is as follows:

- To extrapolate packet (N) from packet ($N - 1$), both high-frequency energy $E(i)$ and PARCOR coefficients $K(i)$ are duplicated, and subband samples $S(i, j)$ are generated by use of a second-order generator. This is possible because in the case of voiced sounds each subband signal cannot contain more than one pitch harmonic and therefore looks like a sine wave signal; for unvoiced sounds, which do not contain harmonics, the low-frequency energy (in fact, the energy in each subband) is very small, and it does not require, as far as the speech quality is concerned, an accurate reconstruction of subband samples.
- To interpolate packet (N) between packets ($N - 1$) and ($N + 1$), the PARCOR coefficients are linearly interpolated, and the high-frequency energy values are geometrically interpolated. Again, subbands are extrapolated with a second-order generator.

This strategy has been found to be very efficient, since it ensures good speech intelligibility and speaker recognition for loss probabilities as high as 33%, and inaudible degradations of the speech for loss probabilities up to 5%. The algorithm details, as well as speech quality evaluation tests, are reported in [7].

Speech quality evaluation

Speech quality implies a human subjective perception of the degradation introduced in the coded speech; therefore, it cannot be evaluated accurately. Quality classification terms such as "toll," "communications," or "synthetic" are commonly used [3, 8]. In the following, the term *toll quality* refers to the quality obtained through a single telephone link equivalent to a 64-kbps CCITT PCM channel. At present, this quality can be achieved with coders operating at rates as

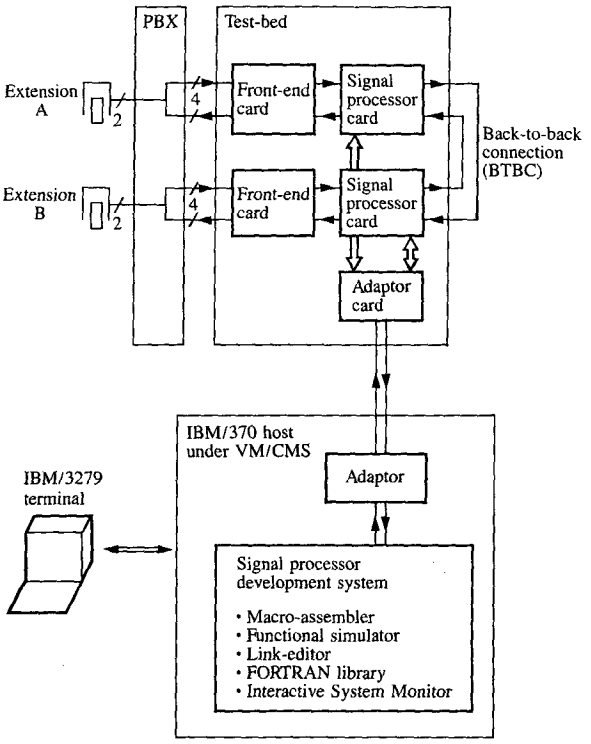

Figure 6

Real-time conversational test, with signal processor test-bed and development system.

low as 20 kbps. The term *communications quality* refers to the quality obtained through an analog/digital telephone link where degradations such as background noise and clipping are audible with a subjective signal-to-noise-ratio level in the range 20–30 dB (for instance, 32-kbps Adaptive Delta-Modulation); such quality can be achieved with efficient coders operating in the medium-bit-rate range of 20 to 5 kbps. Low bit rates (less than 5 kbps) can be achieved by vocoders which provide a "synthetic" quality, with intelligibility that is talker-dependent and with substantial loss of naturalness.

As shown below, the 7.2-kbps VEPC algorithm provides communications quality. Both the intelligibility and the pleasantness of the 7.2-kbps coded speech have been evaluated as follows.

- *Intelligibility*

Intelligibility was evaluated by using the standard diagnostic rhyme test (DRT) [9]. A first set of tests was conducted with DRT lists pronounced by three male and female speakers assuming high-fidelity recording conditions. These lists were listened to by eight individuals, resulting in an overall intelligibility score of 91%. Both speakers and listeners were native Americans. For reference purposes, the intelligibility score of currently available LPC vocoders operating at 2.4 kbps, as reported in [10], is 84%. Furthermore, the intelligibility scores of currently available CVSD adaptive delta modulation codecs operating at 32 and 16 kbps are, respectively, 95% and 91%.

A second set of tests, using French versions of the DRT lists, was conducted with ten male and female French speakers and more than 100 French listeners; it resulted in an average intelligibility score of 95% for high-fidelity input signals and 91% for carbon microphone input signals.

- *Pleasantness*

Pleasantness was evaluated in two ways.

- The standard diagnostic acceptability measure (DAM) [9] was performed on high-fidelity speech samples, assuming six male and female American speakers and fourteen listeners; it resulted in a composite acceptability estimate of 54%, which is equivalent to a predicted user acceptance of 86%.
- Several conversational tests were also conducted. In each test, two persons talked through a tie-line connected to a speech digitizer which operated with a back-to-back connection (BTBC), as shown in **Figure 6**. Note that in the test most of the potential impairments of a real network still exist or can be simulated. The echo impairment is due to the 2W/4W connections which are located at the PBX level. Similarly, the level of the input signal does not depend on the back-to-back connection and therefore can take all the values it would take in a real network. For the same reason, the background noise conditions which would normally occur in a real network are well reproduced in our tests. In addition, to take into account the delay existing in a real network, we introduced in the processors a 200-ms artificial delay in the back-to-back transmission of the voice packets. Similarly, one can randomly lose the transmitted packets at any rate.

Each test consisted in having a normal business conversation lasting about fifteen minutes. At the end of the test, each subscriber was asked to fill out a form on which he rated the various coder attributes according to a five-level scale: 5, excellent; 4, good; 3, fair; 2, poor; and 1, unsatisfactory.

The coder attributes rated with this scale were voice naturalness, speaker recognition, intelligibility, chopped-voice effect and noticeable residual echo (both due to any imperfect echo control), and the effort required to communicate. The obtained partial rates were then combined to provide an overall acceptability measure. Up to now, more than 200 people from the United States, England, Germany, and France have participated in the conversational tests, assuming various conditions (line

length, microphone type, input attenuation, etc.). For reference, **Table 2** shows the overall acceptability mean opinion score (MOS) obtained in two specific tests which were conducted assuming commonly encountered conditions, i.e., a 6-mile line between each of the customers and the PBX (Fig. 6), and hybrid transformers having an echo return loss of −12 dB. Both dynamic and carbon microphones have been used in the tests. The results of these particular tests, conducted with 40 people, show a MOS of 4.2 and a percentage of 100% of fair and better ratings for the dynamic microphone test, and a MOS of 3.1 and a percentage of 80% of fair and better ratings for the carbon microphone test. These figures permit assessment of the degradation introduced on speech pleasantness by a carbon microphone.

Signal processor

The signal processor under consideration was developed as a custom chip [1], using the Essonnes macro-stack applied on bipolar technology with 2.5-μm ground rules. The design used a large bus with a few random logic gates, resulting in a very dense (200 circuits/mm^2) and very fast (100-ns instruction time) prototype chip. Note that due to timing difficulties with currently available memories, a slower cycle time (130 to 150 ns) is used in our implementation. The processor is equivalent to 5000 NAND gates and is housed in a 25-mm^2 chip. The chip is manufactured and tested on a standard line.

- *Architecture*

The architecture of this signal processor was derived [1] from an original proposal by the Zurich Research Laboratory [11]. The original version of this architecture offered the following features:

- The use of two accumulators and of a 16-bit ALU in parallel with a 12 × 12-bit multiplier. One level of pipeline on the multiplier permits completion of a N-tap digital filtering task within $(N + 2)$ cycles. In practical cases, the computational complexity of such a filter is thus reduced to one multiplication per cycle.
- Two levels of pipeline—instruction fetching and execution are performed in the same clock cycle.
- Two levels of parallelism for processor functions—data transfers and arithmetic/logic operations are performed simultaneously and independently of one another.

The characteristics of our signal processor differ slightly from those of the original version [11]; they are summarized as follows with reference to the simplified block diagram shown in **Figure 7**:

- 16-bit ALU, with 20-bit accumulation on two accumulators.

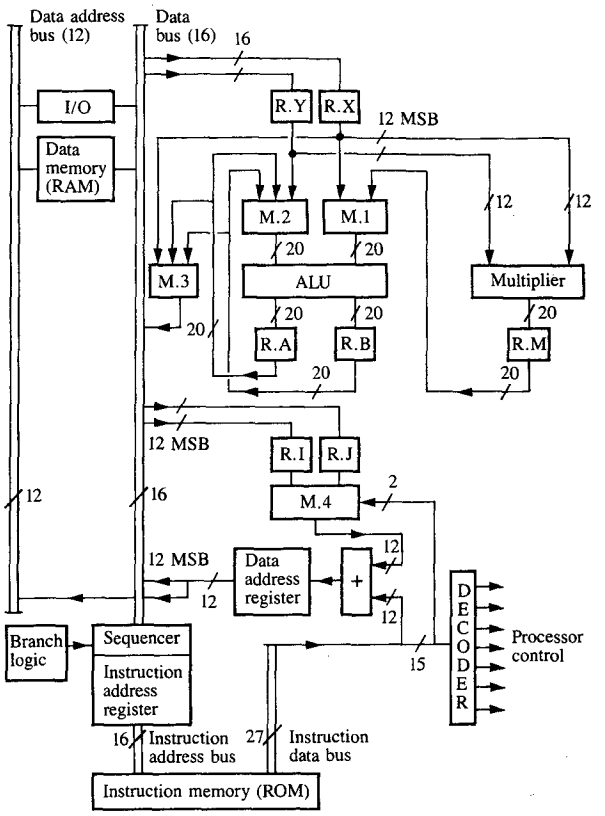

Figure 7

Signal processor simplified block diagram (R.X, registers; M.X, multiplexors).

Table 2 Results of two conversational tests realized with a 6-mile line before the A/D converter and with hybrids having an echo return loss of −12 dB. Both dynamic and carbon microphones have been incorporated in these tests. Number of subjects in each test: 40.

Test condition	*Overall MOS and standard deviation σ*	*Percentage of good or better*	*Percentage of fair or better*
Dynamic microphone	4.2 (0.6)	92	100
Carbon microphone	3.1 (0.7)	20	80

- On-chip 12 × 12-bit multiplier.
- Bidirectional 16-bit data bus.
- Nine internal working registers.
- 4K halfwords of data (RAM) addressability (extendible via paging through an I/O register), and 64K words of instruction addressability.
- 27-bit instruction word with two independent operation codes, and on-chip instruction parity checking.

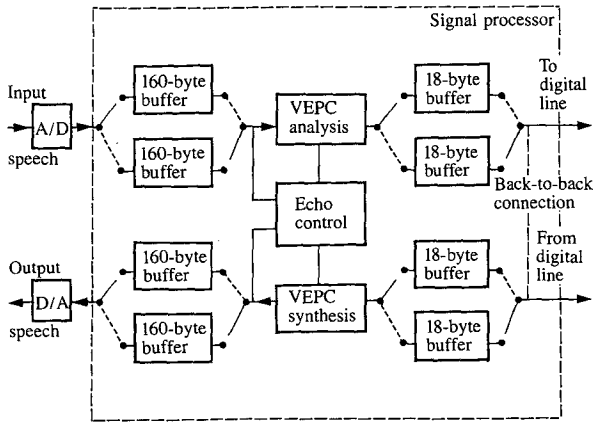

Figure 8

Real-time VEPC analysis/synthesis.

Table 3 VEPC computational complexity.

Function	*Instructions* (27-bit words)	*Data* (16-bit words)	*Load* (ms per 20-ms block)
Analyzer			
Linear prediction analysis	518	68	3.18
Step-up procedure and coding	147	24	0.08
Inverse filtering	45	14	0.86
Baseband downsampling/HFE	185	21	2.46
Baseband coding	677	60	0.78
Multiplexing	144	29	0.01
Echo canceling	730	430	1.32
Voice activity detecting	130	10	0.00
Automatic gain control	150	20	0.00
Synthesizer			
Demultiplexing	171	20	0.01
Baseband decoding	487	24	0.39
Baseband upsampling	102	5	0.52
Upper-band generating	288	51	3.73
Step-up procedure and decoding	137	25	0.09
Linear prediction filtering	45	5	0.86
Post-emphasis	47	32	0.30
Error recovery	290	170	0.06
Service subroutines			
DIV, SQRT, NORM, etc.	436	65	—
Housekeeping subroutines			
I/O, 8 → 12 bits, 12 → 8 bits, etc.	686	1707	1.90
Total	5415	2790	16.55

- Direct and indirect working memory access (two index registers).
- One interrupt level.

- *Development tools*

The signal processor is implemented on a 7 × 4.5-in. (17.78 × 11.43-cm) card, including RAM storage for 8K 27-bit instructions and 4K halfwords of data, clock circuits, and random logic. In the prototype operating card, the processor has a cycle time of 150 ns. The analog interface, including the A/D and D/A converters, the low-pass filters, the PGA, and the signaling circuits, is implemented on another card (front-end card). The speech coder test-bed (Fig. 6) includes two signal processor cards and two front-end cards, as well as an adaptor card which interfaces the development system being operated under VM/CMS. The La Gaude I/S Development and System Test Department has developed and adapted under VM powerful tools which include a macro-assembler, a functional simulator, a link-editor, and the Interactive System Monitor (ISM) for the monitoring of both simulator and NCU/NCUA attachment.

In addition, we have developed a software interface between the functional signal processor simulator and the IBM/370 instruction set. This tool permits merging of signal processor microcode and either FORTRAN or IBM/370 ASSEMBLER subroutines; it has drastically simplified the implementation burden.

VEPC implementation

- *Microcode structure*

The VEPC microcode has been structured in separate control sections corresponding to basic building blocks such as digital filters (signal processing building blocks) or division routines (service building blocks). In addition, the macro-assembler facility available with the signal processor assembler has been extensively used to provide more flexibility and clarity during the debugging phase. A/D and D/A convertors are connected to two of the signal processor's I/Os, as shown in **Figure 8**. At each 8-kHz conversion, the processor is interrupted, and the instruction address register is forced to address 2, where an interrupt program is executed. The input sample available on the A/D I/O is stored in a 160-sample (20-ms) buffer. Similarly, an output sample is taken from a 160-sample buffer and held on the D/A I/O. Then the buffer pointers are incremented by one and tested. If the 160 data locations have been written/read, the pointers are appropriately swapped with the analysis pointer and the synthesis pointer. Simultaneously, the I/O pointers (in the buffers which contain the compressed speech and are connected to the digital line) are swapped. Then the analysis and synthesis programs are activated in sequence. The analysis pointer gives the data address at which the input block of samples to be processed is stored. The synthesis pointer gives the data address at which the output block of samples will be stored after synthesis processing. After analysis/synthesis, the program enters a wait loop and remains there until the next pointer swapping.

• *Computational complexity*

Table 3 gives the computational complexity of each signal processing function in terms of program size in number of instructions, number of required data halfwords, and execution time per 20-ms analysis block, assuming a 150-ns cycle time. It can be seen from these figures that the current implementation of the VEPC analysis/synthesis requires about 16 ms for each block of 20 ms. This processing load corresponds to 5.3 MIPS (millions of instructions per second).

Conclusion

In this paper, we have discussed the implementation of a medium-bit-rate speech coder on a new IBM signal processor prototype with on-chip multiplier, which has a processing capacity of 10 MIPS. However, due to the speed limitation of currently available memories, the processor cycle was slowed down to 150 ns in our implementation.

We have also discussed the application of our coder in a normal telephone environment; this requires mu-law to linear PCM conversion and other signal processing functions such as voice activity detection, automatic gain control, echo control, and error recovery.

Speech quality evaluation tests show that the current VEPC coder, operating at 7.2 kbps, permits digital transmission of communications-quality speech while ensuring intelligibility sufficient for most telephony applications and good speaker recognition. Since its implementation at present requires only 5 MIPS, with a program size of 5K instructions, this coder is a good candidate for real-time applications of speech compression such as digital trunk transmissions, satellite speech communications, secure voice communications, mobile radio, and audio distribution systems.

References

1. Jean Paul Beraud, "Signal Processor Chip Implementation," *IBM J. Res. Develop.* **29**, 140–146 (March 1985, this issue).
2. D. Esteban, C. Galand, D. Mauduit, and J. Menez, "9.6/7.2 kbps Voice Excited Predictive Coder (VEPC)," *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing,* Tulsa, OK, 1978, pp. 307–311.
3. C. Galand, "Sub-Band Coding: Theory and Application to Digital Coding of Speech," Ph.D. Thesis, Nice University, France, 1983.
4. J. D. Markel and A. H. Gray, *Linear Prediction of Speech,* Springer-Verlag, Berlin, 1976.
5. J. Leroux and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients," *IEEE Trans. Acoust., Speech, & Signal Proc.* **ASSP-25**, 257–259 (June 1977).
6. A. Milewski, "Periodic Sequences with Optimal Properties for Channel Estimation and Fast Start-Up Equalization," *IBM J. Res. Develop.* **27**, 426–431 (September 1983).
7. G. Platel, C. Galand, and R. Vermot-Gauchy, "Speech Packet Recovery in a Digital Voice Network," *Proceedings, GLOBECOM Conference,* Atlanta, GA, November 1984, pp. 1330–1334.
8. J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant, and J. M. Tribolet, "Speech Coding," *IEEE Trans. Commun.* **COM-27**, 710–737 (April 1979).
9. W. D. Voiers, "Diagnostic Acceptability Measure for Speech Communication Systems," *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing,* Hartford, CT, May 1977, pp. 204–207.
10. G. S. Kang and S. S. Everett, "Improvement of the Narrowband LPC Analysis," *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing,* Boston, MA, 1983, pp. 89–92.
11. G. Ungerboeck, D. Maiwald, H. P. Kaeser, and P. R. Chevillat, "The SP16 Signal Processor," *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing,* San Diego, CA, 1984, Paper 16.2.

Received March 15, 1984; revised September 27, 1984

Chantal Couturier *IBM France, Centre d'Etudes et Recherches, 06610 La Gaude, France.* Ms. Couturier graduated in 1982 from the Ecole Nationale Supérieure des Telecommunications in Paris, France. She joined IBM in 1982 and since that time has been involved in software development in digital speech processing.

Claude Galand *IBM France, Centre d'Etudes et Recherches, 06610 La Gaude, France.* Dr. Galand graduated from Nice University, receiving the doctorate in electronic engineering in 1974 and the state doctorate of sciences in 1983. From 1972 to 1976, he was an assistant professor with the Department of Electronic Engineering of Nice University, where he taught mathematics and electronics. Since joining IBM at the La Gaude laboratory in 1976, Dr. Galand has conducted research in speech analysis, synthesis and coding, optimal quantization, digital filter design, multirate digital filtering, voice activity detection, echo cancellation, and algorithm implementations on digital signal processors. He has received five IBM Invention Achievement Awards, one IBM Outstanding Technical Achievement Award, one IBM Outstanding Innovation Award for speech processing algorithms, and the 1982 Technical Vitality Award from IBM France.

Guy Platel *IBM France, Centre d'Etudes et Recherches, 06610 La Gaude, France.* Mr. Platel graduated in 1982 from the Ecole Nationale Supérieure des Telecommunications in Paris, France. He joined IBM in 1982 and since that time has been involved in software development in digital speech processing.

Robert Vermot-Gauchy *IBM France, Centre d'Etudes et Recherches, 06610 La Gaude, France.* Mr. Vermot-Gauchy graduated from Paris University, where he received the Master of Sciences degree in 1955. He received a degree in electronic engineering in 1957 from the Ecole Supérieure d'Electricité, Institute of Technology. From 1957 to 1959, his main activity was as lecturer in mathematics and physics (electricity and quantum mechanics). He joined IBM in 1960 as an engineer in programming. Starting in 1962 as a project manager at the IBM France laboratory in La Gaude, he has been responsible for the following activities: the Special Design Automation program, programming education, simulation in optical character readers and in signal processing (audio and modem area), the European Administrative Information System, and communication line protocol in message switching. Mr. Vermot-Gauchy is now working in speech analysis/synthesis and coding and is responsible for microprocessor simulation programs.