Fixed Point Implementation of the QCELP Speech Coder

Byung Sik Yoon, Jae Won Kim, Won Myoung Lee, Seok Jin Jang,
Song In Choi, and Myoung Seob Lim

CONTENTS

ABSTRACT

The Qualcomm code excited linear prediction (QCELP) speech coder was adopted to increase the capacity of the CDMA Mobile System (CMS). In this paper, we implemented the QCELP speech coding algorithm by using TMS320C50 fixed point DSP chip. Also the fixed point simulation was done with C language. The computation complexity of QCELP on TMS320C50 was about 33 MIPS and program size was 10k words and data memory was 4k words. In the normal call test on the CMS, where mobile to mobile call test was done in the bypass mode without double vocoding, mean opinion score for the speech quality was 3.11.

I. INTRODUCTION

Due to the rapid growth of the cellular industry in worldwide, it has been noted the next generation cellular technology provides a 10-fold increase in the capacity, the widened coverage area and the quality improvement over the current analog cellular system [1]. CDMA Mobile System (CMS) require an efficient speech coding method to achieve capacity improvements for accommodating the rapid increase of the cellular subscribers in Korea. The Qualcomn code excited linear prediction (QCELP) speech coder can reduce the average data rate using a voice activity factor. This reduction leads to a 2-fold increase in the capacity of CDMA based digital cellular telephone system.

The QCELP speech coder [6] dynamically selects one of four data rates every 20 ms, depending on the voice activity. The four data rates are 8 kbps ("full rate"), 4 kbps ("half rate"), 2 kbps ("quarter rate"), or 1 kbps ("eighth rate"). Typically, active speech is coded at the 8 kbps rate, while silence and background noise are coded at the lower rates.

The QCELP coder is based on the code excited linear prediction (CELP) which is analysis-by-synthesis algorithm [2]. The basic structure of QCELP coder minimizes complexity by the integrated implementation of all four data rates. At higher rates, the linear predictive coding (LPC) parameters are more finely quantized and the pitch and codebook parameters are updated more frequently.

This paper describes the algorithm and real-time implementation of the 8 kbps QCELP speech coder which has been selected for the CMS in Korea.

II. THE QCELP SPEECH CODER

A speech waveform is produced by three actions: sound source generation, articulation by sound propagation through the vocal tract, and sound radiation from lips and nostrils.

A voiced source is simulated by a train of periodic pulses or triangular waves whose period and amplitude correspond to the pitch and intensity of a voice source, unvoiced source is simulated by white random noise whose average power corresponds to average turbulence energy. Articulation is simulated by the connection of single-resonance circuits in series or in parallel. Articulation by vocal-tract shape is simulated by all-pole or all-zero filters. An all-pole system is identified by linear predictive analysis using the AR process : the poles correspond to formants (resonances) of the speech spectrum. TIA/EIA IS-96, which is based upon the general CELP structure, was adopted in our CDMA digital cellular system as the speech coder. This QCELP encoding algorithm operates at rates of 8, 4, 2, or 1 kbps, depending on the level of voice activity. As shown in the Fig.1, the

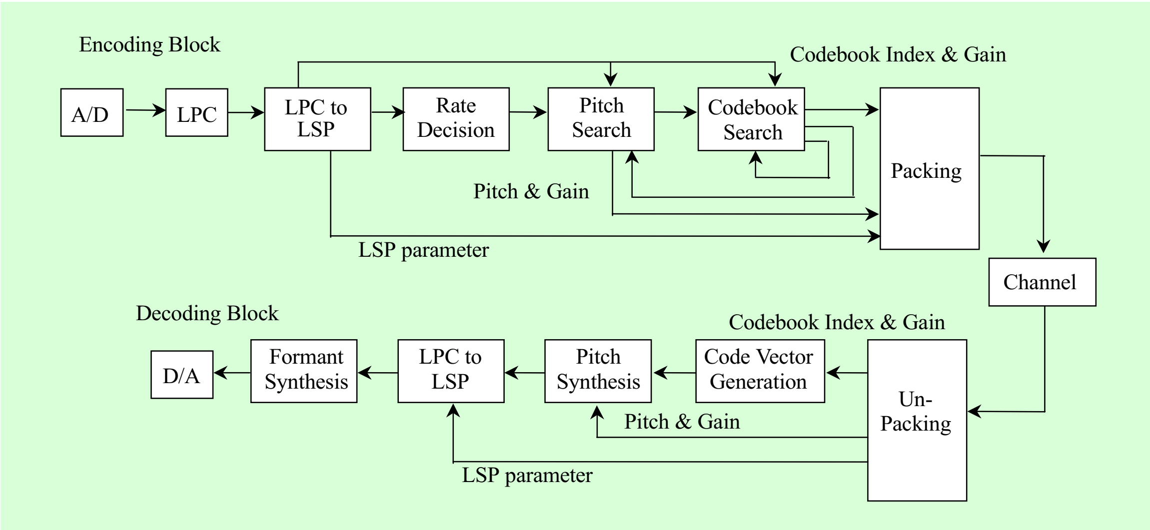

Fig. 1. TIA/EIA IS-96 speech coder architecture.

input speech is sampled at 8 kHz and is broken down 20 ms speech frames consisting of 160 samples. The LPC with 10th order are calculated regardless of data rate selected. The LPC parameters are transformed into line spectrum pairs (LSP) frequency [5] for efficient quantization, interpolation, and easily stability check. The LSPs are determined in each subframe through interpolation of LSPs in neighbor frames. A weighting filter is used to reduce the loudness of the quantization noise.

In the pitch search, pitch gain and pitch lag are determined by the error minimization procedure known as the standard analysis-by-synthesis method. The pitch lag is quantized from 17 to 143 samples using 7 bits for each pitch update. Pitch gain is scalar-quantized with the range from 0 to 2 using 3 bits per pitch parameter update. To determine the optimal pitch gain and pitch lag, the global search is performed over all allowable quantized values. The codebook gain and codebook index are determined once for each codebook update. As in the pitch search procedure, the codebook index and the codebook gain are chosen using the analysis-by-synthesis procedure. The codebook is organized as an overlapping codebook such that each code vector differs from the adjacent code vector by one sample. The circular codebook consists of the 128 elements including 101 zero elements and 27 nonzero elements. The use of overlapping codebook provides several advantages so that the storage requirements for codebooks of various rates can be reduced significantly and fast codebook search can be applied by using the dependency of the neighbor candidate code vector. Moreover, the extra complexity reduc-

Table 1. Parameters used for each rate.

Parameter	rate 1	rate 1/2	rate 1/4	rate 1/8
Linear predictive coding (LPC) updates per frame	1	1	1	1
Samples per LPC update, L_A	160 (20 ms)	160 (20 ms)	160 (20 ms)	160 (20 ms)
Bits per LPC update	40	20	10	10
Pitch updates (subframes) per frame	4	2	1	0
Samples per pitch subframe, L_p	40 (5 ms)	80 (10 ms)	160 (20 ms)	–
Bits per pitch update	10	10	10	–
Codebook updates (subframes) per frame	8	4	2	1
Samples per pitch subframe, L_C	20 (2.5 ms)	40 (5ms)	80 (10ms)	160 (20ms)
Bits per codebook update	10	10	10	6

tion without any performance degradation can be

Table 2. Bit allocation for a rate 1 packet.

LPC Frame	40							
Pitch Subframe	10		10		10		10	
Codebook Subframe	10	10	10	10	10	10	10	10

Total = 160 bits (plus 11 parity check bits)

achieved by using the recursive form in calculation of energy term in pitch parameter search and codebook parameter search. In lower transmission rates the complexity reduction by the recursive calculation of energy term is increased. The important parameters are described in Table 1 through

Table 3. Bit allocation for a rate 1/2 packet.

LPC Frame	20			
Pitch Subframe	10		10	
Codebook Subframe	10	10	10	10

Total = 80 bits

Table 4. Bit allocation for a rate 1/4 packet.

LPC Frame	10	
Pitch Subframe	10	
Codebook Subframe	10	10

Total = 40 bits

5. For example, the LPC parameters are updated once per frame, using 40 bits at full rate, 20 bits at half rate, and 10 bits for

Table 5. Bit allocation for a rate 1/8 packet.

LPC Frame	10	Total = 16 bits
Pitch Subframe	0	
Codebook Subframe	6	

quarter and eighth rates.

1. Determining the Formant Prediction Parameters

The formant synthesis filter is equivalent to the traditional LPC filter. The transfer function for the formant prediction error filter, which removes the short term redundancies in the speech, is

$$\frac{1}{A(z)} = \frac{1}{1-\sum_{i=1}^{p} a_i z^{-i}}. \tag{1}$$

The encoding process begins by determining the formant prediction parameters. This is performed by the following steps:

1) DC block is inserted to prevent a DC offset from artificially increasing R(0) and thus disrupting the rate decision algorithm.
2) The coefficients are computed from a Hamming window of speech centered at the center of the fourth rate 1 pitch subframe to interpolate LPC coefficient during pitch and codebook search. The window is 160 samples long. Following the windowing operation, the autocorrelation function is computed as only the first 11 values of the autocorrelation function R(0) through R(10) need to be computed from the windowed speech signal within the analysis window.
3) The LPC coefficients are obtained from the autocorrelation function by using the Durbin's recursion.
4) The LPC coefficients have bandwidth expansion applied before they are transformed into LSP frequencies. This is done by scaling the poles of the formant synthesis filter radially inwards.
5) The LPC coefficients are transformed into the LSP frequencies for efficient quantization, and easy stability check. The basic computation of the LSP frequencies follows. The polynomial $A(z)$, associated with pth order LPC analysis, satisfied following recurrence relationship,

$$A_{n+1}(z) = A_n(z) - k_{n+1} z^{-(n+1)} A_n(z^{-1}) \tag{2}$$
$$n = 1,\ 2,\ \cdots\cdots,\ p.$$

The parameter $\{k_i\}_{i=1,\ 2,\ \cdots p}$ are called the PARCOR coefficients.

There are two extreme artificial boundary conditions, namely, $k_{p+1} = 1$ and $k_{p+1} = -1$. These conditions correspond, respectively, to a complete closure and a complete opening at the glottis in the acoustic tube model. Under these conditions, the polynomial $A_{p+1}(z)$ coincides with the polynomials

$$P'(w) = \cos 5(2\pi w) + p'_1 \cos 4(2\pi w) + \cdots + p'_4 \cos(2\pi w) + p'_5/2, \tag{3}$$

$$Q'(w) = \cos 5(2\pi w) + q'_1 \cos 4(2\pi w) + \cdots + q'_4 \cos(2\pi w) + q'_5/2, \tag{4}$$

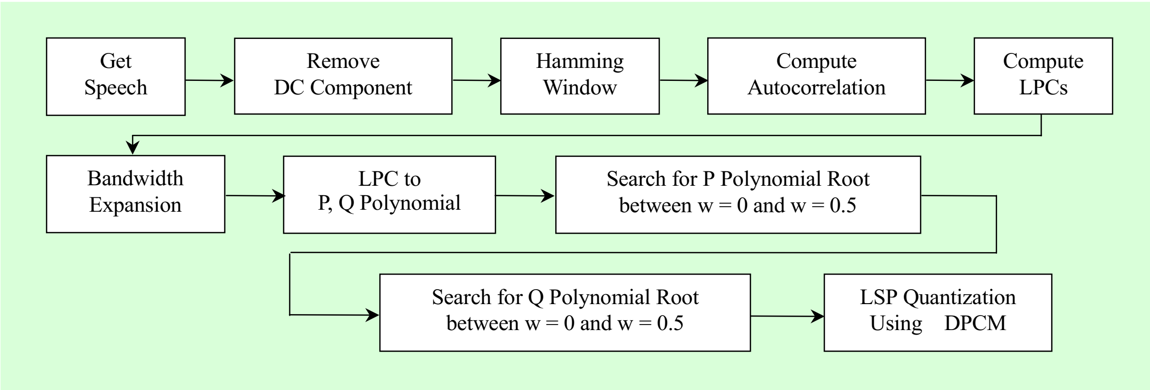

Fig. 2. Encoding flow from speech to LSP code.

where the p' and q' values are computed recursively as follows from the p and q values defined above.

$$p'_0 = q'_0 = 1 \quad (5)$$

$$p'_i = p_i - p_{i-1},\ p_i = -a_i - a_{11-i},\ 1 \le i \le 5 \quad (6)$$

$$q'_i = q_i - q_{i-1},\ q_i = -a_i - a_{11-i},\ 1 \le i \le 5 \quad (7)$$

The LSP frequencies are 10 roots which exist between $w = 0$ and $w = 0.5$ in these two equations. Since the formant synthesis filter is stable, the root of the two functions alternate: therefore, $w1$, $w3$, $w5$, $w7$ and $w9$ are the roots of $P'(w)$, and $w2$, $w4$, $w6$, $w8$ and $w10$ are the roots of $Q'(w)$.

6) Once the LSP frequencies have been computed and the data rate has been selected, each LSP frequency is converted for transmission. LSP frequency is quantized using a differential quantizer. The bias of each frequency is subtracted out from LSP frequencies, and the difference between the resulting value and a predicted value based on the previous frame is then scalar-quantized. Because each LSP frequency varies slowly with time. The LSP frequencies change more quickly in frames containing speech than in frames containing only silence and background noise, so quantizers with greater dynamic range are used in high rate frames. In addition, each frequency has a slightly different distribution of differences, so higher dynamic ranges are used on frequencies which tend to vary more.

The decoding of LPC parameters is performed by the following process:

1) The LSP transmission codes are decoded to the LSP frequencies at both the transmitting encoder and the receiving decoder.
2) Before converting the LSP frequencies back to LPC coefficients, a check is done to ensure that the resulting filter has not

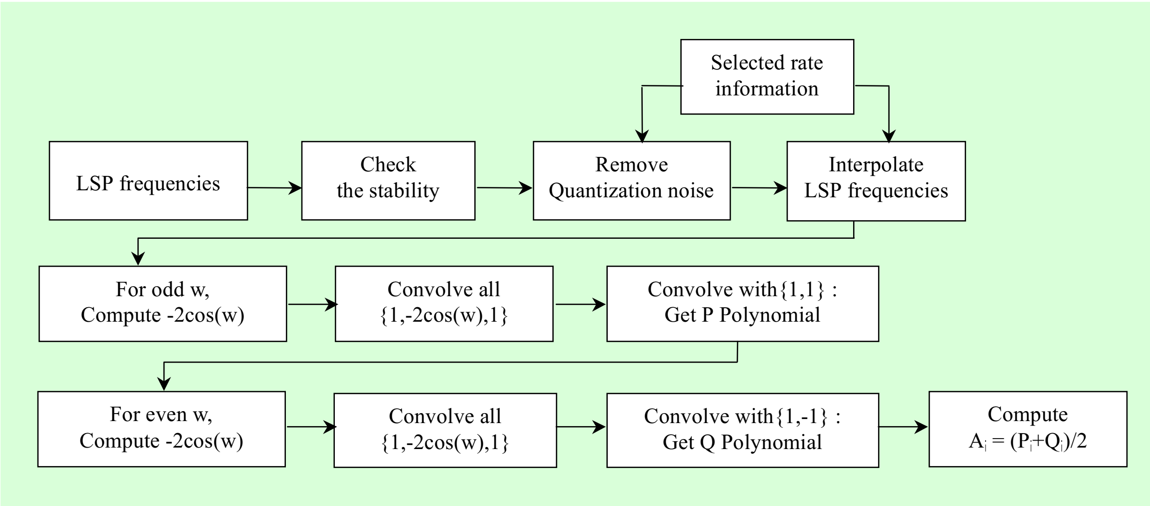

Fig. 3. Conversion flow of the LSP to the LPC parameters.

been made unstable due to the quantization noise or channel errors into one or many LSP frequencies. Stability is guaranteed if the LSP frequencies remain ordered. In addition, the LSP frequencies are forced to be at least 80 Hz apart to prevent unusually large peaks in the formant synthesis filter response.

3) The LSP frequencies are low-pass filtered to remove some of the quantization noise effects at lower rates.

4) The LSP frequencies are interpolated for each subframe of the pitch and codebook searches in the selected rate. The LPC coefficients are recomputed by the method of the interpolation between the LSP frequencies of the previous frame and the current frame. Interpolating in the LSP domain, rather than the LPC domain, has the advantage of ensuring the stability of the resulting filter as well as providing smooth speech transitions because the LSP frequencies vary slowly and smoothly over time, whereas the LPC coefficients vary frequently and unpredictively. Because of the adverse effect according to this interpolation in the LSP domain, the reverse transformation from the LSP to the LPC must be performed rather than just once for each frame.

5) The interpolated LSP frequencies are used for speech generation in the receiving decoder and also converted back into LPC coefficients for use in the pitch and codebook searches.

2. Determining the Pitch Prediction Parameters

In order to select the pitch parameters, an analysis-by-synthesis method, where encoding is done by selecting parameters which minimize the weighted error between the input speech and the synthesized speech, is used.

The pitch synthesis filter can be expressed as :

$$\frac{1}{P(z)} = \frac{1}{1-bz^{-L}}, \tag{8}$$

where L is the pitch lag which is selected from set $\{17, 18, \cdots, 143\}$ and b is the pitch gain which is selected between 0 and 2.0 in steps of 0.25.

The synthesis filter used in the speech encoder differs from that in the speech decoder. The synthesis for the speech encoder includes the same noise weighting parameter, ζ, in the weighting filter. The perceptual weighting filter has the form of

$$W(z) = \frac{A(z)}{A(z/\zeta)}, \tag{9}$$

where $A(z)$ is the formant prediction error filter and ζ, which is 0.8, is a perceptual weighting parameter. The perceptual weighting filter is used to match the weighting applied to the input speech. The perceptual weighting filter has a filter state associated with it at the start of each subframe. In order to remove the effects of the weighted synthesis filter initial state from the subframe parameter determinations, the zero input response (ZIR) of the weighted synthesis filter shall be computed and subtracted from the weighted input speech for the subframe.

The speech coder uses the closed loop approach to choose the pitch lag. In the closed loop case the lag is determined only from the past output of the long term filter and the current input speech. The closed loop pitch search is performed as the codebook is made up of the past reconstructed formant residual samples. These samples are the output of the pitch prediction filter up to the end of the previous pitch subframe. Some of the formant residual samples, $p(n)$ for $n \geq 0$, that are needed in the pitch search are not available since they fall in the current subframe. For large values of Lp, the pitch search can produce an unstable pitch prediction filter if these samples are replaced by the open loop formant residual. Therefore the pitch search is done over 143 closed loop formant residual samples $\{p(n)$ for $n \geq 0\}$ plus (Lp - 17) open loop formant residual samples$\{P_0(n)$ for $n \geq 0\}$. The search changes gradually from mostly open loop search to completely closed loop search over all possible values L and b.

The mean square error (MSE) can be represented by

$$\begin{aligned} MSE &= \frac{1}{Lp}\sum_{n=0}^{Lp-1}\{x(n)-x'(n)\}^2 \\ &= \frac{1}{Lp}\sum_{n=0}^{Lp-1}\{x(n)-by(n)\}^2, \end{aligned} \tag{10}$$

where $x(n) = x_p(n)$ for $0 \leq n \leq L_p - 1$ is the perceptually weighted (by W(z)) speech input with ZIR subtracted, and

$$Y(n) = h(n)^{*}p(n-L), \text{ for } 0 \leq n \leq L_p - 1 \tag{11}$$

is the weighted synthesized speech with pitch lag L when $b = 1$. It can be shown that this is equivalent to maximizing

$$E_L = \frac{(E_{xy})^2}{E_{yy}},$$

$$\text{where} \quad E_{xy} = \sum_{n=0}^{L_p-1} x(n)y(n), \quad \text{and}$$

$$E_{yy} = \sum_{n=0}^{L_p-1} y(n)y(n). \tag{12}$$

Hence the optimum b for the given L is found to be $b_L = E_{xy}/E_{yy}$. This search is repeated for all allowed values of L. The optimum b is restricted to be positive; E_L that produces a negative b, is ignored in the search. Finally the pitch lag L and the pitch gain b that maximizes E_L are chosen for transmission. Notice that x(n) needs to be computed only once but $y(n)$ needs to be computed for each lag L.

3. Determination of the Excitation Codebook Parameters

Codebook search is nearly identical to the pitch search. The index of the codebook I and the gain factor G are determined once for each codebook update. A gaussian, center-clipped, recursive circular codebook of length 128 is used [7]. As in the pitch search procedure, I and G are chosen using analysis-by-synthesis procedures. The differences are:

1) The optimum pitch lag L and pitch gain b found in the pitch search is used for the pitch synthesis filter $1/P(z)$ which is now in the search loop. For rates 1/4 and 1/8, the pitch gain b is set to zero.
2) For each codebook index, the codebook gain G is optimized.
3) The codebook index I and the codebook gain G that minimized the perceptually weighted mean squared error are selected for transmission.

Due to the high update rate of codebook parameters, there is significant correlation in the codebook gain which allow gains to be encoded differentially. The magnitude of codebook gain is coded using a single differential coder operating on the log of the magnitude of G. The differential coder employs a 2 bit linear quantizer similar to that used for LSP frequencies. The sign of G is transmitted using 1 bit. As in the pitch search procedure, the search is performed over all allowable quantization levels of I and G.

The coder structure is modified slightly for eighth rate frames to code background noise more efficiently. Because the pitch filter provides no improvement in background noise, the pitch gain is set to zero for all eighth rate frames. In addition, the codebook index I and the sign of the codebook gain are not transmitted and the codebook itself is replaced by a white noise generator. The seed for the generator is a function of the eighth rate packet of data, which is available at both the encoder and the decoder. This ensures that both the encoder and decoder produce the same random noise sequence, keeping them synchronized.

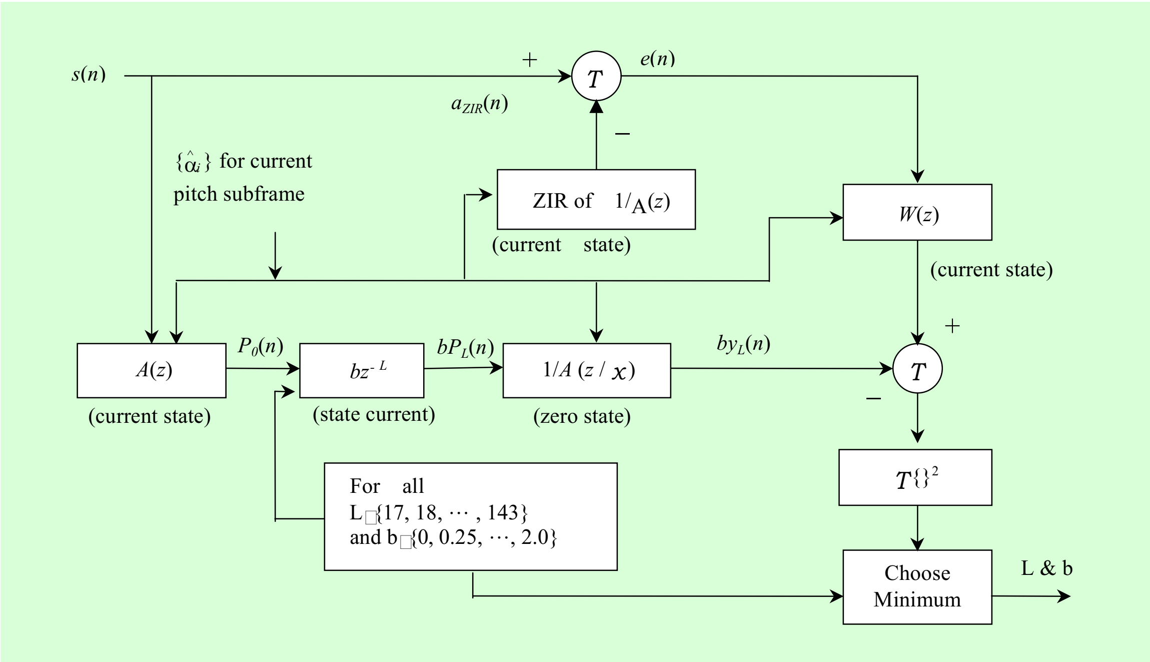

Fig. 4. Analysis by synthesis procedure for the pitch parameter search.

4. Rate Decision

QCELP uses an adaptive algorithm to determine the data rate for each frame. The algorithm keeps a running estimate of the background noise energy, and selects the data rate based on the difference between the background noise energy estimate and the current frame's energy.

In each frame, the previous estimate of the background noise energy is compared with the current frame's energy if the previous estimate is higher than the current frame's energy, then the estimate is replaced by that energy. Otherwise, the estimate is increased slightly. When no speech is present, the background noise estimate follows the input signal energy. During active speech, the estimate slowly increases, but fluctuations inherent in the energy of the speech signal cause it to be reset continually.

The data rate is then selected based on a set of thresholds which "float" above the background noise estimate. If the current frame's energy is above all three thresholds, the coder encodes the speech at full rate. If the energy is less than all three thresholds, the coder encodes the speech at eighth rate. If the energy is between the thresholds, the intermediate rates are chosen.

With this algorithm, background noise is almost always coded at eighth rate regardless of its energy. If the background noise suddenly increases, such as when a

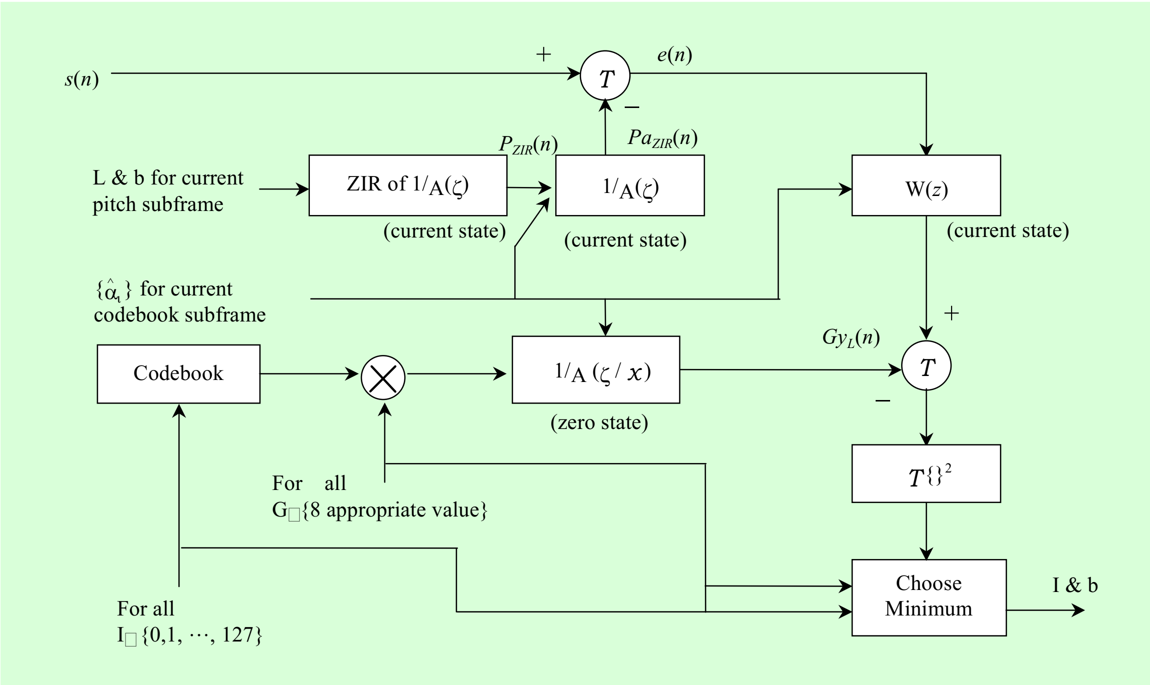

Fig. 5. Analysis by synthesis procedure for the codebook parameter search.

driver using a car phone opens his window, initially the background noise will be coded at the higher rates. After a few seconds the background noise estimate will rise to the new level of noise and the background will once again be coded at the eighth rate. If the background noise suddenly drops, the estimate immediately drops with it, preventing speech from being coded at the lower rates.

III. REAL-TIME IMPLEMENTATION

1. Fixed Point Simulation of the QCELP Speech Coder

We simulated the QCELP coder using the C language and evaluated its performance before implementing the QCELP algorithm on a fixed point DSP chip. We investigated each block to prevent overflows and underflows. The flow diagram of the QCELP encoder and decoder is shown in Fig. 6. Each routine in a function block uses different operations because the effect of the finite word-length variable varies in each function block. The dynamic ranges of all variables in each function block are examined to find the optimum word length of the variables [8], [9].

The double precision operations were used in some variables which have the wide dynamic ranges and degrade severely the overall performance when using a single

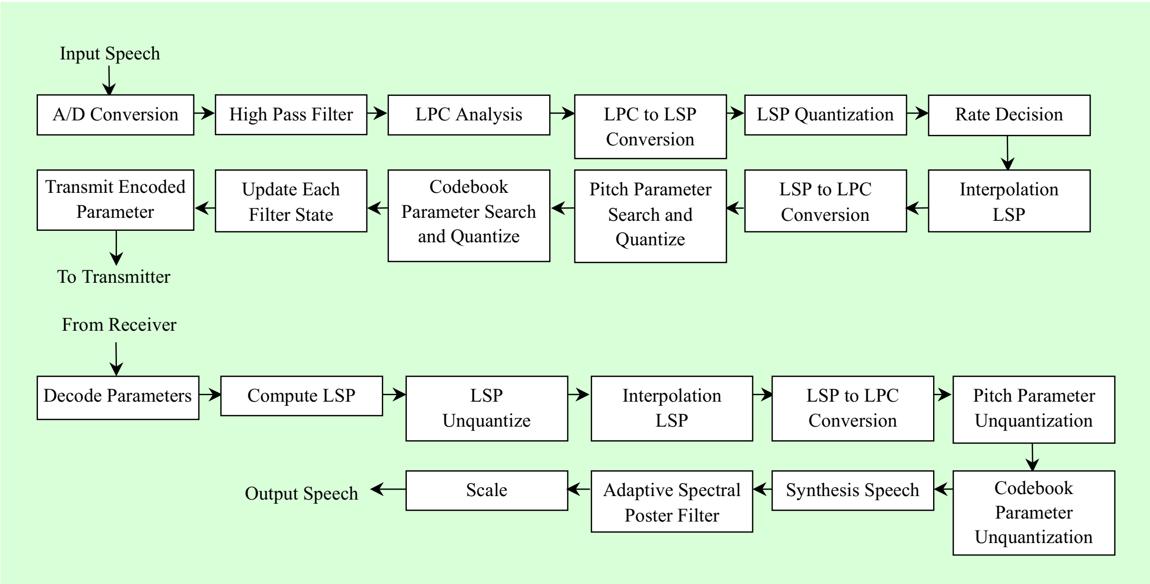

Fig. 6. Flow diagram of QCELP encoder and decoder.

precision operation. Instead of using a direct double precision multiplication, the variable scaling method is used. The scaling factors of important variables in each function block are variably decided in every subframe by checking the dynamic range of variables in the subframe. The direct double precision multiplication can be avoided by the use of variable scaling factor. The single precision multiplication (16×16 bits) is performed only in the magnitude part and the normalization factor of output variable is adjusted according to variable input scaling factors. This variable scaling method can significantly reduce the complexity in routines such as pitch parameter search and codebook parameter search function block in which they require extensive multiplications. The special efforts are required to reduce the number of instructions in pitch and codebook parameter search routine.

2. QCELP Implementation on the Fixed Point DSP Chip

To implement the QCELP algorithm on TMS320C50 DSP chip, we used a Tiger 5XF DSP evaluation board. This DSP board has 40 MIPS execution time and 64 kword external memories with 10 ns access time and 14 bit A/D & D/A converter. All instructions that operate the QCELP algorithm are written in TMS320C50 assembly language. The program consists of an encoding routine, decoding routine, and interrupt handler routine. The voice signal from microphone is converted into 14 bit PCM data and is sent to the DSP chip

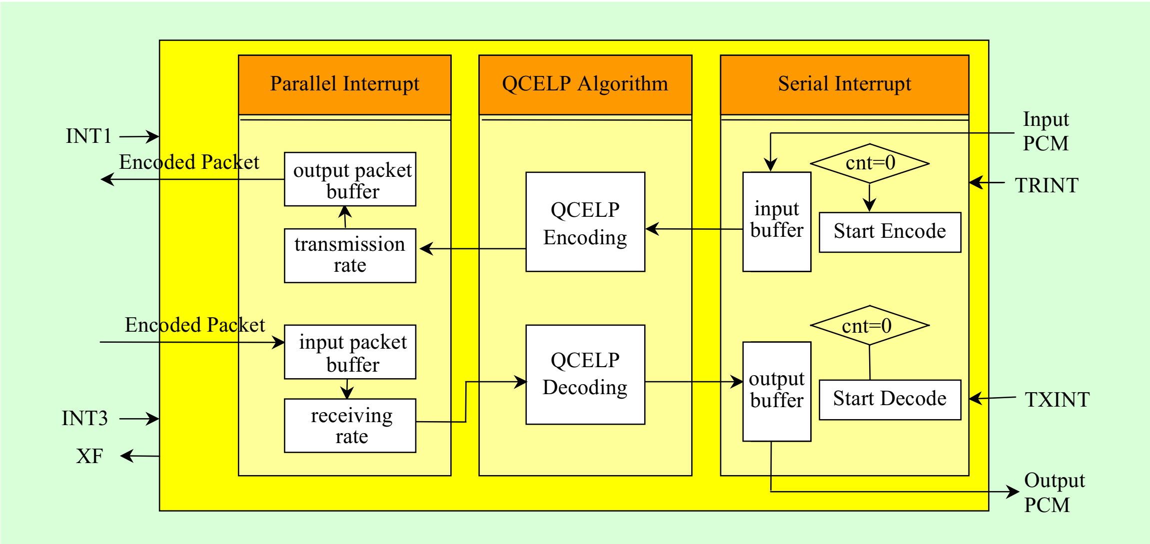

Fig. 7. The block diagram of the implemented speech coder using TMS320C50.

through the serial port. At encoding routine, the PCM data is converted to transmission packet data after the QCELP encoding procedure. The transmission data varies from 1 kbps to 8 kbps. The output PCM data is reconstructed from the transmitted speech packet data in decoding routine. The architecture of real-time QCELP implementation is shown in Fig. 7. The architecture of real-time QCELP implementation is divided into three parts. The encoding/decoding part contains the QCELP speech coder algorithm. The serial port processing part is responsible for transmitting and receiving PCM and also controls the start of encoding and decoding. The external command processing part interprets commands and controls transmission/receiving of encoded data between the speech coder controller and processing several external commands, using interrupt pin (INT 3) and XF pin of TMS320C50 processor.

Speech coder is located at transcoder and selector board (TSB) in the BSC of the CMS. When power is on, CDSP (Control DSP) downloads the execution code of the QCELP coder to the SRAM from the external ROM in the TSB board. In the beginning, data memory, internal register, and serial port of the TMS320C50 are initialized. In addition to the normal call test, DSP initialization, sleep mode, offset control, PCM loop back and packet loopback were done for the interface test between DSP and system. The functional block of the speech coder DSP in Fig.7 includes parallel interrupt for receiving the control information in the system and processing the coded packet data, serial interrupt for transmitting and receiving PCM data, vocoder reference strobe (VRS) for system synchronization, and QCELP speech coder encoding/decoding processing.

Three types of call are possible. There are mobile to mobile, mobile to PSTN, and PSTN to mobile call. In case of mobile to mobile call, two cases are possible. The first is double vocoding where encoding and decoding procedure is done twice. The second is bypass mode which the encoded packets from the mobile are not decoded at the speech coder DSP in BSC and passed directly to the control DSP. The latter shows the better performance in quality than the former.

3. Experimental Result

We evaluated the performance of the fixed point implemention of QCELP algorithm through signal-to-noise ratio (SNR) and an informal listening test. Table. 6 shows the SNR of floating point and fixed point QCELP coder. The performance degradation due to fixed point implementation was about 0.15 dB. However, the degradation of speech quality was not perceived in an informal listening test. We used 5 English sentences and 11 Korean sentences for evaluation.

Table. 7 and 8 show the MIPS in the each subroutine of the algorithm. Total 33 MIPs which is used for the real time processing is less than the processing power (40MIPS) of the TMS320C50. Program memory is 10k words and data memory is 4k words. Hence it is possible to adopt C53 with low power consumption, small size, and internal RAM.

The mean opinion score (MOS) is quantifier of subjectively rated transmission performance, which is computed by averaging the individual opinion scores for each circuit condition evaluated by a sample of listeners. Typically opinion scores represent five point scale {excellent, good, fair, poor, bad} which is mapped to the decimal {5, 4,

Table 6. Performance comparison (SNR) of floating point QCELP and fixed point QCELP.

Sentence	Floating Point Simulation	Fixed Point Simulation
English Sentences	12.37 dB	12.24 dB
Korean Sentences	15.23 dB	15.07 dB

Table 7. Complexity in the encoding block.

Subroutine	Instruction Cycle	# of runs / Frame	MIPS
LPC Analysis	23,000	1	1.1
LPC to LSP	26,000	1	1.3
LSP Quantization	450	1	0.02
Rate Decision	630	1	0.03
Pitch Search	60,000	4	12
Codebook Search	35,000	8	14
LSP to LPC	3,520	4	0.7
Reconstruct	4,560	8	1.8
Total of encode			31

Table 8. Complexity in the decoding block.

Subroutine	Instruction Cycle	# of runs / Frame	MIPS
1st subframe decode	14,000	1	0.7
2nd subframe decode	10,000	1	0.5
3rd subframe decode	10,000	1	0.5
4th subframe decode	10,000	1	0.5
total of decode			2.2

3, 2, 1}. We informally evaluate fixed point implementation of QCELP coder's performance using MOS. We record 72 Korean voice sentences of male and female. Each sentence is coded transmission data using fixed point implemented QCELP algorithm and decoded as a sentence. Table. 9 shows the performance of implemented QCELP Vocoder which is evaluated by 20 Listeners. The voice quality is verified to be good and the algorithm has been shown to be very robust in a variety of mobile environments in the field test.

Table 9. Informal listening test result(Error free channel).

	Male	Female	Total
MOS	3.31	2.96	3.11

IV. CONCLUSION

The QCELP speech coder to increase the capacity of the CMS was adopted. Before implementing the QCELP speech coder, the integer simulation was done with C language. We informally evaluated QCELP coder's performance by using MOS and the performance was good. Thereafter, assembler code was programed to be portable for TMS320C50 fixed point DSP chip. After several interface test, three types of call are tested. In the normal call test on the CMS, where mobile to mobile call test was done in the bypass mode without double vocoding, the speech quality was good.

As a further development, 13 kbps speech coder is under review. Algebraic [10] or cyclic speech coder which has speech quality will be equivalent to 32 kbps ADPCM .

REFERENCES

[1] Cellular Telecommunications Industry Association, *User's Performance Requirements*, Sep. 1988.

[2] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): High quality speech at very low bit rates," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 937-940, 1985.

[3] Bishnu S. Atal, Vladmior Cuperman, Allen Gersho, *Speech and Audio Coding for Wireless and Network Applications*. Kluwer Academic Publishers, 1993.

[4] Shuzo Saito and Kazuo Kakata, *Fundamentals of Speech Signal Processing*. Academic Press, 1985.

[5] Noboru Sugamura and Nariman Farvardin, "Quantizer design in LSP speech analysis– synthesis," *IEEE Journal on Selected Areas in*

Communications, vol. 6, no. 2, pp. 432- 440, 1988.

[6] TIA/EIA IS-96, *Speech Service Option Standard for Wideband Spread Sprectrum Digital Cellular System*, 1994.

[7] D. Lin, "Speech coding using pseudo-stochastic block codes," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 1354-1357. Apr. 1987

[8] M. H. Sunwoo and S. Park, "Real-time implementation of the VSELP on a 16-bit DSP chip," *IEEE Trans. Consumer Electronics*, pp. 772-782, vol. 37, no. 4, Nov. 1991.

[9] B. Yoon, I. Lee, J. Kim, W. Lee, S. Jang, M. Lim, and K. Han Jang, "Real time implementation of QCELP vocoder in CDMA system using TMS320C50 fixed point DSP chip," *Proc. International Conference on Signal Processing Applications & Technology*, Boston, USA, Oct. 1995. pp. 1841-1845.

[10] Redwan Salami, Claude Laflamme, Jean-Pierre Adoul, and Dominique Massaloux, "A toll quality 8 Kb/s speech codec for the personal communications system," *IEEE Trans. on Vehicular Technology*, vol. 43, no. 3, pp. 808-816, Aug. 1994.

Byung Sik Yoon was born on Oct. 31, 1967 in KOREA. He received the B.S. M.S. degrees from Kyungbook National University in 1990 and 1992, respectively. He has worked at Electronics and Telecommunications Research Institute (ETRI) from 1992 as a member of the research staff and is currently working at Signal Processing Section in Mobile Communications System Department. He has been involved in development of CDMA digital mobile communication system. His research interests include digital signal processing, speech processing, mobile communications, and image processing.

Jae Won Kim received his B.S. and M.S. degree in electronics engineering from Kyoungpook National University, Taegu, Korea in 1986 and 1988, respectively. Since 1991 he has been a senior research staff engineer at ETRI, Taejon, Korea. He was involved in research projects including speech coding and digital mobile system. His research interests include digital signal processing, speech processing, and digital mobile communication system.

Won Myoung Lee received his B.S. and M.S. degree in telecommunication and information engineering from Hankuk Aviation University, Seoul, Korea in 1991 and 1993, respectively. Since 1993 he has been a research staff engineer at ETRI, Taejeon, Korea. He was involved in research project including digital mobile communication, speech processing. His research interests include digital signal processing, speech coding, and noise control algorithm.

Seok Jin Jang received BS degree from electronics engineering, Korea University in 1991. and received MS degree in electronics engineering, Korea University in 1993. He has been a nember of research staff of ETRI since 1993. He is interested in digital signal processing on wireless communications.

Song In Choi received his B.S. degree in electronics engineering and his M.S. degree in computer engineering from Kwangwoon University, Seoul, Korea in 1982 and 1987 respectively. Since 1982 he has been a senior research staff engineer at ETRI, Taejeon, Korea. He was involved in research including project TDX and CDMA mobile communication. He is the Head of Signal Processing Section. His research interests include digital mobile communication, speech coding, image coding, and system architecture.

Myoung Seob Lim received his B.S., M.S. and Ph.D. degree in electronics engineering from Yeonsei University in 1980, 1982 and 1990 respectively. he had worked at Daewoo Telecommunication, Inc. from January 1984 for one and a half year and moved to ETRI in September 1985. He has involved in development of CDMA digital cellular system, especially QCELP vocoder implementation for CDMA mobile systems and developed a wideband CDMA modem for the next generation CDMA system. He moved to Chonbuk National University in October 1996 and is now an Assistant Professor at Department of Computer & Communication Engineering. His research interests are spread spectrum communication, signal processing and cellular engineering.