

Vector Quantizer Method and Apparatus

5 Field of the Invention

The present invention generally relates to speech coders using Code Excited Linear Predictive Coding (CELP), Stochastic Coding or Vector Excited Speech Coding and more specifically
10 to vector quantizers for Vector-Sum Excited Linear Predictive Coding (VSELP).

Background of the Invention

15 Code-excited linear prediction (CELP) is a speech coding technique used to produce high quality synthesized speech. This class of speech coding, also known as vector-excited linear prediction, is used in numerous speech communication and speech synthesis applications. CELP is particularly
20 applicable to digital speech encrypting and digital radiotelephone communications systems wherein speech quality, data rate, size and cost are significant issues.

In a CELP speech coder, the long-term (pitch) and the short-term (formant) predictors which model the
25 characteristics of the input speech signal are incorporated in a set of time varying filters. Specifically, a long-term and a short-term filter may be used. An excitation signal for the filters is chosen from a codebook of stored innovation sequences, or codevectors.

30 For each frame of speech, an optimum excitation signal is chosen. The speech coder applies an individual codevector to

REPLACED SHEETS		SG
☐	Description : paged	to paged
☒	Claim(s) : paged	to paged
☐	Drawing(s) : paged	to paged
☐	Abstract : paged	to paged
DATE OF RECEIPT		

the filters to generate a reconstructed speech signal. The reconstructed speech signal is compared to the original input speech signal, creating an error signal. The error signal is then weighted by passing it through a spectral noise weighting filter. The spectral noise weighting filter has a response based on human auditory perception. The optimum excitation signal is a selected codevector which produces the weighted error signal with the minimum energy for the current frame of speech.

Typically, linear predictive coding (LPC) is used to model the short term signal correlation over a block of samples, also referred to as the short term filter. The short term signal correlation represents the resonance frequencies of the vocal tract. The LPC coefficients are one set of speech model parameters. Other parameter sets may be used to characterize the excitation signal which is applied to the short term predictor filter. These other speech model parameters include: Line Spectral Frequencies (LSF), cepstral coefficients, reflection coefficients, log area ratios, and arc sines.

A speech coder typically vector quantizes the excitation signal to reduce the number of bits necessary to characterize the signal. The LPC coefficients may be transformed into the other previously mentioned parameter sets prior to quantization. The coefficients may be quantized individually (scalar quantization) or they may be quantized as a set (vector quantization). Scalar quantization is not as efficient as vector quantization, however, scalar quantization is less expensive in computational and memory requirements than vector quantization. Vector quantization of LPC parameters is used for applications where coding efficiency is of prime concern.

Multi-segment vector quantization may be used to balance coding efficiency, vector quantizer search complexity, and vector quantizer storage requirements. The first type of multi-segment vector quantization partitions a N_p -element LPC parameter vector into n segments. Each of the n segments is vector quantized separately. A second type of multi-segment vector quantization partitions the LPC parameter among n vector codebooks, where each vector codebook spans all N_p vector elements. For illustration of vector quantization assume $N_p = 10$ elements and each element is represented by 2 bits. Traditional vector quantization would require 2^{20} codevectors of 10 elements each to represent all the possible codevector possibilities. The first type of multi-segment vector quantization with two segments would require $2^{10} + 2^{10}$ codevectors of 5 elements each. The second type of multi-segment vector quantization with 2 segments would require $2^{10} + 2^{10}$ codevectors of 5 elements each. Each of these methods of vector quantization offering differing benefits in coding efficiency, search complexity and storage requirements. Thus, the speech coder state of the art would benefit from a vector quantizer method and apparatus which increases the coding efficiency or reduces search complexity or storage requirements without changes in the corresponding requirements.

Brief Description of the Drawings

FIG. 1 is a block diagram of a radio communication system including a speech coder in accordance with the present invention.

FIG. 2 is a block diagram of a speech coder in accordance with the present invention.

FIG. 3 is a graph of the arcsine function used in accordance with the present invention.

5

Description of a Preferred Embodiment

A variation on Code Excited Linear Predictive Coding (CELP) called Vector-Sum Excited Linear Predictive Coding (VSELP), described herein, is a preferred embodiment of the
10 present invention. VSELP uses an excitation codebook having a predefined structure, such that the computations required for the codebook search process are significantly reduced. This VSELP speech coder uses a single or multi-segment
15 vector quantizer of the reflection coefficients based on a Fixed-Point-Lattice-Technique (FLAT). Additionally, this speech coder uses a pre-quantizer to reduce the vector codebook search complexity and a high-resolution scalar quantizer to reduce the amount of memory needed to store the reflection coefficient
20 vector codebooks. The result is a high performance vector quantizer of the reflection coefficients, which is also computationally efficient, and has reduced storage requirements.

FIG. 1 is a block diagram of a radio communication
25 system 100. The radio communication system 100 includes two transceivers 101, 113 which transmit and receive speech data to and from each other. The two transceivers 101, 113 may be part of a trunked radio system or a radiotelephone communication system or any other radio communication
30 system which transmits and receives speech data. At the transmitter, the speech signals are input into microphone 108,

and the speech coder selects the quantized parameters of the speech model. The codes for the quantized parameters are then transmitted to the other transceiver 113. At the other transceiver 113, the transmitted codes for the quantized parameters are received 121 and used to regenerate the speech in the speech decoder 123. The regenerated speech is output to the speaker 124.

FIG. 2 is a block diagram of a VSELP speech coder 200. A VSELP speech coder 200 uses a received code to determine which excitation vector from the codebook to use. The VSELP coder uses an excitation codebook of 2^M codevectors which is constructed from M basis vectors. Defining $v_m(n)$ as the m th basis vector and $u_i(n)$ as the i th codevector in the codebook, then:

$$u_i(n) = \sum_{m=1}^M \theta_{im} v_m(n) \quad (1.10)$$

where $0 \leq i \leq 2^M - 1$; $0 \leq n \leq N - 1$. In other words, each codevector in the codebook is constructed as a linear combination of the M basis vectors. The linear combinations are defined by the θ parameters.

θ_{im} is defined as:

$$\theta_{im} = +1 \text{ if bit } m \text{ of codeword } i = 1$$

$$\theta_{im} = -1 \text{ if bit } m \text{ of codeword } i = 0$$

Codevector i is constructed as the sum of the M basis vectors where the sign (plus or minus) of each basis vector is determined by the state of the corresponding bit in codeword i . Note that if we complement all the bits in codeword i , the corresponding codevector is the negative of codevector i . Therefore, for every codevector, its negative is also a codevector in the codebook. These pairs are called complementary

codevectors since the corresponding codewords are complements of each other.

After the appropriate vector has been chosen, the gain block 205 scales the chosen vector by the gain term, γ . The output of the gain block 205 is applied to a set of linear filters 207, 209 to obtain N samples of reconstructed speech. The filters include a "long-term" (or "pitch") filter 207 which inserts pitch periodicity into the excitation. The output of the "long-term" filter 207 is then applied to the "short-term" (or "formant") filter 209. The short term filter 209 adds the spectral envelope to the signal.

The long-term filter 207 incorporates a long-term predictor coefficient (LTP). The long-term filter 207 attempts to predict the next output sample from one or more samples in the distant past. If only one past sample is used in the predictor, then the predictor is a single-tap predictor. Typically one to three taps are used. The transfer function for a long-term ("pitch") filter 207 incorporating a single-tap long-term predictor is given by (1.1).

$$B(z) = \frac{1}{1 - \beta z^{-L}} \quad (1.1)$$

$B(z)$ is characterized by two quantities L and β . L is called the "lag". For voiced speech, L would typically be the pitch period or a multiple of it. L may also be a non integer value. If L is a non integer, an interpolating finite impulse response (FIR) filter is used to generate the fractionally delayed samples. β is the long-term (or "pitch") predictor coefficient.

The short-term filter 209 incorporates short-term predictor coefficients, α_i , which attempt to predict the next output sample from the preceding N_p output samples. N_p

typically ranges from 8 to 12. In the preferred embodiment, N_p is equal to 10. The short-term filter 209 is equivalent to the traditional LPC synthesis filter. The transfer function for the short-term filter 209 is given by (1.2).

$$A(z) = \frac{1}{1 - \sum_{i=1}^{N_p} \alpha_i z^{-i}} \quad (1.2)$$

The short-term filter 209 is characterized by the α_i parameters, which are the direct form filter coefficients for the all-pole "synthesis" filter. Details concerning the α_i parameters can be found below.

The various parameters (code, gain, filter parameters) are not all transmitted at the same rate to the synthesizer (speech decoder). Typically the short term parameters are updated less often than the code. We will define the short term parameter update rate as the "frame rate" and the interval between updates as a "frame". The code update rate is determined by the vector length, N . We will define the code update rate as the "subframe rate" and the code update interval as a "subframe". A frame is usually composed of an integral number of subframes. The gain and long-term parameters may be updated at either the subframe rate, the frame rate or some rate in between depending on the speech coder design.

The codebook search procedure consists of trying each codevector as a possible excitation for the CELP synthesizer. The synthesized speech, $s'(n)$, is compared against the input speech, $s(n)$, and a difference signal, e_i , is generated. This difference signal, $e_i(n)$, is then filtered by a spectral weighting filter, $W(z)$, (and possibly a second weighting

filter, $C(z)$) to generate a weighted error signal, $e'(n)$. The power in $e'(n)$ is computed at the energy calculator 215. The codevector which generates the minimum weighted error power is chosen as the codevector for that subframe. The spectral weighting filter 213 serves to weight the error spectrum based on perceptual considerations. This weighting filter 213 is a function of the speech spectrum and can be expressed in terms of the α parameters of the short term (spectral) filter 209.

$$W(z) = \frac{1 - \sum_{i=1}^{N_b} \alpha_i z^{-i}}{1 - \sum_{i=1}^{N_b} \tilde{\alpha}_i z^{-i}} \quad (1.3)$$

There are two approaches that can be used for calculating the gain, γ . The gain can be determined prior to codebook search based on residual energy. This gain would then be fixed for the codebook search. Another approach is to optimize the gain for each codevector during the codebook search. The codevector which yields the minimum weighted error would be chosen and its corresponding optimal gain would be used for γ . The latter approach generally yields better results since the gain is optimized for each codevector. This approach also implies that the gain term must be updated at the subframe rate. The optimal code and gain for this technique can be computed as follows:

1. Compute $y(n)$, the weighted input signal, for the subframe.
2. Compute $d(n)$; the zero-input response of the $B(z)$ and $W(z)$ (and $C(z)$ if used) filters for the subframe. (Zero input response

is the response of the filters with no input; the decay of the filter states.)

3. $p(n) = y(n) - d(n)$ over subframe ($0 \leq n \leq N-1$)

4. for each code i

5 a. Compute $g_i(n)$, the zero state response of $B(z)$ and $W(z)$ (and $C(z)$ if used) to codevector i . (Zero-state response is the filter output with initial filter states set to zero.)

 b. Compute

$$C_i = \sum_{n=0}^{N-1} g_i(n) p(n) \quad (1.5)$$

10 the cross correlation between the filtered codevector i and $p(n)$

 c. Compute

$$G_i = \sum_{n=0}^{N-1} [g_i(n)]^2 \quad (1.6)$$

 the power in the filtered codevector i .

15 5. Choose i which maximizes $\frac{(C_i)^2}{G_i}$ (1.7)

6. Update filter states of $B(z)$ and $W(z)$ (and $C(z)$ if used) filters using chosen codeword and its corresponding quantized gain. This is done to obtain the same filter states that the synthesizer would have at the start of the next subframe for

20 step 2.

The optimal gain for codevector i is given by (1.8)

$$\gamma_i = \frac{C_i}{G_i} \quad (1.8)$$

And the total weighted error for codevector i using the optimal gain, γ_i is given by (1.9).

$$E_i = \left(\sum_{n=0}^{N-1} p^2(n) \right) \cdot \frac{(C_i)^2}{G_i} \quad (1.9)$$

The short term predictor parameters are the α_i 's of the short term filter 209 of FIG. 2. These are standard LPC direct form filter coefficients and any number of LPC analysis techniques can be used to determine these coefficients. In the preferred embodiment, a fast fixed point covariance lattice algorithm (FLAT) was implemented. FLAT has all the advantages of lattice algorithms including guaranteed filter stability, non-windowed analysis, and the ability to quantize the reflection coefficients within the recursion. In addition FLAT is numerically robust and can be implemented on a fixed-point processor easily.

The short term predictor parameters are computed from the input speech. No pre-emphasis is used. The analysis length used for computation of the parameters is 170 samples ($N_A = 170$). The order of the predictor is 10 ($N_P = 10$).

This section will describe the details of the FLAT algorithm. Let the samples of the input speech which fall in the analysis interval be represented by $s(n)$; $0 \leq n \leq N_A - 1$. Since FLAT is a lattice algorithm one can view the technique as trying to build an optimum (that which minimizes residual energy) inverse lattice filter stage by stage. Defining $b_j(n)$ to be the backward residual out of stage j of the inverse lattice filter and $f_j(n)$ to be the forward residual out of stage j of the inverse lattice filter we can define:

$$F_j(i, k) = \sum_{n=N_P}^{N_A-1} f_j(n-i) f_j(n-k) \quad (2.1)$$

the autocorrelation of $f_j(n)$;

$$B_j(i,k) = \sum_{n=N_p}^{N_A-1} b_j(n-i-1)b_j(n-k-1) \quad (2.2)$$

the autocorrelation of $b_j(n-1)$ and:

$$C_j(i,k) = \sum_{n=N_p}^{N_A-1} f_j(n-i)b_j(n-k-1) \quad (2.3)$$

5 the cross correlation between $f_j(n)$ and $b_j(n-1)$.

Let r_j represent the reflection coefficient for stage j of the inverse lattice. Then:

$$F_j(i,k) = F_{j-1}(i,k) + r_j(C_{j-1}(i,k) + C_{j-1}(k,i)) + r_j^2 B_{j-1}(i,k) \quad (2.4)$$

and

10

$$B_j(i,k) = B_{j-1}(i+1,k+1) + r_j(C_{j-1}(i+1,k+1) + C_{j-1}(k+1,i+1)) + r_j^2 F_{j-1}(i+1,k+1) \quad (2.5)$$

and

15

$$C_j(i,k) = C_{j-1}(i,k+1) + r_j(B_{j-1}(i,k+1) + F_{j-1}(i,k+1)) + r_j^2 C_{j-1}(k+1,i) \quad (2.6)$$

The formulation we have chosen for the determination of r_j can be expressed as:

$$r_j = -2 \frac{C_{j-1}(0,0) + C_{j-1}(N_p-j, N_p-j)}{F_{j-1}(0,0) + B_{j-1}(0,0) + F_{j-1}(N_p-j, N_p-j) + B_{j-1}(N_p-j, N_p-j)} \quad (2.7)$$

20

The FLAT algorithm can now be stated as follows.

1. First compute the covariance (autocorrelation) matrix from the input speech:

$$\phi(i,k) = \sum_{N_p}^{N_A-1} s(n-i) s(n-k) \quad (2.8)$$

for $0 \leq i,k \leq NP$.

$$2. \quad F0(i,k) = f(i,k) \quad 0 \leq i,k \leq NP-1 \quad (2.9)$$

$$B0(i,k) = f(i+1,k+1) \quad 0 \leq i,k \leq NP-1 \quad (2.10)$$

$$C0(i,k) = f(i,k+1) \quad 0 \leq i,k \leq NP-1 \quad (2.11)$$

3. set $j = 1$

5 4. Compute r_j using (2.7)

5. If $j = NP$ then done.

6. Compute $F_j(i,k) \quad 0 \leq i,k \leq NP-j-1$ using (2.4)

Compute $B_j(i,k) \quad 0 \leq i,k \leq NP-j-1$ using (2.5)

Compute $C_j(i,k) \quad 0 \leq i,k \leq NP-j-1$ using (2.6)

10 7. $j = j+1$; go to 4.

Prior to solving for the reflection coefficients, the ϕ array is modified by windowing the autocorrelation functions.

$$\phi'(i,k) = \phi(i,k)w(|i-k|) \quad (2.12)$$

15 Windowing of the autocorrelation function prior to reflection coefficient computation is known as spectral smoothing (SST).

From the reflection coefficients, r_j , the short term LPC predictor coefficients, α_j , may be computed.

A 28-bit three segment vector quantizer of the reflection coefficients is employed. The segments of the vector quantizer span reflection coefficients r_1 - r_3 , r_4 - r_6 , and r_7 - r_{10} respectively. The bit allocations for the vector quantizer segments are:

20 Q_1 11 bits
 Q_2 9 bits
 25 Q_3 8 bits.

To avoid the computational complexity of an exhaustive vector quantizer search, a reflection coefficient vector prequantizer is used at each segment. The prequantizer size at each segment is:

P1 6 bits
P2 5 bits
P3 4 bits

At a given segment, the residual error due to each vector from
5 the prequantizer is computed and stored in temporary
memory. This list is searched to identify the four prequantizer
vectors which have the lowest distortion. The index of each
selected prequantizer vector is used to calculate an offset into
the vector quantizer table at which the contiguous subset of
10 quantizer vectors associated with that prequantizer vector
begins. The size of each vector quantizer subset at the k-th
segment is given by:

$$S_k = \frac{2^{Q_k}}{2^{P_k}} \quad (2.13)$$

The four subsets of quantizer vectors, associated with the
15 selected prequantizer vectors, are searched for the quantizer
vector which yields the lowest residual error. Thus at the first
segment 64 prequantizer vectors and 128 quantizer vectors are
evaluated, 32 prequantizer vectors and 64 quantizer vectors are
evaluated at the second segment, and 16 prequantizer vectors
20 and 64 quantizer vectors are evaluated at the third segment.
The optimal reflection coefficients, computed via the FLAT
technique with bandwidth expansion as previously described
are converted to an autocorrelation vector prior to vector
quantization.

25 An autocorrelation version of the FLAT algorithm,
AFLAT, is used to compute the residual error energy for a
reflection coefficient vector being evaluated. Like FLAT, this
algorithm has the ability to partially compensate for the
reflection coefficient quantization error from the previous
30 lattice stages, when computing optimal reflection coefficients

or selecting a reflection coefficient vector from a vector quantizer at the current segment. This improvement can be significant for frames that have high reflection coefficient quantization distortion. The AFLAT algorithm, in the context
 5 of multi-segment vector quantization with prequantizers, is now described:

Compute the autocorrelation sequence $R(i)$, from the optimal reflection coefficients, over the range $0 \leq i \leq N_p$. Alternatively,
 10 the autocorrelation sequence may be computed from other LPC parameter representations, such as the direct form LPC predictor coefficients, α_i , or directly from the input speech.

Define the initial conditions for the AFLAT recursion:
 15

$$\bar{P}_0(i) = R(i), \quad 0 \leq i \leq N_p - 1 \quad (2.14)$$

$$\bar{V}_0(i) = R(i+1), \quad 1 - N_p \leq i \leq N_p - 1 \quad (2.15)$$

20 Initialize k , the vector quantizer segment index:

$$k = 1 \quad (2.16)$$

Let $l_l(k)$ be the index of the first lattice stage in the k -th segment, and $l_h(k)$ be the index of the last lattice stage in the k -th segment. The recursion for evaluating the residual error
 25 out of lattice stage $l_h(k)$ at the k -th segment, given $\hat{r}$, a reflection coefficient vector from the prequantizer or the reflection coefficient vector from the quantizer is given below.

Initialize j , the index of the lattice stage, to point to the
 30 beginning of the k -th segment:

$$j = I_l(k) \quad (2.17)$$

Set the initial conditions P_{j-1} and V_{j-1} to:

$$P_{j-1}(i) = \bar{P}_{j-1}(i), \quad 0 \leq i \leq I_h(k) - I_l(k) + 1 \quad (2.18)$$

$$V_{j-1}(i) = \bar{V}_{j-1}(i), \quad -I_h(k) + I_l(k) - 1 \leq i \leq I_h(k) - I_l(k) + 1 \quad (2.19)$$

5

Compute the values of V_j and P_j arrays using:

$$P_j(i) = (1 + \hat{r}_j^2) P_{j-1}(i) + \hat{r}_j [V_{j-1}(i) + V_{j-1}(-i)], \quad 0 \leq i \leq I_h(k) - j \quad (2.20)$$

$$V_j(i) = V_{j-1}(i+1) + \hat{r}_j^2 V_{j-1}(-i-1) + 2\hat{r}_j P_{j-1}(i+1), \quad j - I_h(k) \leq i \leq I_h(k) - j \quad (2.21)$$

10

Increment j :

$$j = j + 1 \quad (2.22)$$

If $j \leq I_h(k)$ go to (2.20).

15

The residual error out of lattice stage $I_h(k)$, given the reflection coefficient vector $\hat{r}$, is given by:

$$E_r = P_{I_h(k)}(0) \quad (2.23)$$

20 Using the AFLAT recursion outlined, the residual error due to each vector from the prequantizer at the k -th segment is evaluated, the four subsets of quantizer vectors to search are identified, and residual error due to each quantizer vector from the selected four subsets is computed. The index of $\tilde{r}$, the
25 quantizer vector which minimized E_r over all the quantizer vectors in the four subsets, is encoded with Q_k bits.

If $k < 3$ then the initial conditions for doing the recursion at segment $k+1$ need to be computed. Set j , the lattice stage index,
30 equal to:

$$j = I_1(k) \quad (2.24)$$

Compute:

$$\bar{P}_j(i) = (1 + \tilde{r}_j^2) \bar{P}_{j-1}(i) + \tilde{r}_j [\bar{V}_{j-1}(i) + \bar{V}_{j-1}(-i)], \quad 0 \leq i \leq N_p - j - 1 \quad (2.25)$$

5

$$\bar{V}_j(i) = \bar{V}_{j-1}(i+1) + \tilde{r}_j^2 \bar{V}_{j-1}(-i-1) + 2\tilde{r}_j \bar{P}_{j-1}(i+1), \quad j - N_p + 1 \leq i \leq N_p - j - 1 \quad (2.26)$$

Increment j,

$$j = j + 1 \quad (2.27)$$

If $j \leq I_h(k)$ go to (2.25).

Increment k, the vector quantizer segment index:

$$k = k + 1 \quad (2.28)$$

If $k \leq 3$ go to (2.17). Otherwise, the indices of the reflection coefficient vectors for the three segments have been chosen, and the search of the reflection coefficient vector quantizer is terminated.

To minimize the storage requirements for the reflection coefficient vector quantizer, eight bit codes for the individual reflection coefficients are stored in the vector quantizer table, instead of the actual reflection coefficient values. The codes are used to look up the values of the reflection coefficients from a scalar quantization table with 256 entries. The eight bit codes represent reflection coefficient values obtained by uniformly sampling an arcsine function illustrated in FIG. 3. Reflection coefficient values vary from -1 to +1. The non-linear spacing in the reflection coefficient domain (X axis) provides more

precision for reflection coefficients when the values are near the extremes of ± 1 and less precision when the values are near 0. This reduces the spectral distortion due to scalar quantization of the reflection coefficients, given 256
5 quantization levels, as compared to uniform sampling in the reflection coefficient domain.

What is claimed is:



1. A method of vector quantizing an optimal reflection coefficient vector, the method comprising the steps of:

- a) segmenting the optimal reflection coefficient vector into
5 at least a first and a second segment;
- b) providing a first array of predetermined vectors of reflection coefficients, each vector having multiple elements;
- c) selecting a first vector from the first array of predetermined vectors, providing a first selected vector;
- 10 d) calculating the residual error corresponding to the first selected vector;
- e) repeating steps c and d for each vector of the first array;
- f) choosing a vector from the first array having the lowest residual error, forming a first chosen vector;
- 15 g) defining, responsive to the first chosen vector, initial conditions for the second segment;
- h) providing a second array of predetermined vectors of reflection coefficients, each vector having multiple elements; and
- 20 i) repeating steps c-f for the second segment, using the second array of predetermined vectors and forming a second chosen vector.





2. A method of vector quantizing an optimal reflection coefficient vector, the method comprising the steps of:

- a) segmenting the optimal reflection coefficient vector into
5 at least a first and a second segment;
- b) providing a first array of predetermined vectors of reflection coefficients, each vector having multiple elements;
- c) providing an autocorrelation vector corresponding to the optimal reflection coefficient vector;
- 10 d) defining, responsive to the step of providing an autocorrelation vector, initial conditions for a correlation array and a cross-correlation array;
- e) setting the correlation array and the cross-correlation array to the defined initial conditions;
- 15 f) selecting a first vector from the first array of predetermined vectors, providing a first selected vector;
- g) updating the correlation array and cross-correlation array for each element of the first selected vector;
- h) defining, responsive to the step of updating, the
20 residual error corresponding to the first selected vector;
- i) repeating steps e-h for each vector of the first array;
- j) choosing a vector from the first array having the lowest residual error, forming a first chosen vector;
- k) defining, responsive to the first chosen vector, initial
25 conditions for the second segment;
- l) providing a second array of predetermined vectors of reflection coefficients, each vector having multiple elements; and
- 30 m) repeating steps e-j for the second segment, using the second array of predetermined vectors and forming a second chosen vector.





3. A method of vector quantizing an optimal reflection coefficient vector comprising the steps of:

- 5 providing a first array of X predetermined vectors of reflection coefficients;
- pre-quantizing the optimal reflection coefficient vector comprising the steps of:
 - providing a second array of Y predetermined vectors of reflection coefficients, wherein X is
 - 10 greater than Y,
 - relating each of the Y predetermined vectors to at least one of the X predetermined vectors,
 - calculating the residual error corresponding to each of the Y predetermined vectors, and
 - 15 choosing, responsive to the residual error, a portion of the Y predetermined vectors, forming chosen Y predetermined vectors;
 - selecting a subset of the X predetermined vectors which are related to the chosen Y predetermined vectors;
 - 20 determining the residual error corresponding to each vector of the subset of X predetermined vectors; and
 - choosing the vector of the subset of X predetermined vectors which has the lowest residual error.





4. A method of speech coding comprising the steps of:
receiving speech data, forming a speech data vector;
providing a first array of predetermined vectors;
5 choosing a first predetermined vector from the first array,
forming a first chosen vector representing a first segment of
the speech data vector;
providing a second array of predetermined vectors; and
choosing a second predetermined vector from the second
10 array, forming a second chosen vector representing a second
segment of the speech data vector.

This Sheet Has Been Replaced





5. A radio communication system comprising:

a first transceiver comprising:

5 means for receiving data, forming a data vector,

 means for providing a first array of predetermined vectors,

10 means for choosing a first predetermined vector from the first array, forming a first chosen vector representing a first segment of the speech data vector,

 means for providing a second array of predetermined vectors, and

15 means for choosing a second predetermined vector from the second array, forming a second chosen vector representing a second segment of the speech data vector, and

20 means for transmitting the first and the second chosen vector to a second transceiver; and

a second transceiver comprising:

25 means for receiving the first and the second chosen vector, and

 means, responsive to said means for receiving, for reconstructing the data vector.





6. A method of storing reflection coefficient vectors comprising the steps of:

5 providing a table having 2^N reflection coefficient values, each reflection coefficient value addressable with an N-bit code; and

storing an N-bit code to represent one of the reflection coefficient values in the table.

10 7. A method of storing in accordance with claim 6 wherein the reflection coefficient values are non-linearly scaled.

15 8. A method of storing in accordance with claim 6 wherein the reflection coefficient values are arcsine scaled between the values of -1 and +1.

This Sheet Has Been Replaced

