2266822

SPEECH CODING

The present application is concerned with methods of, and apparatus for, the coding of speech signals; particularly (though not exclusively) to code existed linear predictive coding (LPC) in which input speech is analysed to derive the parameters of an appropriate time-varying synthesis filter, and to select from a "codebook" of excitation signals those which, when (after appropriate scaling) supplied in succession to such a synthesis filter, produce the best approximation to the original speech. The filter parameters, codewords identifying codebook entries, and gains, can be sent to a receiver where they are used to synthesise received speech.

Substantial processing of the signals is required in order to identify the relevant codebook entry. With a view to improving the speed of such processing, the invention provides a method of speech coding in which input speech is analysed to determine the parameters of a synthesis filter and to select at least one excitation component from a plurality of possible components, including the step of determining the scalar product of the response of the filter to an excitation component and the response of the filter to the same or another excitation component, wherein the product of a filter response matrix $\underline{H}$ and its transpose $\underline{H}^T$ to form a product matrix $\underline{H}^T\underline{H}$ is formed once, and each scalar product is formed by multiplying the product matrix by the relevant possible excitation components, in which for each set of diagonal terms of the product matrix, a first member of the set is calculated, and each further member of that set is obtaining by adding a further term to the preceding member of the set.

In another aspect of the invention there is provided a speech coding apparatus comprising

(a) means for analysing an input speech signal to determine the parameters of a synthesis filter; and

(b) means for selecting at least one excitation component from a plurality of possible components by determining the scalar product of the response of the filter to an excitation component and the response of the filter to the same or another excitation component, including means for forming the product of a filter response matrix $\underline{H}$ and its transpose $\underline{H}^T$ to form a product matrix $\underline{H}^T\underline{H}$,

in which the selecting means further includes

(c) a first store for storing elements of the product matrix $\underline{H}^T\underline{H}$;

(d) a second store for storing, for each possible pair of an excitation component and the same or another excitation component, the address of each location in the first store which contains an element of the product matrix which is to by multiplied by nonzero elements of both excitation components of the pair; and

(e) means operable to retrieve addresses from the second store, to retrieve the contents of the locations in the first store thereby addressed, and to add the retrieved contents.

According to a further aspect of the invention there is provided a speech coding apparatus comprising

(a) means for analysing an input speech signal to determine the parameters of a synthesis filter; and

(b) means for selecting at least one excitation component from a plurality of possible components by determining the scalar product of the response of the filter to an excitation component and the response of the filter to the same or another excitation component, including means for forming the product of a filter response matrix $\underline{H}$ and its transpose $\underline{H}^T$ to form a product matrix $\underline{H}^T\underline{H}$; in which the plurality of possible components consists of a plurality of subsets of components, each component of a subset being a shifted version of another member of the same subset, and the selecting means includes

(c) a first store for storing elements of the product matrix $\underline{H}^T\underline{H}$;

(d) a second store storing, for one representative component of each subset of excitation components, the addresses of each location in the first store which contains an element of the product matrix which is to be multiplied by nonzero elements of the representative component;

(e) means operable to retrieve addresses from the second store, to modify the addresses in respect of components other than the representative components, to retrieve the contents of the location in the first store thereby addressed and to add the retrieved contents.

The invention also includes apparatus for implementing the methods mentioned above.

Some embodiments of the invention will now be described, by way of example, with reference to the figures 2 to 10 of the accompanying drawings, in which:

Figure 1 is a block diagram of a conventional decoder;

Figure 2 is a block diagram of a decoder to be used with coders according to the invention;

Figure 3 is a block diagram of a speech coder in accordance with one embodiment of the invention;

Figure 4, 5 and 6 are diagrams illustrating operation of parts of the coder of Figure 3;

Figure 7 is a flowchart demonstrating part of the operation of unit 224 of Figure 3;

Figure 8 is a second embodiment of speech coder according to the invention;

Figure 9 is a diagram illustrating the look-up process used in the coder of Figure 8; and

Figure 10 is a flowchart showing the overall operation of the coders.

Before describing the speech coder, we first describe a decoder to illustrate the manner in which the coded signals are used upon receipt to synthesis a speech signal. The basic structure involves the generation of an excitation signal, which is then filtered.

Commonly in such systems a long-term predictor is employed in addition to the LPC filter. This is best

illustrated by reference to Figure 1 of the accompanying drawings, which shows a block diagram of a decoder. The coded signal includes a codeword identifying one of a number of stored excitation pulse sequences and a gain value; the codeword is employed at the decoder to read out the identified sequence from a codebook store 1, which is then multiplied by the gain value in a multiplier 2. Rather than being used directly to drive a synthesis filter, this signal is then added in an adder 3 to a predicted signal to form the desired composite excitation signal. The predicted signal is obtained by feeding back past values of the composite excitation via a variable delay line 4 and a multiplier 5, controlled by a delay parameter and further gain value included in the coded signal. Finally the composite excitation drives an LPC filter 6 having variable coefficients. The rationale behind the use of the long term predictor is to exploit the inherent periodicity of the required excitation (at least during voiced speech); an earlier portion of the excitation forms a prediction to which the codebook excitation is added. This reduces the amount of information that the codebook excitation has to carry, viz it carries information about changes to the excitation rather than its absolute value.

A more complex coder, for use with coders of the invention, is shown in Figure 2. The filter parameters are changed once every 20ms; a 20ms period of the excitation signal being referred to as a block; however the block is assembled from shorter segments ("sub-blocks") of duration 5ms.

Every 5ms the decoder receives a codebook entry code k, and two gain values g_1, g_2 (though only one, or more than two, gain values maybe used if desired). It has a codebook store 100 containing a number (typically 128) of entries each of which defines a 5ms period of excitation at a sampling rate of 8 kHz. The excitation is a ternary signal (i.e. may take values +1, 0 or -1 at each 125μs sampling instant) and each entry contains 40 elements of three bits each, two of

which define the amplitude value. If a sparse codebook (i.e. where each entry has a relatively small number of elements) is used a more compressed representation might however be used.

The code k from an input register 101 is applied as an address to the store 100 to read out an entry into a 3-bit wide parallel-in-serial out register 102. The output of this register (at 8k/samples per second) is then multiplied by one or other of the gains g_1, g_2 from a further input register 103 by multipliers 104, 105; which gain is used for a given sample is determined by the third bit of the relevant stored element, as illustrated schematically by a changeover switch 106.

The filtering is performed in two stages, firstly by a long term predictor indicated generally by reference numeral 107, and then by an LPC (linear predictive coding) filter 108. The LPC filter, of conventional construction, is updated at 20ms intervals with coefficients a_i from an input register 109.

The long term filter is a "single tap" predictor having a variable delay (delay line 110) controlled by signals d from an input register 111 and variable feedback

gain (multiplier 112) controlled by a gain value g from the register 111. An adder 113 forms the sum of the filter input and the delayed scaled signal from the multiplier 112. Although referred to as "single tap" the delay line actually has two outputs one sample period delay apart, with a linear interpolator 114 to form (when required) the average of the two values, thereby providing an effective delay resolution of 1/2 sample period.

The parameters k, g_1, g_2, d, g and a_i are derived from a multiplexed input signal by means of a demultiplexer 115. However, the gains g_1, g_2 and g are identified by a single codeword G which is used to look up a gain combination from a gain codebook store 116 containing 128 such entries.

The task of the coder is to generate, from input speech, the parameters referred to above. The general architecture of the coder is shown in Figure 3. The input speech is divided into frames of digital samples and each frame is analysed by an LPC analysis unit 200 to derive the coefficients a_i of an LPC filter (impulse response $\underline{h}$) having a spectral response similar to that of each 20ms block of input speech. Such analysis is conventional and will not be described further; it is however worth noting that such filters commonly have a recursive structure and the impulse response $\underline{h}$ is (theoretically) infinite in length.

The remainder of the processing is performed on a sub-block by sub-block basis. Preferably the LPC coefficient values used in this process are obtained by LSP (line spectral pair) interpolation between the calculated coefficients for the preceding frame and those for the current frame. Since the latter are not available until the end of the frame this results in considerable system delay; a good compromise is to use the 'previous block' coefficients for the first half of the frame (i.e. in this example, the first two sub-blocks) and interpolated coefficients for the second half (i.e. the third and fourth

sub-blocks). The forwarding and interpolation is performed by an interpolation unit 201.

The input speech sub-block and the LPC coefficients for that sub-block are then processed to evaluate the other parameters. First, however, because the decoder LPC filter, due to the length of its impulse response, will produce for a given sub-block an output in the absence of any input to the filter. This output - the filter memory $\underline{M}$ - is generated by a local decoder 230 and subtracted from the input speech in a subtractor 202 to produce a target speech signal $\underline{y}$. Note that this adjustment does not include any memory contribution from the long term predictor as its new delay is not yet known.

Secondly, this target signal $\underline{y}$ and the LPC coefficients a_i are used in a first analysis unit 203 to find that LTP delay d which produces in a local decoder with optimal LTP gain g and zero excitation a speech signal with minimum difference from the target.

Thirdly, the target signal, coefficients a_i and delay d are used by a second analysis unit 204 to select an entry from a codebook store 205 having the same contents as the decoder store 100, and the gain values g_1, g_2 to be applied to it.

Finally, the gains g, g_1, g_2 are jointly selected to minimise the difference between a local decoder output and the speech input.

Looking in more detail at the first analysis unit 204, this models (Figure 4) a truncated local decoder having a delay line 206, interpolator 207, multiplier 208 and LPC filter 209 identical to components 110, 112, 114 and 108 of Figure 2. The contents of the delay line and the LPC filter coefficients are set up so as to be the same as the contents of the decoder delay line and LPC filter at the commencement of the sub-block under consideration. Also shown is a subtractor 210 which forms the difference between the target signal $\underline{y}$ and the output $g\underline{X}$ of the LPC

filter 209 to form a mean square error signal e^2. $\underline{X}$ is a vector representing the first n samples of a filtered version of the content of the delay line shifted by the (as yet undetermined) integer delay d or (if interpolation is involved) of the mean of the delay line contents shifted by delays d and d+1. The value d will be supposed to have an additional bit to indicate switching between integer delay prediction (with tap weights (0,1) and "half step" prediction with tap weights (½,½). $\underline{y}$ is an n element vector. n is the number of samples per sub-block - 40, in this example. Vectors are, in the matrix analysis used, column vectors - row vectors are shown as the transpose, e.g. "$\underline{y}^T$".
The error is:

$$e^2 = \| \underline{y} - g\underline{X} \|^2 = (\underline{y} - g\underline{X})^T \ (\underline{y}-g\underline{X}) \qquad \ldots(1)$$

$$= \underline{y}^T\underline{y} - g\underline{X}^T\underline{y} - g\,\underline{y}^T\underline{X} + g^2\,\underline{X}^T\underline{X} \qquad \ldots(2)$$

$$= \underline{y}^T\underline{y} - 2\,g\underline{X}^T\underline{y} + g^2\,\underline{X}^T\underline{X} \qquad \ldots(3)$$

To minimise this error we set the differential with respect to g to zero. (Where g' denotes the optimum value of g at this stage).

$$\frac{d(e^2)}{dg} = -2\underline{X}^T\underline{y} + 2g\,\underline{X}^T\underline{X} = 0 \qquad \ldots(4)$$

$$g'\,\underline{X}^T\underline{X} = \underline{X}^T\underline{y} \qquad \ldots(5)$$

$$\text{and } g' = (\underline{X}^T\underline{X})^{-1}\,\underline{X}^T\underline{y} \qquad \ldots(6)$$

Substituting in (3)

$$e^2 = \underline{y}^T\underline{y} - \underline{y}^T\underline{X}\,(\underline{X}^T\underline{X})^{-1}\,\underline{X}^T\underline{y} \qquad \ldots(7)$$

gives the mean square error for optimum gain.

If the delay line output for a delay d is $\underline{D}(d)$, then

$$\underline{X} = \underline{H}\,\underline{D}(d) \qquad \ldots(8)$$

and the second term of equation (7) can be written.

$$\underline{y}^T \underline{H}\,\underline{D}(d)\ [\underline{D}(d)^T \underline{H}^T\underline{H}\,\underline{D}(d)]\ \underline{D}(d)^T \underline{H}^T\underline{y} \qquad \ldots(9)$$

The delay d is found by computing (control unit 211) the second term in equation (7) for each of a series of trial values of d, and selecting that value of d which gives the largest value of that term (see, below, however, for a modification of this procedure). Note that, although apparently a recursive filter, it is more realistic to regard the delay line as being an "adaptive codebook" of excitations. If the smallest trial value of d is less than the sub-block length then one would expect that the new output from the adder 113 of the decoder would be fed back and appear again at the input of the multiplier. (In fact, it is preferred not to do this but to repeat samples. For example, if the sub-block length is s, then the latest d samples would be used for excitation, followed by the oldest s-d of these). The value of the gain g is found from eq. 6.

Returning to Figure 3, the second analysis unit 204 serves to select the codebook entry. An address generator 231 accesses, in sequence, each of the entries in the codebook store 205 for evaluation by the analysis unit 204. The actual excitation at the decoder is the selected entry selectively multiplied by the gains g_1, g_2 (or, more generally, g_1, g_2 ... g_{m-1} where m is the total number of gains including the long term predictor gain g; the mathematics quoted below assumes m=3). The entry can be thought of as being the sum of m-1 partial entries - each containing the non-zero elements to be multiplied by the relevant gain with zeros for the elements to be subjected

to a different gain - each multiplied by a respective gain. The entry is selected by finding, for each entry, the mean squared error - at optimum gain - between the output of a local decoder and the target signal $\underline{y}$.

Suppose the partial entries are $\underline{C}_1$, $\underline{C}_2$ and the selected LTP delay gives an output $\underline{C}_0$ from the delay line.

The total input to the LPC filter is

$$g_1 \underline{C}_1 + g_2 \underline{C}_2 + g \underline{C}_0 \qquad \ldots(10)$$

And the filter output is

$$g_1 \underline{H} \underline{C}_1 + g_2 \underline{H} \underline{C}_2 + g \underline{H} \underline{C}_0 \qquad \ldots(11)$$

Where $\underline{H}$ is a convolution matrix consisting of the impulse response $\underline{h}^T$ and shifted versions thereof.

If the products $\underline{H} \underline{C}_1$, $\underline{H} \underline{C}_2$, $\underline{H} \underline{C}_0$ are written as $\underline{Z}_{i1}$, $\underline{Z}_{i2}$, $\underline{Z}_0$ where i is in the codebook entry, and $(g_1, g_2, g)^T = \underline{g}$ then the decoder output is

$$\underline{Z}_{ij} \cdot \underline{g} = \begin{bmatrix} \underline{Z}_{i1} & \underline{Z}_{i2} & \underline{Z}_0 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g \end{bmatrix} \qquad \ldots(12)$$

$\underline{Z}_{ij}$ is a n x m matrix where n is the number of samples and m the total number of gains.

Thus the mean squared error is

$$e^2 = \| \underline{y} - \underline{Z}_{ij} \underline{g} \|^2 \qquad \ldots(13)$$

By the same analysis as given in equations (1) to (7) setting the derivative with respect to $\underline{g}$ to zero gives an optimum gain of

$$\underline{g} = (\underline{Z}_{ij}^T \underline{Z}_{ij})^{-1} \underline{Z}_{ij}^T \underline{y} \qquad \ldots(14)$$

and substituting this into equation 13 gives an error of

$$e^2 = \underline{y}^T \underline{y} - \underline{y}^T \underline{Z}_{ij} (\underline{Z}_{ij}^T \underline{Z}_{ij})^{-1} \underline{Z}_{ij}^T \underline{y} \qquad \ldots(15)$$

And hence a need to choose the codebook entry to maximise:

$$\underline{y}^T \underline{Z}_{ij} (\underline{Z}_{ij}^T \underline{Z}_{ij})^{-1} \underline{Z}_{ij}^T \underline{y} \qquad \ldots(16)$$

This process is illustrated by the diagram of Figure 5 where a local decoder 220, having the structure shown in Figure 2, produces an error signal in a subtractor 221 for each trial i and a control unit 222 selects that entry (i.e. entry k) giving the best result. Note particularly that this process does not presuppose the previous optimum value g' assumed by the analysis unit 203. Rather, it assumes that g (and g1, g2 etc) assumes the optimum value for each of the candidate excitation entries.

The operation of the gain analysis unit 206, illustrated in Figure 6, is similar (similar components having reference numerals with a prime (') added), but involves a vector quantisation of the gains. That gain codeword G is selected for output which addresses that combination of gains from a gain codebook store 223 (also shown in Figure 3) which produces the smallest error e^2 from the subtractor 221'. The store 223 had the same contents as the decoder store 116 of Figure 2.

It should be noted that Figures 4, 5 and 6 are shown for illustrative purposes; in practice the derivations performed by the analysis units 203, 204, 206 may be more effectively performed by a suitably programmed digital signal processing (DSP) device. Flowcharts for the operation of such devices are presented in Figure 10.

Firstly, however we describe a number of measures which serve to reduce the complexity of the computation which needs to be carried out.

(a) Consider the product $\underline{Z}_{ij}^{T}\underline{Z}_{ij}$ of expression

(16) This is a 3 x 3 symmetric matrix:

$$\begin{bmatrix} |Z_{11}|^2 & Z_{11}^T Z_{12} & Z_{11}^T Z_D \\ Z_{12}^T Z_{11} & |Z_{12}|^2 & Z_{12}^T Z_D \\ Z_D^T Z_{11} & Z_D^T Z_{12} & |Z_D|^2 \end{bmatrix} \quad \ldots (17)$$

Each term of this is a product of the form $\underline{Z}_a^{T}\underline{Z}_b$ where a, b are any of i1, i2, D and can be written as

$$Z_a^T Z_b = (H C_a)^T H C_b = C_a^T H^T H C_b \quad \ldots (18)$$

A similar term is present also in expression (9) for the LTP search. $\underline{H}^T\underline{H}$ can be precalculated as it remains constant for the LTP and excitation search. In Figure 3 this calculation is shown as performed in a calculation unit 224 feeding both analysis units 203, 204. Note that the diagonals of the $\underline{H}^T\underline{H}$ matrix are the same sum with increasing limits, so that successive elements can be calculated by adding one term to an element already calculated. This is illustrated below with $\underline{H}$ shown as a 3 x 3 matrix, although in practice of course it would be larger: the size of $\underline{H}$ would be chosen to give a reasonable approximation to the conventionally infinite impulse response.

$$If\ H = \begin{bmatrix} h_1 & 0 & 0 \\ h_2 & h_1 & 0 \\ h_3 & h_2 & h_1 \end{bmatrix}$$

Then

$$H^TH = \begin{bmatrix} h_1 & h_2 & h_3 \\ 0 & h_1 & h_2 \\ 0 & 0 & h_1 \end{bmatrix} \begin{bmatrix} h_1 & 0 & 0 \\ h_2 & h_1 & 0 \\ h_3 & h_2 & h_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{3} h_i^2 & \sum_{i=1}^{2} h_i h_{i+1} & \sum_{i=1}^{1} h_i h_{i+2} \\ \sum_{i=1}^{2} h_i h_{i+1} & \sum_{i=1}^{2} h_i^2 & \sum_{i=1}^{1} h_i h_{i+1} \\ \sum_{i=1}^{1} h_i h_{i+2} & \sum_{i=1}^{1} h_i h_{i+1} & \sum_{i=1}^{1} h_i^2 \end{bmatrix}$$

from which it can be seen that each of the higher elements can be obtained by adding a further term to the element diagonally below it to the right.

Thus, if each term of the $\underline{H}^T\underline{H}$ matrix is H_{ij} (for the i'th row and j'th column) then, for example

$H_{23} = h_1 h_2$

$H_{12} = H_{23} + h_2 h_3$

Also since $H_{ij} = H_{ji}$ $(i \neq j)$ each of these pairs of terms can each be calculated only once and then multiplied by 2.

This process is further illustrated in the flowchart of Figure 7 where the terms H_{ij} (i = 1 ... N, j = 1 ... N), upwards on each diagonal D (D = 1 being the top right - hand corner of the matrix) are successively computed for each element after the lowest in position (for which the index I=0) by adding a further h.h. product term.

As $\underline{C}$ is ternary, finding $\underline{C}_1 \underline{H} \underline{H}^T \underline{C}_1^T$ (for example) from the $\underline{H}^T\underline{H}$ matrix simply amounts to selecting the appropriate elements from it (of appropriate sign) and adding them up.

This can be performed by means of a pointer table arrangement, using the modified apparatus shown in Figure 8. The elements of the $\underline{H}^T\underline{H}$ matrix, calculated by the unit 224, are stored in a store 301; or rather - in view of the symmetry of the matrix - the elements on the leading diagonal along with the elements above (or below) the

leading diagonal are stored. A second store 302 (in practice, part of the same physical store) stores the same elements but with negative values. Alongside the codebook store 205 is a pointer table 303 which stores, for each codebook entry, a list of the addresses of those locations within the stores 301,302 which contain the required elements. This process is illustrated schematically in figure 9 where the stores 301, 302, 303 are represented by rectangles and the contents by A_{11}, etc. (where A_{ij} is the j'th member of the address list for codeword i and H_{11} etc. are as defined above. The actual contents will be binary numbers representing the actual values of these quantities. The addresses are indicated by numbers external to the rectangles.

Suppose, by way of example, that the codeword no. 2 represents an excitation (-1,0,1,0,0,.....,0); then the desired elements of the $\underline{H}^T\underline{H}$ matrix are (+)H_{11}, (+)H_{33}, -H_{31}, -H_{13}. Therefore the relevant addresses are:

A_{21} = 1

A_{22} = 3

A_{23} = 1101

(A_{24} = 1101)

Thus codeword 2 addresses the pointer table 303; the addresses A_{21} etc. are read out and used to access the store 301/302; the contents thereby accessed are added together by an adder 304 to produce the required value $\underline{C}^T\underline{H}^T\underline{H}\underline{C}$. Since the elements off the leading diagonal always occur in pairs, in practice separate addresses would not be stored but the partial result multiplied by two (a simple binary shift) instead.

In a modification of this method, groups of excitations are shifted versions of one another; for example if excitation 3 is simply a one-place right-shift of excitation 2 (i.e. (0,-1,0,1..) in the above example, when

the desired elements are $+H_{22}$, $+H_{44}$, $-H_{24}$, $-H_{42}$ and the addresses are:

$A_{31} = 2$

$A_{32} = 4$

$A_{33} = 1102$

$(A_{34} = 1102)$

Therefore, to avoid a fresh look-up access to the pointer table 303 the addresses found for codeword 2 can be simply be modified to provide the new addresses for codeword 3. With the addressing scheme of figure 9 where elements of a diagonal of the matrix occupy locations with consecutive addresses this merely requires incrementing of all the addresses by one. This scheme fails if a pulse is lost (or needs to be gained) in the shift; whilst it may be possible to accommodate lost pulses by suppressing out-of-range addresses. A fresh access to the pointer table is then required for each new group.

Since this modification involves a loss of "randomness" of the excitations it may be wise to allow the pulses to take a wider range of values - i.e. discard the "ternary pulse" restriction. In this case each pointer table entry would contain, as well as a set of addresses, a set of $C_{ij} \cdot C_{ik}$ products ($\underline{C}_i = \{C_{i1}, C_{i2}, \ldots\}$) by which the retrieved H^TH elements would be multiplied. In Figure 8 this is provided for by the multipliers 305, 306 and the dotted connections from the pointer table.

In the case of the upper right-hand terms $\underline{Z}_{i1}{}^T\underline{Z}_0$ and $\underline{Z}_{i2}{}^T\underline{Z}_0$, these are equal to $\underline{C}_1^T H^T H \underline{C}_0$ and $\underline{C}_2{}^T H^T H \underline{C}_0$ respectively and since $\underline{C}_0$ is fixed for the codebook search $H^T H \underline{C}_0$ can be precalculated.

In the case of the $\| \underline{Z}_0 \|^2$ term, this is the term X^TX already computed in the analysis unit 203 for the selected delay and is obtained from the latter via a path 225 in Figures 3 and 8.

(b) Consider now the term

$$Z_{ij}^T y = \begin{bmatrix} Z_{i1}^T \\ Z_{i2}^T \\ Z_D^T \end{bmatrix} y = \begin{bmatrix} C_1^T H^T \\ C_2^T H^T \\ C_D^T H^T \end{bmatrix} y \qquad \ldots (19)$$

For $C_1^T H^T y$ and $C_2^T H^T y$, $H^T y$ is precalculated both for the expressions (9) and (19) in a unit 226 in Figures 3 and 8. $C_0^T H^T y$ is available from the LTP search (via the path 225).

(c) The term $y^T Z_{ij}$ is just the transpose of $Z_{ij}^T y$, of course.

(d) Consider now the term $H^T y$ (or its transpose $y^T H$), from (b) above. This is a cross-correlation between the target and impulse response H.

We note that the LPC filter is a recursive filter having an infinite impulse response. H is a 40 x 40 matrix representing an FIR approximation to this response. Evaluation of $H^T y$ involves typically 800 multiplications and this would be extremely onerous.

In order to explain the proposed method for evaluating this quantity, it is necessary to define a new mathematical notation.

If A is a pxq matrix, then A^R is a row and column mirror image of A.

eg

$$A^R = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} R = \begin{bmatrix} a_{23} & a_{22} & a_{21} \\ a_{13} & a_{12} & a_{11} \end{bmatrix}$$

It follows that

$A^R B = (AB^R)^R$

Consider now the vector $H^T y$. Since H is symmetric, $H^T = H^R$, so $H^T y = H^R y = (H\, y^R)^R$

$\underline{H}\,\underline{y}^R$ represents a 'time reversed' target signal $\underline{y}$ filtered by the response $\underline{h}$ and thus the correlation can be replaced by a convolution and implemented by a recursive filtering operation.

(e) Having discussed the individual parts of

$$\underline{y}^T Z_{ij} \left(Z_{ij}^T Z_{ij}\right)^{-1} Z_{ij}^T \underline{y}$$

we now require to find the maximum value of this expression. In order to avoid the division by the determinant of $Z_{ij}^T Z_{ij}$ required for finding the inverse, we compute, separately,

$$Num = \det\left[Z_{ij}^T Z_{ij}\right] \underline{y}^T Z_{ij} \left(Z_{ij}^T Z_{ij}\right)^{-1} Z_{ij}^T \underline{y}$$

$$Den = \det\left[Z_{ij}^T Z_{ij}\right]$$

The values Num_{max} and Den_{max} for the previous largest value are stored, and (default = 0, 1.). The test

$$\frac{Num}{Den} > \frac{Num_{max}}{Den_{max}} \; ?$$

is then performed as $Num.\ Den_{max} > Num_{max}.\ Den$?

In the modification discussed above employing excitations which are shifted versions of one another, the number of addresses that need to be retrieved from the pointer table store 303 is reduced, because addresses already retrieved can be modified. This presupposes that the codebook analysis unit 204 keeps all the addresses for a given codebook entry so that they are available to be modified for the next one, and therefore it will require local storage; for example if it is a digital signal processing chip its on-board registers may be used for this purpose. The number of addresses is $p(p+1)/2$ where p is the number of pulses in an excitation (assuming p is constant and truncation of $\underline{H}$ (see below) is not employed). If this exceeds the number of available registers, the problem can be alleviated by the use of "sub-vectors".

This proposal provides that each excitation of the codebook set is a concatenation of two (or more) partial excitations or sub-vectors belonging to a set of sub-vectors, viz:

$$\underline{C}_i = \begin{bmatrix} c_{i1} \\ c_{i2} \\ c_{iu} \end{bmatrix}$$

where $\underline{c}_{ij}$ is a sub-vector and u is the number of sub-vectors in an excitation. Necessarily each sub-vector occurs in a number of different excitations. The computation of $\underline{C}^T\underline{H}^T\underline{H}\underline{C}$ terms can then be partitioned into u^2 partial results each of which involves the multiplication of a sub-block of the $\underline{H}^T\underline{H}$ matrix by the two relevant partial excitations. If the sub-block is J_{rs} (r=1, ...u; s=1, ...u) so that:

$$H^TH = \begin{bmatrix} J_{11} & J_{12} & \cdots & J_{1u} \\ J_{21} & \cdots & \cdots & \cdot \\ J_{u1} & \cdots & \cdots & J_{uu} \end{bmatrix}$$

and

$$J_{rs} = \begin{bmatrix} H_{(r-1)\frac{N}{u}+1,\,(s-1)\frac{N}{u}+1} & H_{(r-1)\frac{N}{u}+1,\,(s-1)\frac{N}{u}+2} & \cdots & H_{(r-1)\frac{N}{u}+1,\,s\frac{N}{u}} \\ H_{r-1\frac{N}{u}+2,\,(s-1)\frac{N}{u}+1} & \cdot & \cdot\cdot & \cdot \\ \cdot & \cdot & \cdot\cdot & \cdot \\ H_{r\frac{N}{u},\,(s-1)\frac{N}{u}+1} & \cdot & \cdot\cdot & H_{r\frac{N}{u},\,s\frac{N}{u}} \end{bmatrix}$$

then the partial product is:

$$P_{r,s} = \underline{C}_{ir}\, \underline{J}_{rs}\, \underline{C}_{is}$$

and the final result is:

$$\underline{C}^T\underline{H}^T\underline{H}\underline{C} = \sum_{r=1}^{n}\sum_{s=1}^{n} P_{r,s}$$

In this scheme, the partial excitations $\underline{c}_{ij}$ (rather than the excitations $\underline{C}_i$) are shifted versions of one another (within a group thereof). The sequence of operations is modified so that all the partial products $P_{r,s}$ involving given values of r and s are performed consecutively and the addresses corresponding to that pair are then modified to obtain the addresses for the next pair (with additional address retrieval if either $\underline{c}_{ir}$ or $\underline{c}_{is}$ crosses a group boundary as i is incremented. Naturally there is an overhead in that the partial products need to be stored and, at the end of the process retrieved and combined to produce the final results.

As any given pair (of the same or different) sub-vectors in given positions r,s will occur in more than one $\underline{C}_i$, the relevant partial product can be formed and stored once and retrieved several times for the relevant excitations $\underline{C}_i$. (This is so whether or not "shifting" is used.

It is observed in practice that, since the later terms of the impulse response $\underline{h}$ tend to be fairly small, terms in the $\underline{H}^T\underline{H}$ matrix which relate to contributions from pulses of the excitation which are far apart - i.e. the terms in the upper right-hand corner and lower left-hand corner of $\underline{H}^T\underline{H}$ (as set out on page 11 above) - are also small and can be assumed zero with little loss of accuracy. This can readily be achieved by omitting the corresponding addresses from the pointer table 203 and, of course, by the analysis unit 204 not retrieving them. The same logic may be applied to systems using sub-vectors. Where it is desired, for simplicity of address retrieval, that the number of pulses, and the number of addresses per codebook entry are always the same, it may be convenient to omit those K

addresses (where K is the number desired to be omitted) which relate the furthest apart pulse pairs (or those pulse pairs which are furthest apart in terms of their number in the pulse sequence - as opposed to their positions in the frame -; a similar but not identical criterion). Where sub-vectors are used, then the proximity of a pulse in one sub-vector to pulses in an adjacent sub-vector needs to be considered; terms involving a pulse pair within the same sub-vector probably cannot be ignored. For example, if we suppose that there are three pulses per sub-vector we may assume that:

(a) terms involving the first pulse of the first sub-vector and the second or third pulse of the second sub-vector; and

(b) terms involving the second pulse of the first sub-vector and the third pulse of the second sub-vector may be ignored.

* C L M S *

CLAIMS

1. A method of speech coding in which input speech is analysed to determine the parameters of a synthesis filter and to select at least one excitation component from a plurality of possible components, including the step of determining the scalar product of the response of the filter to an excitation component and the response of the filter to the same or another excitation component, wherein the product of a filter response matrix $\underline{H}$ and its transpose $\underline{H}^T$ to form a product matrix $\underline{H}^T\underline{H}$ is formed once, and each scalar product is formed by multiplying the product matrix by the relevant possible excitation components, characterised in that for each set of diagonal terms of the product matrix, a first member of the set is calculated, and each further member of that set is obtained by adding a further term to the preceding member of the set.

2. A method according to claim 1 in which each term H_{ij} $(i=j)$ of the matrix is calculated and each pair of terms $H_{ij} = H_{ij}$ $(i \neq j)$ is calculated once only.

3. A speech coding apparatus comprising

(a) means for analysing an input speech signal to determine the parameters of a synthesis filter; and

(b) means for selecting at least one excitation component from a plurality of possible components by determining the scalar product of the response of the filter to an excitation component and the response of the filter to the same or another excitation component, including means for forming the product of a filter response matrix $\underline{H}$ and its transpose $\underline{H}^T$ to form a product matrix $\underline{H}^T\underline{H}$,

characterised in that the selecting means further includes

(c) a first store for storing elements of the product matrix $\underline{H}^T\underline{H}$;

* C L M S *

(d) a second store storing, for each possible pair of an excitation component and the same or another excitation component, the address of each location in the first store which contains an element of the product matrix which is to be multiplied by nonzero elements of both excitation components of the pair; and

(e) means operable to retrieve addresses from the second store, to retrieve the contents of the locations in the first store thereby addressed, and to add the retrieved contents.

4. A speech coding apparatus comprising

(a) means for analysing an input speech signal to determine the parameters of a synthesis filter; and

(b) means for selecting at least one excitation component from a plurality of possible components by determining the scalar product of the response of the filter to an excitation component and the response of the filter to the same or another excitation component, including means for forming the product of a filter response matrix $\underline{H}$ and its transpose $\underline{H}^T$ to form a product matrix $\underline{H}^T\underline{H}$; characterised in that the plurality of possible components consists of a plurality of subsets of components, each component of a subset being a shifted version of another member of the same subset, and that the selecting means includes

(c) a first store for storing elements of the product matrix $\underline{H}^T\underline{H}$;

(d) a second store storing, for one representative component of each subset of excitation components, the addresses of each location in the first store which contains an element of the product matrix which is to be multiplied by nonzero elements of the representative component;

(e) means operable to retrieve addresses from the second store, to modify the addresses in respect of components

other than the representative components, to retrieve the contents of the location in the first store thereby addressed, and to add the retrieved contents.

5. A speech coding apparatus comprising

(a) means for analysing an input speech signal to determine the parameters of a synthesis filter; and

(b) means for selecting at least one excitation component from a plurality of possible components by determining the scalar product of the response of the filter to an excitation component and the response of the filter to the same or another excitation component, including means for forming the product of a filter response matrix $\underline{H}$ and its transpose $\underline{H}^T$ to form a product matrix $\underline{H}^T\underline{H}$; characterised in that each of the plurality of possible components is formed by concatenation of a plurality of partial components from a set of partial components and that the selecting means includes

(c) a first store for storing elements of the product matrix $\underline{H}^T\underline{H}$;

(d) a second store storing, for the partial components, the addresses of locations in the first store which contain an element of the product matrix which is to be multiplied by nonzero elements of the partial component;

(e) means operable to retrieve addresses from the second store, to retrieve the contents of the location in the first store with the aid of those addresses, and to add the retrieved contents to form partial products, and to add partial products to produce the said scalar products.

6. An apparatus according to Claim 5 in which each set of partial components consists of a plurality of subsets of partial components, each partial component of a subset being a shifted version of another member of the same subset, in which the second store contains only addresses corresponding to a representative partial component of each

subset, and the retrieval means is operable to modify addresses in respect of components other than the representative components.

7. A speech coding apparatus substantially as herein described with reference to the accompanying drawings.