(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 267 327 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
18.12.2002 Bulletin 2002/51

(51) Int Cl.[7]: **G10L 19/06**

(21) Application number: **02005056.3**

(22) Date of filing: **06.03.2002**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **06.03.2001 US 800071**
26.10.2001 US 39528

(71) Applicant: **DoCoMo Communications Laboratories USA, Inc.**
San Jose, CA 95110 (US)

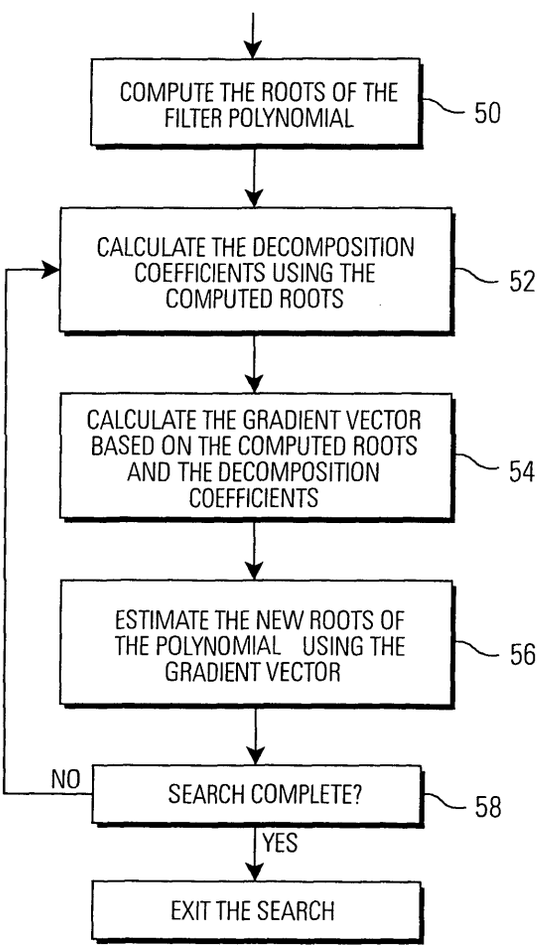
(72) Inventors:
- **Lashkari, Khosrow**
  Fremont, CA 94539 (US)
- **Miki, Toshio**
  Cupertino, CA 95014 (US)

(74) Representative: **HOFFMANN - EITLE**
Patent- und Rechtsanwälte
Arabellastrasse 4
81925 München (DE)

(54) Optimization of model parameters in speech coding

(57) A speech synthesis system is provided that optimizes a synthesis filter. Optimization is achieved by minimizing a synthesis error between the original speech sample and a synthesized speech sample. A gradient search algorithm in the root domain is also provided to aid minimization of the synthesis error.

FIG.3



EP 1 267 327 A2

Description

BACKGROUND

[0001] The present invention relates generally to speech encoding, and more particularly, to an encoder and a gradient search algorithm.

[0002] Speech compression is a well known technology for encoding speech into digital data for transmission to a receiver which then reproduces the speech. The digitally encoded speech data can also be stored in a variety of digital media between encoding and later decoding (i.e., reproduction) of the speech.

[0003] Speech synthesis systems differ from other analog and digital encoding systems that directly sample an acoustic sound at high bit rates and transmit the raw sampled data to the receiver. Direct sampling systems usually produce a high quality reproduction of the original acoustic sound and is typically preferred when quality reproduction is especially important. Common examples where direct sampling systems are usually used include music phonographs and cassette tapes (analog) and music compact discs and DVDs (digital). One disadvantage of direct sampling systems, however, is the large bandwidth required for transmission of the data and the large memory required for storage of the data. Thus, for example, in a typical encoding system which transmits raw speech sampled from the original acoustic sound, a data rate as high as 96,000 bits per second is often required.

[0004] In contrast, speech synthesis systems use a mathematical model of the human speech production. The fundamental techniques of speech modeling are known in the art and are described in B.S. Atal and Suzanne L. Hanauer, *Speech Analysis and Synthesis by Linear Prediction of the Speech Wave,* The Journal of the Acoustical Society of America 637-55 (vol. 50 1971). The model of human speech production used in speech synthesis systems is usually referred to as a source-filter model. Generally, this model includes an excitation signal that represents air flow produced by the vocal folds, and a synthesis filter that represents the vocal tract (i.e., the glottis, mouth, tongue, nasal cavities and lips). Therefore, the excitation signal acts as an input signal to the synthesis filter similar to the way the vocal folds produce air flow to the vocal tract. The synthesis filter then alters the excitation signal to represent the way the vocal tract manipulates the air flow from the vocal folds. Thus, the resulting synthesized speech signal becomes an approximate representation of the original speech.

[0005] One advantage of speech synthesis systems is that the bandwidth needed to transmit a digitized form of the original speech can be greatly reduced compared to direct sampling systems. Thus, by comparison, whereas direct sampling systems transmit raw acoustic data to describe the original sound, speech synthesis systems transmit only a limited amount of control data needed to recreate the mathematical speech model. As a result, a typical speech synthesis system can reduce the bandwidth needed to transmit speech to about 4,800 bits per second.

[0006] One problem with speech synthesis systems is that the quality of the reproduced speech is sometimes relatively poor compared to direct sampling systems. Most speech synthesis systems provide sufficient quality for the receiver to accurately perceive the content of the original speech. However, in some speech synthesis systems, the reproduced speech is not transparent. That is, while the receiver can understand the words originally spoken, the quality of the speech may be poor or annoying. Thus, a speech synthesis system that provides a more accurate speech production model is desirable.

BRIEF SUMMARY

[0007] Accordingly, a speech encoding system is provided for optimizing the mathematical model of human speech production. The speech synthesis system uses the LPC technique to compute coefficients of the synthesis filter. The synthesis filter is then optimized by minimizing the synthesis error between the original speech and the synthesized speech. To make minimization of the synthesis error easier, the LPC coefficients are converted into roots of the synthesis filter. A gradient search algorithm is then used to find the optimal roots. When the optimal roots are found, the roots are converted back into polynominal coefficients and are quantized for transmission.

[0008] This solution involves minimizing a synthesis error between an original speech sample and a synthesized speech sample. One difficulty that was discovered is that speech synthesis systems are of a highly nonlinear nature with respect to the synthesis error, which made the problem previously mathematically intractable. This difficulty is overcome by solving the problem using the roots of the synthesis filter polynomial instead of the coefficients of the polynomial.

[0009] Accordingly, a root searching algorithm is described for finding the roots of the synthesis filter polynomial.

[0010] In parametric speech coder that resolve the synthesis filter polynomial using roots instead of coefficients, the effectiveness and efficiency of the root searching algorithm used has an impact on the quality and performance of the speech coder. One root searching algorithm that may be used in such speech coders is a gradient search algorithm. Here, gradient search algorithms use an iterative solution process that calculates a gradient vector for a function and estimates the unknown variables using the calculated gradient vector.

[0011] According to a further aspect of the invention, an improved gradient search algorithm is provided. The improved algorithm recalculates the gradient vector by taking into account the variations of the decomposition coefficients with respect to the roots. Thus, the gradient search algorithm is especially useful with linear predictive coding speech systems that optimize synthesized speech by searching for roots of a polynomial.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0012] The invention, including its construction and method of operation, is illustrated more or less diagrammatically in the drawings, in which:

- Fig. 1 is a block diagram of a speech synthesis system;
- Fig. 2 is a flow chart of a speech synthesis system;
- Fig. 2B is a flow chart of an alternative speech synthesis system;
- Fig. 3 is a flow chart of a gradient search algorithm;
- Fig. 4 is a timeline-amplitude chart, comparing an original speech sample to an LPC synthesized speech and an optimally synthesized speech;
- Fig. 5 is a chart, showing synthesis error reduction and improvement as a result of the optimization;
- Fig. 6 is a spectral chart, comparing an original speech sample to an LPC synthesized speech and an optimally synthesized speech;
- Fig. 7 is another timeline-amplitude chart, comparing an original speech sample to an LPC synthesized speech and an optimally synthesized speech;
- Fig. 8 is another chart, showing synthesis error reduction and improvement as a result of the optimization; and
- Fig. 9 is another spectral chart, comparing an original speech sample to an LPC synthesized speech and an optimally synthesized speech.

DESCRIPTION

[0013] Referring now to the drawings, and particularly to Fig. 1, a speech synthesis system is provided that minimizes synthesis filter errors in order to more accurately model the original speech. In Fig. 1, a speech analysis-by-synthesis (AbS) system is shown which is commonly referred to as a source-filter model. As is well known in the art, source-filter models are designed to mathematically model human speech production. Typically, the model assumes that the human sound-producing mechanisms that produce speech remain fixed, or unchanged, during successive short time intervals (e.g., 20 to 30 ms). The model further assumes that the human sound producing mechanisms change after each interval or between successive intervals. The physical mechanisms modeled by this system include air pressure variations generated by the vocal folds, the glottis, the mouth, the tongue, the nasal cavities and the lips. Therefore, by limiting the digitally encoded data to a small set of control data for each interval, the speech decoder can reproduce the model and recreate the original speech. Thus, raw sampled data of the original speech is not transmitted from the encoder to the decoder. As a result, the digitally encoded data which is transmitted or stored (i.e., the bandwidth, or the number of bits) is much less than typical direct sampling systems require.

[0014] Accordingly, Fig. 1 shows an original digitized speech 10 delivered to an excitation module 12, thereby delivering an original speech sample s(n) to the excitation module 12. The excitation module 12 then analyzes each sample s(n) of the original speech and generates an excitation function u(n). The excitation function u(n) is typically a series of pulse signals that represent air bursts from the lungs which are released by the vocal folds to the vocal tract. Depending on the nature of the original speech sample s(n), the excitation function u(n) may be either a voiced 13, 14 or an unvoiced signal 15.

[0015] One way to improve the quality of reproduced speech in speech synthesis systems involves improving the accuracy of the voiced excitation function u(n). Traditionally, the excitation function u(n) has been treated as a preset series of pulses 13 with a fixed magnitude G and period P between the pitch pulses. As those in the art well know, the magnitude G and period P may vary between successive intervals. In contrast to the traditional fixed magnitude M and period P, it has previously been shown to the art that speech synthesis can be improved by optimizing the excitation function u(n) by varying the magnitude and pitch period of the excitation pulses 14. This improvement is described in Bishnu S. Atal and Joel R. Remde, *A New Model of LPC Excitation For Producing Natural-Sounding Speech At Low Bit Rates,* IEEE International Conference On Acoustics, Speech, And Signal Processing 614-17 (1982). This optimization technique usually requires more intensive computing to encode the original speech s(n), but this problem has not been a significant disadvantage since modern computers provide sufficient computing power for optimization 14 of the excitation function u(n). A greater problem with this improvement has been the additional bandwidth that is required to transmit data for the variable excitation pulses 14. One solution to this problem is a coding system that is described in Manfred R. Schroeder and Bishnu S. Atal, *Code-Excited Linear Prediction (CELP): High-Quality Speech*

At Very Low Bit Rates, IEEE International Conference On Acoustics, Speech, And Signal Processing 937-40 (1985). This solution involves categorizing a number of optimized excitation functions into a library of functions, or a codebook. The encoding excitation module 12 will then select an optimized excitation function from the codebook that produces a synthesized speech that most closely matches the original speech s(n). Then, a code that identifies the optimum codebook entry is transmitted to the decoder. When the decoder receives the transmitted code, the decoder then accesses a corresponding codebook to reproduce the selected optimal excitation function u(n).

[0016] The excitation module 12 can also generate an unvoiced 15 excitation function u(n). An unvoiced 15 excitation function u(n) is used when the speaker's vocal folds are open and turbulent air flow is produced through the vocal tract. Most excitation modules 12 model this state by generating an excitation function u(n) consisting of white noise 15 (i. e., a random signal) instead of pulses.

[0017] Next, the synthesis filter 16 models the vocal tract and its effect on the air flow from the vocal folds. Typically, the synthesis filter 16 uses a polynomial equation to represent the various shapes of the vocal tract. This technique can be visualized by imagining a multiple section hollow tube with a number of different diameters along the length of the tube. Accordingly, the synthesis filter 16 alters the characteristics of the excitation function u(n) similar to the way the vocal tract alters the air flow from the vocal folds, or like a variable diameter hollow tube alters inflowing air.

[0018] According to Atal and Remde, *supra.,* the synthesis filter 16 can be represented by the mathematical formula:

$$H(z) = G/A(z) \tag{1}$$

where G is a gain term representing the loudness of the voice. A(z) is a polynomial of order M and can be represented by the formula:

$$A(z) = 1 + \sum_{k=1}^{M} a_k z^{-k} \tag{2}$$

[0019] The order of the polynomial A(z) can vary depending on the particular application, but a 10th order polynomial is commonly used with an 8 kHz sampling rate. The relationship of the synthesized speech s(n) to the excitation function u(n) as determined by the synthesis filter 16 can be defined by the formula:

$$\hat{s}(n) = Gu(n) - \sum_{k=1}^{M} a_k \hat{s}(n-k) \tag{3}$$

[0020] Typically, the coefficients $a_1 \ldots a_m$ of this polynomial have been computed using a technique known in the art as linear predictive coding (LPC). LPC-based techniques compute the polynomial coefficients $a_1 \ldots a_M$ by minimizing the total prediction error e_p. Accordingly, the sample prediction error $e_p(n)$ is defined by the formula:

$$e_p(n) = s(n) + \sum_{k=1}^{M} a_k s(n-k) \tag{4}$$

The total prediction error E_p is then defined by the formula:

$$E_p = \sum_{k=0}^{N-1} e_p^{\,2}(k) \tag{5}$$

where N is the length of the analysis window in number of samples. The polynomial coefficients $a_1 \ldots a_M$ can now be resolved by minimizing the total prediction error E_p using well known mathematical techniques.

[0021] One problem with the LPC technique of resolving the polynomial coefficients $a_1 \ldots a_M$ is that only the prediction

error is minimized. Thus, the LPC technique does not minimize the error between the original speech s(n) and the synthesized speech $\hat{s}(n)$. Accordingly, the sample synthesis error $e_s(n)$ can be defined by the formula:

$$e_s(n) = s(n) - \hat{s}(n) \tag{6}$$

The total synthesis error E_s can then be defined by the formula:

$$E_s = \sum_{n=0}^{N-1} e_s^2(n) = \sum_{n=0}^{N-1} (s(n) - \hat{s}(n))^2 \tag{7}$$

where N is the length of the analysis window. Like the total prediction error E_p discussed above, the total synthesis error E_s should be minimized to resolve the optimum filter coefficients $a_1 \ldots a_M$. However, one difficulty with this technique is that the synthesized speech $\hat{s}(n)$ as represented in formula (3) makes the total synthesis error E_s a highly nonlinear function that is generally mathematically intractable.

[0022] One solution to this mathematical difficulty is to minimize the total synthesis error E_s using the roots of the polynomial A(z) instead of the coefficients $a_1 \ldots a_M$. Using roots instead of coefficients for optimization also provides control over the stability of the synthesis filter 16. Accordinqly, assuming that h(n) is the impulse response of the synthesis filter 16 the synthesized speech $\hat{s}(n)$ is now defined by the formula:

$$\hat{s}(n) = h(n) * u(n) = \sum_{k=0}^{n} h(k)u(n-k) \tag{8}$$

where * is the convolution operator. In this formula, it is also assumed that the excitation function u(n) is zero outside of the interval 0 to N-1. Using the roots of A(z), the polynomial can now be expressed by the formula:

$$A(z) = (1-\lambda_1 z^{-1}) \quad (1-\lambda_M z^{-1}) \tag{9}$$

where $\lambda_1 \ldots \lambda_M$ represent the roots of the polynomial A(z). These roots may be either real or complex. Thus, in the preferred 10th order polynomial, A(z) will have 10 different roots.

[0023] Using parallel decomposition, the synthesis filter function H(z) is now represented in terms of the roots by the formula:

$$H(z) = G/A(z) = \sum_{i=1}^{M} b_i/(1-\lambda_i z^{-1}) \tag{10}$$

(the gain term G is omitted from this and the remaining formulas for simplicity).

[0024] The decomposition coefficients b_i are then calculated by the residue method for polynomials, thus providing the formula:

$$b_i = G \prod_{j=1,\ j\neq i}^{M} (1/(1-\lambda_j \lambda_i^{-1})) \tag{11}$$

[0025] The impulse response h(n) can also be represented in terms of the roots by the formula:

$$h(n) = \sum_{i=1}^{M} b_i(\lambda_i)^n \qquad (12)$$

[0026] Next, by combining formula (12) with formula (8), the synthesized speech $\hat{s}(n)$ can be expressed by the formula:

$$\hat{s}(n) = \sum_{k=0}^{n} h(k)u(n-k) = \sum_{k=0}^{n} u(n-k) \sum_{i=1}^{M} b_i(\lambda_i)^k \qquad (13)$$

Therefore, by substituting formula (13) into formula (7), the total synthesis error E_s can be minimized using polynomial roots and a gradient search algorithm.

[0027] A number of root searching algorithms may be used to minimize the total synthesis error E_s. One possible algorithm, however, is an iterative gradient search algorithm. Accordingly, denoting the root vector at the j-th iteration as $\Lambda^{(j)}$, the root vector can be expressed by the formula:

$$\Lambda^{(j)} = [\lambda_1^{(j)} \ldots \lambda_i^{(j)} \ldots \lambda_M^{(j)}]^T \qquad (14)$$

where $\lambda_i^{(j)}$ is the value of the i-th root at the j-th iteration and T is the transpose operator. The search algorithm begins with the LPC solution as the starting point, which is expressed by the formula:

$$\Lambda^{(0)} = [\lambda_1^{(0)} \ldots \lambda_i^{(0)} \ldots \lambda_M^{(0)}]^T \qquad (15)$$

To compute $\Lambda^{(0)}$, the LPC coefficients $a_1 \ldots a_M$ are converted to the corresponding roots $\lambda_1^{(0)} \ldots \lambda_M^{(0)}$ using a standard root finding algorithm.

[0028] Next, the roots at subsequent iterations can be expressed by the formula:

$$\Lambda^{(j+1)} = \Lambda^{(j)} + \mu \nabla_j E_s \qquad (16)$$

where μ is the step size and $\nabla_j E_s$ is the gradient of the synthesis error E_s relative to the roots at iteraton j. The step size μ can be either fixed for each iteration, or alternatively, it can be variable and adapted for each iteration. Using formula (7), the synthesis error gradient vector $\nabla_j E_s$ can now be calculated by the formula:

$$\nabla_j E_s = \sum_{k=1}^{N-1} (s(k) - \hat{s}(k)) \nabla_j \hat{s}(k) \qquad (17)$$

[0029] Formula (17) demonstrates that the synthesis error gradient vector $\nabla_j E_s$ can be calculated using the gradient vector of the synthesized speech samples $\hat{s}(k)$. Accordingly, the synthesized speech gradient vector $\nabla_j \hat{s}(k)$ can be defined by the formula:

$$\nabla_j \hat{s}(k) = [\partial \hat{s}(k)/\partial \lambda_1^{(j)} \quad \partial \hat{s}(k)/\partial \lambda_i^{(j)} \quad \partial \hat{s}(k)/\partial \lambda_M^{(j)}] \qquad (18)$$

where $\partial \hat{s}(k)/\partial \lambda_1^{(j)}$ is the partial derivative of $\hat{s}(k)$ at iteration j with respect to the i-th root. Using formula (13) and assuming constant decomposition coefficients during successive iterations of the gradient search, the partial derivatives can be calculated by the formula:

$$\partial\hat{s}(k)/\partial\lambda_1^{(j)} = b_i \sum_{m=1}^{k} mu(k-m)(\lambda_1^{(j)})^{(m-1)} \quad k \geq 1 \qquad (19A)$$

where $\partial\hat{s}(0)/\partial\lambda_1^{(j)}$ is always zero.

[0030] Further, using formula (13) and for considering varying decomposition coefficients during successive iterations of a gradient search, the partial derivative $\partial s(k)/\partial\lambda_r^{(j)}$ can be calculated by the formula:

$$\partial\hat{s}(k)/\partial\lambda_r = \sum_{m=0}^{k}\sum_{i=1}^{M} u(k-m)\partial\left[b_i\lambda_i^m\right]/\partial\lambda_r \qquad (19B)$$

(the superscript j is omitted from formula (19B) through formula (28) for notational simplicity). Formula (19B) can now be expressed using the chain rule of differentiation by the formula:

$$\partial[b_i\lambda_i^m]/\partial\lambda_r = \lambda_i^m\, b_i/\lambda_r + mb_i\lambda_r^{(m-1)}\delta(r-i) \qquad (20)$$

where $\delta(r-i)$ is the delta function (i.e., $\delta(r-i)=1$ for $r=i$ and $\delta(r-i)=0$ for $r\neq i$).

[0031] To resolve formula (20), the partial derivative $\partial bi/\partial\lambda_r$ must be calculated. Therefore, formula (11) can be substituted into the partial derivative $\partial bi/\partial\lambda_r$ to provide the formula:

$$\partial b_i/\partial\lambda_r = \left\{\prod_{j=1,j\neq i}^{M}[1/(1-\lambda_j\lambda_i^{-1})]\right\}/\lambda_r \qquad (21)$$

To resolve the partial derivative of formula (21), the partial derivative must be calculated for two cases, including $r\neq i$ and $r=i$.

[0032] In the first case of formula (20), where $r\neq i$, only one multiplicative term of $1/(1-\lambda_r\lambda_l^{-1})$, which corresponds to $j=r$, depends on λ_r. Therefore, the partial derivative of formula (21) can be expressed by the formula:

$$\left\{\prod_{j=1,j\neq i}^{M}[1/(1-\lambda_j\lambda_i^{-1})]\right\}/\lambda_r = \left\{\prod_{j=1,j\neq l,j\neq r}^{M}[1/(1-\lambda_j\lambda_i^{-1})]\right\}[1/(1-\lambda_r\lambda_i^{-1})]/\lambda_r \quad (r\neq i) \qquad (22a)$$

Next, the partial derivative of $1/(1-\lambda_r\lambda_i^{-1})$ can be calculated by the formula:

$$[1/(1-\lambda_r\lambda_l^{-1})]/\lambda_r = \lambda_l/(\lambda_l-\lambda_r)^2 \qquad (22b)$$

By substituting formula (22b) Into formula (22a) and simplifying, formula (22a) can be expressed by the formula:

$$\left\{\prod_{j=1,j\neq i}^{M}[1/(1-\lambda_j\lambda_i^{-1})]\right\}/\lambda_r = b_i/(\lambda_i-\lambda_r) \quad (r\neq i) \qquad (22c)$$

By substituting formula (22c) into formula (21) and further simplifying, the partial derivative of $\partial b_i/\partial\lambda_r$ for the case of r i can now be expressed by the formula:

$$\partial b_i/\partial\lambda_r = (b_i/\lambda_i)[1/(1-\lambda_r\lambda_i^{-1})] \qquad (r\neq i) \qquad (22d)$$

In the second case of formula (21) where r=i, all of the M-1 multiplicative terms of

$$1/(1-\lambda_j\lambda_i^{-1})$$

depend on λ_i. Therefore, the partial derivative of formula (21) can be calculated as the sum of the M-1 contributions to the partial derivative. Thus, using the q-th multiplicative term (i.e., $1/(1-\lambda_q\lambda_i^{-1})$), the contribution to the partial derivative due to this term alone can be expressed by the formula:

$$\left\{\prod_{j=1,j\neq i,j\neq q}^{M}[1/(1-\lambda_j\lambda_i^{-1})]\right\}\ [1/(1-\lambda_q\lambda_i^{-1})]/\ \lambda_i \qquad (r=i) \qquad (23a)$$

Next, the partial derivative of $1/(1-\lambda_q\lambda_i^{-1})$ can be calculated by the formula:

$$\partial[1/(1-\lambda_q\lambda_i^{-1})]/\partial\lambda_j = -\lambda_q/(\lambda_i-\lambda_q)^2 \qquad (23b)$$

By substituting formula (23b) into formula (23a) and simplifying, formula (23a) can be expressed by the formula:

$$\left\{\prod_{j=1,j\neq i,j\neq q}^{M}[1/(1-\lambda_j\lambda_i^{-1})]\right\}\ [1/(1-\lambda_q\lambda_i^{-1})]/\ \lambda_i = b_i/\lambda_i(1-\lambda_i\lambda_q^{-1}) \qquad (23c)$$

Using formula (23c) to add up all of the contributions In formula (23a) and then substituting the result into formula (21) and further simplifying, the partial derivative of

$$\partial b_i/\partial\lambda_r$$

$\partial b_i/\partial\lambda_r$ for the case of r=i can now be expressed by the formula:

$$\partial b_i/\partial\lambda_r = (b_i/\lambda_i)\sum_{j=1,j\neq i}^{j=M}[1/(1-\lambda_i\lambda_j^{-1})] \qquad (r=i) \qquad (23d)$$

[0033] In order to unify the two cases of r≠i and r=i, the function K(i,r) can be defined by the following formulas:

$$K(i,r) = 1/(1-\lambda_r\lambda_i^{-1}) \qquad (\text{if } r\neq i) \qquad (24a)$$

$$K(i,r) = \sum_{j=1,j\neq i}^{M}[1/(1-\lambda_i\lambda_j^{-1})] \qquad (\text{if } r=i) \qquad (24b)$$

The partial derivative of

$$\partial b_i/\partial\lambda_r$$

can now be simplified for both cases by the formula:

$$\partial b_i/\partial \lambda_r = b_i K(i,r)/\lambda_i \qquad \text{(for any r)} \tag{25}$$

[0034] By substituting formula (25) Into formula (20), the partial derivative of $\partial[b_i\lambda_i^m]/\partial\ \lambda_r$ can now be expressed by the formula:

$$\partial[b_i\lambda_i^m]/\partial\lambda_r = b_i[K(i,r)\lambda_i^{(m-1)} + m\lambda_r^{(m-1)}\delta(r-i)] \tag{26}$$

In formula (26), the first term of the formula (i.e., $K(i,r)\lambda_i^{(m-1)}$)is the contribution of $\partial\ b_i/\partial\lambda_i$, while the second term of the formula (i.e., $m\lambda_r^{(m-1)}\delta(r-i)$) is the contribution of the m-th power of λ_i.

[0035] By substituting formula (26) into formula (19), the partial derivative of the k-th sample of the synthesized speech with respect to the r-th root can be expressed by the formula:

$$\partial \hat{s}(k)/\partial\lambda_r = \sum_{m=0}^{k} u(k-m)\sum_{i=1}^{i=M} b_i\left[K(i,r)\lambda_i^{(m-1)} + m\lambda_r^{(m-1)}\delta(r-i)\right] \tag{27}$$

By simplifying formula (27), the partial derivative can be expressed by the formula:

$$\partial \hat{s}(k)/\partial\lambda_r = \sum_{m=0}^{k} u(k-m)\sum_{i=1}^{i=M} b_i K(i,r)\lambda_i^{(m-1)} + b_r\sum_{m=1}^{k} mu(k-m)\lambda_r^{(m-1)} \tag{28}$$

For completeness, the Iteration index can be inserted back into formula (28) to express the partial derivative of the synthesized speech at iteration j by the formula:

$$\partial \hat{s}(k)/\partial\lambda_r^{(j)} = \sum_{m=0}^{k} u(k-m)\sum_{i=1}^{i=M} b_i K(i,r)(\lambda_i^{(j)})^{(m-1)} + b_r\sum_{m=1}^{k} mu(k-m)(\lambda_r^{(j)})^{(m-1)} \quad (k>0) \tag{29}$$

[0036] The synthesis error gradient vector $\nabla_j E_s$ is now calculated by substituting formula (29) into formula (18) and formula (18) into formula (17). The subsequent root vector $\Lambda^{(j+1)}$ at the next iteration can then be calculated by substituting the result of formula (17) into formula (16). The iterations of the gradient search algorithm are then repeated until either the synthesis error E_s is reduced by a desired percentage from the LPC prediction error E_p, a predetermined number of iterations are completed, or the roots are resolved within a predetermined acceptable range.

[0037] The synthesis error gradient vector $\nabla_j E_s$ is now calculated by substituting formula (19A) (for constant decomposition coefficients) or formula (29) (for varying decomposition coefficients) into formula (18) and formula (18) into formula (17).

The subsequent root vector $\Lambda^{(j)}$ at the next iteration can then be calculated by substituting the result of formula (17) into formula (16). The iterations of the gradient search algorithm are then repeated until either the synthesis error gradient vector $\nabla_j E_s$ is reduced to a predetermined acceptable range - e.g., the synthesis error E_S is reduced by a desired percentage from the LPC prediction error E_p -, a predetermined number of iterations are completed, or the roots are resolved within a predetermined acceptable range.

[0038] Although control data for the optimal synthesis polynomial A(z) can be transmitted in a number of different formats, it is preferable to convert the roots found by the optimization technique described above back into polynomial coefficients $a_1 \ldots a_M$. The conversion can be performed by well known mathematical techniques. This conversion allows the optimized synthesis polynomial A(z) to be transmitted in the same format as in the existing speech encoding, thus promoting compatibility with current standards.

[0039] Now that the synthesis model has been completely determined, the control data for the model is quantized into digital data for transmission or storage. Many different industry standards exist for quantization. However, in one example, the control data that is quantized includes ten synthesis filter coefficients $a_1 \ldots a_{10}$, one gain value G for the

magnitude of the excitation function pulses, one pitch period value P for the frequency of the excitation function pulses, and one indicator for a voiced 13 or unvoiced 15 excitation function u(n). As is apparent, this example does not include an optimized excitation pulse 14, which could be included with some additional control data. Accordingly, the described example requires the transmission of thirteen different variables at the end of each speech frame. Commonly, the thirteen variables are quantized into a total of 80 bits. Thus, according to this example, the synthesized speech s(n), including optimization, can be transmitted within a bandwidth of 4,000 bits/s (80 bits/frame ÷ .020 s/frame).

[0040] As shown in both Figure 1 and 2, the order of operations can be changed depending on the accuracy desired and the computing capacity available. Thus, in the embodiment described above, the excitation function u(n) was first determined to be a preset series of pulses 13 for voiced speech or an unvoiced signal 15. Second, the synthesis filter polynomial A(z) was determined using conventional techniques, such as the LPC method. Third, the synthesis polynomial A(z) was optimized.

[0041] In Figures 2A and 2B, different encoding sequences are shown which should provide more accurate synthesis and may be used with CELP-type speech encoders. However, some additional computing power will be needed. In this sequence, the original digitized speech sample 30 is used to compute 32 the polynomial coefficients $a_1 \ldots a_M$ using the LPC technique described above or another comparable method. The polynomial coefficients $a_1 \ldots a_M$, are then used to find 36 the optimum excitation function u(n) from a codebook. Alternatively, an individual excitation function u (n) can be found 40 from the codebook for each frame. After selection of the excitation function u(n), the polynomial coefficients $a_1 \ldots a_M$ are then also optimized. To make optimization of the coefficients $a_1 \ldots a_M$ easier, the polynomial coefficients $a_1 \ldots a_M$ are first converted 34 to the roots of the polynomial A(z). A gradient search algorithm is then used to optimize 38, 42, 44 the roots. Once the optimal roots are found, the roots are then converted 46 back to polynomial coefficients $a_1 \ldots a_M$ for compatibility with existing encoding-decoding systems. Lastly, the synthesis model and the index to the codebook entry is quantized 48 for transmission or storage.

[0042] In Figure 3, a flow chart of the gradient search algorithm is shown. After the polynominal coefficients $a_1 \ldots a_m$ have been converted to roots 34, first roots of the polynominal are computed 50. The initial roots may be determined by several methods, including root finding algorithms such as Newton-Raphson or interval halving. Decomposition coefficients b_l are then calculated using the first computed roots 52. Next, the gradient vector of the polynominal is calculated using the contribution of the decomposition coefficients b_i 54. Once the gradient vector is calculated for the first computed roots, the gradient vector is used to calculate second estimated roots 56. A test is then performed to determine whether the search should end or whether it should continue 58. Several tests may be used, induding testing whether the LPC prediction error Ep has been reduced by a desired percentage, whether a limited number of iterations has been completed, or whether the estimated roots are within an acceptable range. If the search is determined to be complete, the gradient search algorithm stops and the estimated roots are passed on to the speech synthesis system for further processing 58. On the other hand, if the search is not determined to be complete, the decomposition coefficients b_i are recalculated using the second estimated roots 52. The process of calculating the gradient vector and re-estimating the roots is then repeated using the new contribution of the recalculated decomposition coefficients b_i 54, 56.

[0043] Additional encoding sequences are also possible for improving the accuracy of the synthesis model or for changing the computing capacity needed to encode the synthesis model. Some of these altemative sequences are demonstrated in Figure 1 by dashed routing lines. For example, the excitation function u(n) can be reoptimized at various stages during encoding of the synthesis model.

[0044] Figures 4-6 show the improved results provided by the optimized speech synthesis system assuming constant decomposition coefficients. The figures show several different comparisons between a prior art LPC synthesis system and the optimized synthesis system. The speech sample used for this comparison is a segment of a voiced part of the nasal "m". In Figure 4, a timeline-amplitude chart of the original speech, a prior art LPC synthesized speech and the optimized synthesized speech is shown. As can be seen, the optimally synthesized speech matches the original speech much closer than the LPC synthesized speech.

[0045] In Figure 5, the reduction in the synthesis error is shown for successive iterations of optimization. At the first iteration, the synthesis error equals the LPC synthesis error since the LPC coefficients serve as the starting point for the optimization. Thus, the improvement in the synthesis error is zero at the first iteration. Accordingly, the synthesis error steadily decreases with each iteration. Noticeably, the synthesis error increases (and the improvement decreases) at iteration number three. This characteristic occurs when the root searching algorithm overshoots the optimal roots. After overshooting the optimal roots, the search algorithm can be expected to take the overshoot into account in successive iterations, thereby resulting in further reductions in the synthesis error. In the example shown, the synthesis error can be seen to be reduced by 37% after six iterations. Thus, a significant improvement over the LPC synthesis error is possible with the optimization.

[0046] Figure 6 shows a spectral chart of the original speech, the LPC synthesized speech and the optimized synthesized speech. The first spectral peak of the original speech can be seen in this chart at a frequency of about 280 Hz. Accordingly, the optimized synthesized speech matches the spectral peak of the original speech at 280 Hz much closer than the LPC synthesized speech.

[0047] Further improvements of the gradient search algorithm will become apparent when decomposition coefficients are not assumed to be constant during successive iterations of the gradient search. While this assumption provides acceptable results for some applications, improved results are achieved by the gradient search algorithm with variations in the decomposition coefficients that occur during successive iterations when calculating the gradient vector.

[0048] Figs. 7-9 show the improved results achieved in a speech synthesis system optimized according to such further improved gradient search algorithms. The figures show several different comparisons between a prior art LPC synthesis system and the optimized synthesis system. The speech sample used for this comparison is a segment of a voiced part of the nasal "m". In Fig. 7, a timeline-amplitude chart of the original speech, a prior art LPC synthesized speech and the optimized synthesized speech is shown. As can be seen, the optimally synthesized speech matches the original speech much closer than the LPC synthesized speech.

[0049] In Figure 8, the reduction in the synthesis error is shown for successive iterations of optimization. At the first iteration, the synthesis error equals the LPC synthesis error since the LPC coefficients serve as the starting point for the optimization. Thus, the improvement in the synthesis error is zero at the first iteration. Accordingly, the synthesis error steadily decreases with each Iteration. Noticeably, the synthesis error increases (and the improvement decreases) at iteration number three. This characteristic occurs when the root searching algorithm overshoots the optimal roots. After overshooting the optimal roots, the search algorithm can be expected to take the overshoot into account in successive iterations, thereby resulting in further reductions in the synthesis error. In the example shown, the synthesis error can be seen to be reduced by 59% after six iterations. Thus, a significant improvement over the LPC synthesis error is possible with the optimization.

[0050] Figure 9 shows a spectral chart of the original speech, the LPC synthesized speech and the optimized synthesized speech. As seen in this chart, the spectrum of the optimized speech provides a much better match to the spectrum of the original speech as compared to the LPC spectrum. The improvement in the synthesized spectrum is especially apparent in the frequency range of 0 to 1,500 Hz.

[0051] While preferred embodiments of the invention have been described, it should be understood that the invention is not so limited, and modifications may be made without departing from the invention. The scope of the invention is defined by the appended claims, and all devices that come within the meaning of the claims, either literally or by equivalence, are intended to be embraced therein.

Claims

1. A speech synthesis system for encoding original speech comprising an excitation module responsive to an original speech sample and generating an excitation function; a synthesis filter responsive to said excitation function and said original speech sample and generating a synthesized speech sample; and a synthesis filter optimizer responsive to said excitation function and said synthesis filter and generating an optimized synthesized speech sample; wherein said synthesis filter optimizer minimizes a synthesis error between said original speech sample and said synthesized speech sample.

2. The speech synthesis system according to Claim 1, wherein said synthesis filter optimizer comprises a root optimization algorithm, thereby making possible said minimization of said synthesis error.

3. The speech synthesis system according to Claim 2, wherein said synthesis filter comprises a predictive coding technique producing said synthesized speech sample from said original speech sample.

4. The speech synthesis system according to Claim 3, wherein said predictive coding technique produces first coefficients of a polynomial; wherein said root optimization algorithm is an iterative algorithm using first roots derived from said first coefficients in a first iteration; and wherein said root optimization algorithm produces second roots in successive iterations resulting in a reduction of said synthesis error compared to said successive iterations.

5. The speech synthesis system according to Claim 4, wherein said synthesis filter optimizer is adapted to convert said second roots to second coefficients of said polynomial.

6. The speech synthesis system according to Claim 1, wherein said excitation module is adapted to generate an excitation function with pulses of varying magnitude and period for voiced and unvoiced portions of said original speech sample.

7. The speech synthesis system according to Claim 6, wherein said excitation module is adapted to regenerate said excitation function after said synthesis filter optimizer generates said optimized synthesized speech sample, there-

by further optimizing said synthesized speech sample.

8. The speech synthesis system according to Claim 7, wherein said synthesis filter is adapted to regenerate said synthesized speech sample after said synthesis filter optimizer generates said optimized synthesized speech sample, thereby further optimizing said synthesized speech sample.

9. The speech synthesis system according to Claim 1, further comprising a quantizer digitally encoding said synthesized speech sample for transmission or storage after generation of said optimized synthesized speech sample.

10. The speech synthesis system according to Claim 1, wherein said synthesis filter optimizer comprises an root optimization algorithm, thereby simplifying said minimization of said synthesis error; wherein said synthesis filter comprises a predictive coding technique producing said synthesized speech sample from said original speech sample; wherein said predictive coding technique produces first coefficients of a polynomial, wherein said root optimization algorithm is an iterative algorithm using first roots derived from said first coefficients in a first iteration, and wherein said root optimization algorithm produces second roots in successive iterations resulting in a reduction of said synthesis error compared to said first iteration; wherein said synthesis filter optimizer is adapted to convert said second roots to second coefficients of said polynomial; wherein said excitation module is adapted to regenerate said excitation function after said synthesis filter optimizer generates said optimized synthesized speech sample, thereby further optimizing said synthesized speech sample; wherein said synthesis filter is adapted to regenerate said synthesized speech sample after said synthesis filter optimizer generates said optimized synthesized speech sample, thereby further optimizing said synthesized speech sample; and further comprising a quantizer digitally encoding said synthesized speech sample for transmission or storage after generation of said optimized synthesized speech sample.

11. A method of generating a speech synthesis filter representative of a vocal tract comprising computing first coefficients of a speech synthesis polynomial using an original speech sample, thereby producing a first synthesized speech sample; converting said first coefficients of said polynomial to first roots; and computing second roots, thereby producing a second synthesized speech sample more representative of said original speech sample than said first synthesized speech sample.

12. The method according to Claim 11, further comprising computing a first synthesis error between said original speech and said first synthesized speech sample; and computing a second synthesis error between said original speech and said second synthesized speech; wherein said second synthesis error is less than said first synthesis error.

13. The method according to Claim 12, wherein said computing of said second roots comprises iteratively searching for said second roots using the gradient of said first synthesized speech sample.

14. The method according to Claim 13, wherein said computing of said first coefficients comprises minimizing a prediction error of said original speech sample using a linear predictive coding technique.

15. The method according to Claim 14, further comprising converting said second roots into second coefficients of said polynomial.

16. A method for digitally encoding speech comprising means for producing an original speech sample, means for generating an excitation function; means for computing LPC polynomial coefficients, thereby producing a synthesized speech sample; means for optimizing said polynomial coefficients by minimizing a synthesis error between said original speech sample and said synthesized speech sample.

17. The method according to Claim 16, wherein said means for optimizing comprises means for converting said LPC coefficients to first roots and iteratively searching for second roots.

18. The method according to Claim 17, wherein said means for iteratively searching comprises means for calculating the gradient of said synthesized speech sample.

19. The method according to Claim 18, further comprising reoptimizing said excitation function after said means for computing LPC polynomial coefficients.

20. The method according to Claim 16, further comprising recomputing said polynomial coefficients after said means for optimizing said polynomial coefficients.

21. A gradient search algorithm for a speech coding system, comprising calculating a gradient vector; and calculating a contribution to said gradient vector in response to variations in decomposition coefficients.

22. The gradient search algorithm according to Claim 21, used in combination with finding roots of a speech synthesis polynomial, wherein said gradient search algorithm further comprises iteratively calculating said gradient vector and recalculating said contribution at each iteration, whereby said decomposition coefficients vary between Iterations.

23. The gradient search algorithm according to Claim 22, wherein one of said decomposition coefficients corresponds to each of a plurality of said roots.

24. The gradient search algorithm according to Claim 23, wherein said gradient vector and said contribution to said gradient vector are calculated using the formula:

$$\partial\hat{s}(k)/\partial\lambda_r^{(j)} = \sum_{m=0}^{k} u(k-m) \sum_{i=1}^{i=M} b_i K(i,r)(\lambda_i^{(j)})^{(m-1)} + b_r \sum_{m=1}^{k} . mu(k-m)(\lambda_r^{(j)})^{(m-1)} \ (k>0).$$

25. The gradient search algorithm according to Claim 21, used in combination with a speech coding system for encoding original speech, the speech coding system comprising an excitation module responsive to an original speech sample and generating an excitation function; a synthesis filter responsive to said excitation function and said original speech sample and generating a synthesized speech sample; and a synthesis filter optimizer responsive to said excitation function and said synthesis filter and generating an optimized synthesized speech sample; wherein said synthesis filter optimizer minimizes a synthesis error between said original speech sample and said synthesized speech sample; wherein the gradient search algorithm Is used by said synthesis filter optimizer.

26. The gradient search algorithm according to Claim 25, wherein said synthesis filter optimizer comprises a root optimization algorithm, thereby making possible said minimization of said synthesis error, wherein said synthesis filter comprises a predictive coding technique producing said synthesized speech sample from said original speech sample; wherein said predictive coding technique produces first coefficients of a polynomial; wherein said root optimization algorithm is an Iterative algorithm using first roots derived from said first coefficients in a first iteration; and wherein said root optimization algorithm produces second roots using the gradient search algorithm in successive Iterations resulting In a reduction of said synthesis error In said successive iterations.

27. The gradient search algorithm according to Claim 26, wherein the gradient search algorithm further comprises iteratively calculating said gradient vector and recalculating said contribution at each iteration, whereby said decomposition coefficients vary between iterations, and wherein one of said decomposition coefficients corresponds to each of a plurality of said roots.

28. The gradient search algorithm according to Claim 27, wherein said gradient vector and said contribution to said gradient vector are calculated using the formula:

$$\partial\hat{s}(k)/\partial\lambda_r^{(j)} = \sum_{m=0}^{k} u(k-m) \sum_{i=1}^{i=M} b_i K(i,r)(\lambda_i^{(j)})^{(m-1)} + b_r \sum_{m=1}^{k} mu(k-m)(\lambda_r^{(j)})^{(m-1)} \ (k>0).$$

29. A gradient search algorithm for a speech coding system, comprising calculating decomposition coefficients; calculating a first gradient of a polynomial using said decomposition coefficients; estimating roots of said polynomial using said first gradient; recalculating said decomposition coefficients based on said estimating; calculating a second gradient of said polynomial using said recalculated decomposition coefficients; and estimating said roots of said polynomial using said second gradient.

30. The gradient search algorithm according to Claim 29, wherein said gradient and said decomposition coefficients

are calculated using the formulas:

$$\partial\hat{s}(k)/\partial\lambda_r^{(j)} = \sum_{m=0}^{k} u(k-m)\sum_{i=1}^{i\neq M} b_i K(i,r)(\lambda_i^{(j)})^{(m-1)} + b_r\sum_{m=1}^{k} mu(k-m)(\lambda_r^{(j)})^{(m-1)} \quad (k>0)$$

$$b_i = \prod_{j=1,\ j\neq i}^{M} [1/(1-\lambda_j\lambda_i^{-1})].$$

31. The gradient search algorithm according to Claim 29, used in combination with a linear predictive coding speech system.

32. The gradient search algorithm according to Claim 29, used in combination with a method of generating a speech synthesis filter representative of a vocal tract, the method comprising computing a first synthesis error between an original speech and a first synthesized speech sample corresponding to roots estimated with said first gradient; and computing a second synthesis error between said original speech and a second synthesized speech corresponding to roots estimated with said second gradient; wherein said second synthesis error is less than said first synthesis error.

33. The gradient search algorithm according to Claim 32, wherein said gradient and said decomposition coefficients are calculated using the formulas:

$$\partial\hat{s}(k)/\partial\lambda_r^{(j)} = \sum_{m=0}^{k} u(k-m)\sum_{i=1}^{i\neq M} b_i K(i,r)(\lambda_i^{(j)})^{(m-1)} + b_r\sum_{m=1}^{k} mu(k-m)(\lambda_r^{(j)})^{(m-1)} \quad (k>0)$$

$$b_i = \prod_{j=1,\ j\neq i}^{M} [1/(1-\lambda_j\lambda_i^{-1})].$$

34. A gradient search algorithm for a speech coding system, comprising means for calculating decomposition coefficients of a speech synthesis polynomial; means for calculating first roots of said polynomial using said decomposition coefficients; means for recalculating said decomposition coefficients based on said first roots; and means for calculating second roots of said polynomial using said recalculated decomposition coefficients.

FIG.1

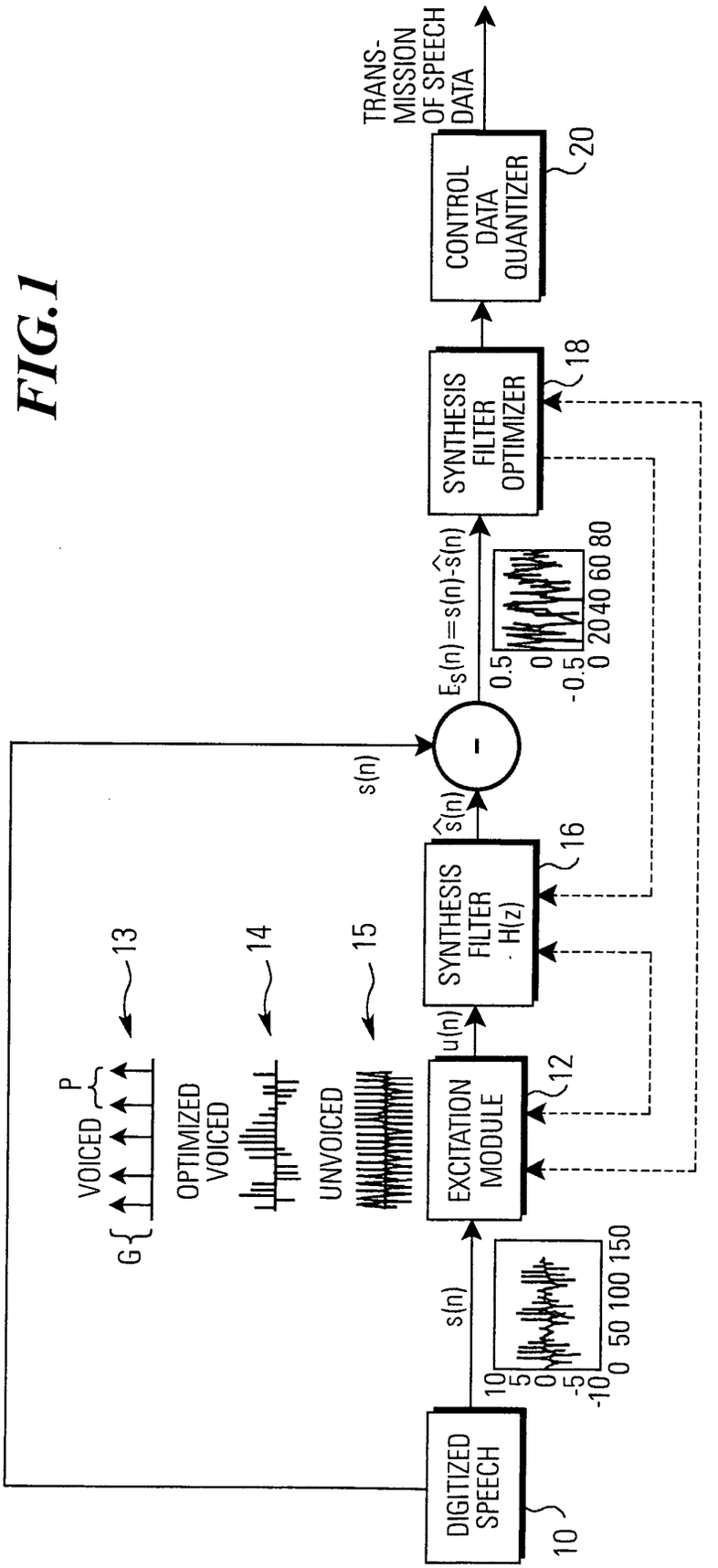

DIGITIZED SPEECH
10
s(n)
EXCITATION MODULE
12
u(n)
SYNTHESIS FILTER · H(z)
16
$\hat{s}(n)$
s(n)
$E_S(n)=s(n)-\hat{s}(n)$
SYNTHESIS FILTER OPTIMIZER
18
CONTROL DATA QUANTIZER
20
TRANS-MISSION OF SPEECH DATA
VOICED
P
G
13
OPTIMIZED VOICED
14
UNVOICED
15

FIG.2A

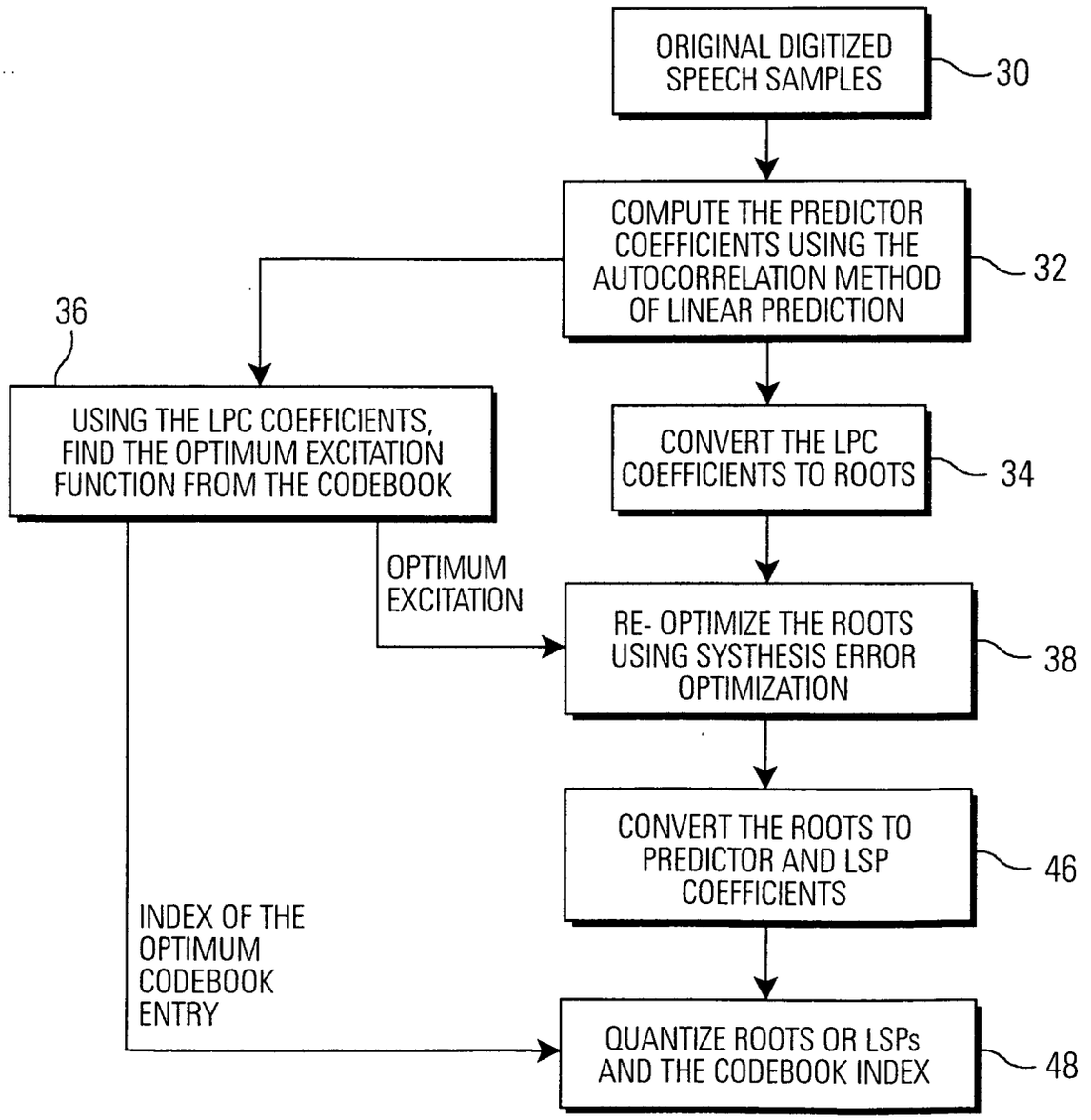

ORIGINAL DIGITIZED SPEECH SAMPLES
30
COMPUTE THE PREDICTOR COEFFICIENTS USING THE AUTOCORRELATION METHOD OF LINEAR PREDICTION
32
36
USING THE LPC COEFFICIENTS, FIND THE OPTIMUM EXCITATION FUNCTION FROM THE CODEBOOK
CONVERT THE LPC COEFFICIENTS TO ROOTS
34
OPTIMUM EXCITATION
RE- OPTIMIZE THE ROOTS USING SYSTHESIS ERROR OPTIMIZATION
38
CONVERT THE ROOTS TO PREDICTOR AND LSP COEFFICIENTS
46
INDEX OF THE OPTIMUM CODEBOOK ENTRY
QUANTIZE ROOTS OR LSPs AND THE CODEBOOK INDEX
48

FIG.2B

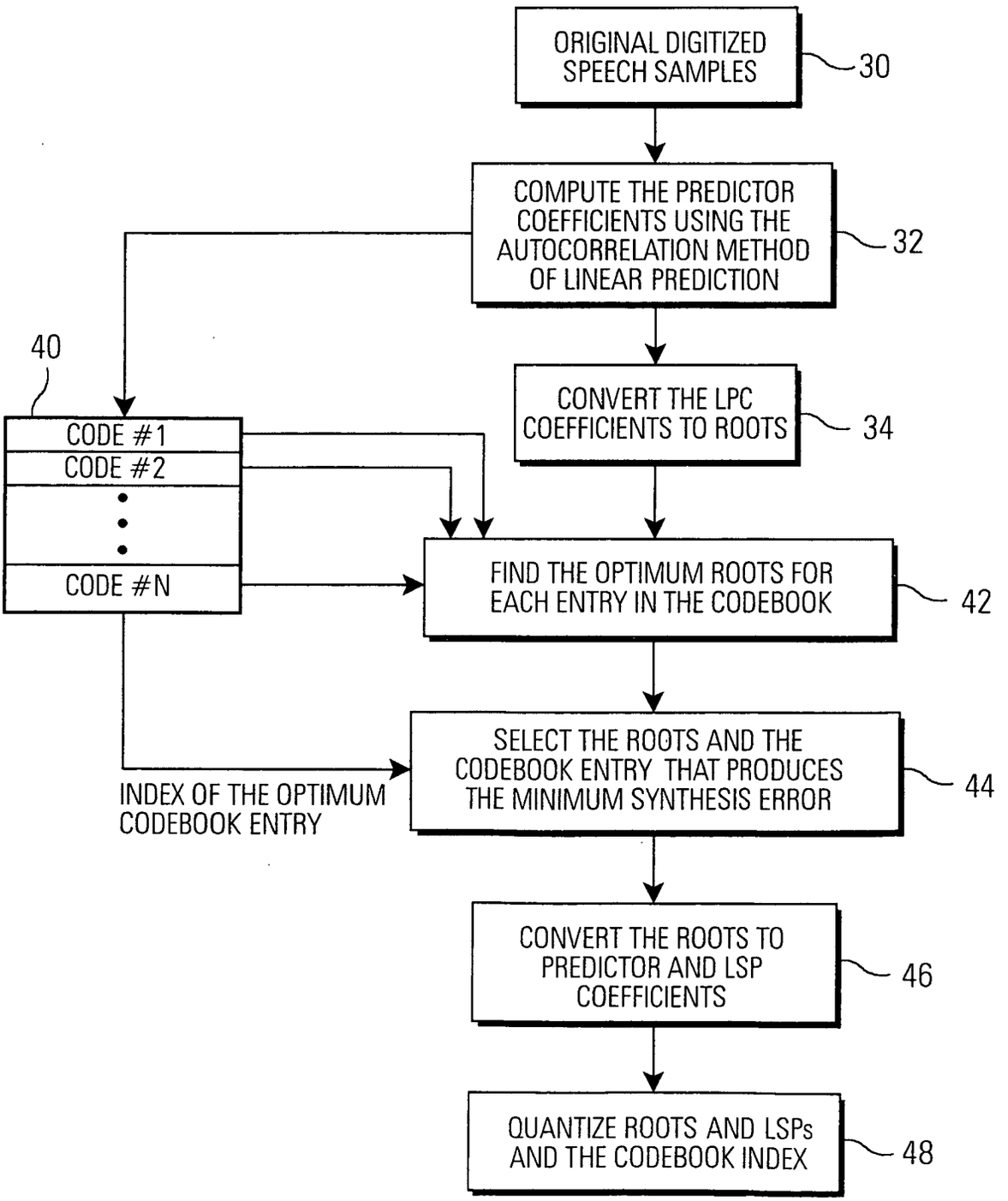

ORIGINAL DIGITIZED SPEECH SAMPLES
30
COMPUTE THE PREDICTOR COEFFICIENTS USING THE AUTOCORRELATION METHOD OF LINEAR PREDICTION
32
40
CONVERT THE LPC COEFFICIENTS TO ROOTS
34
CODE #1
CODE #2
CODE #N
FIND THE OPTIMUM ROOTS FOR EACH ENTRY IN THE CODEBOOK
42
INDEX OF THE OPTIMUM CODEBOOK ENTRY
SELECT THE ROOTS AND THE CODEBOOK ENTRY THAT PRODUCES THE MINIMUM SYNTHESIS ERROR
44
CONVERT THE ROOTS TO PREDICTOR AND LSP COEFFICIENTS
46
QUANTIZE ROOTS AND LSPs AND THE CODEBOOK INDEX
48

FIG.3

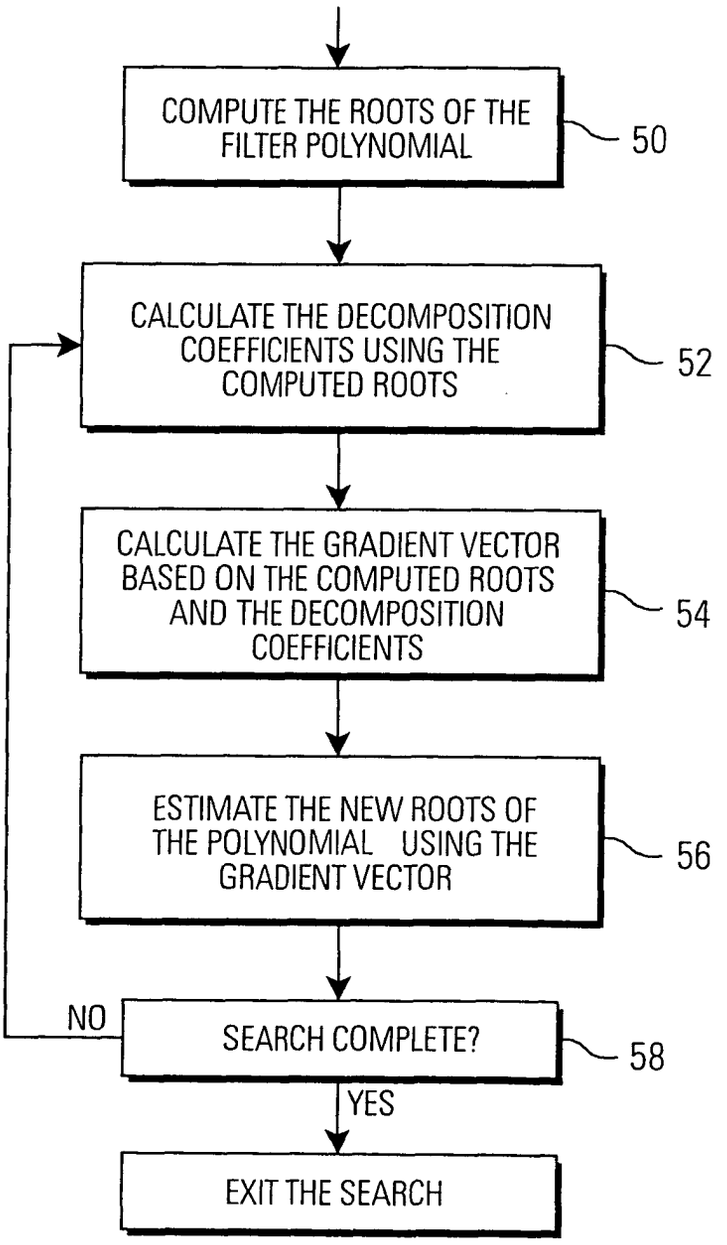

COMPUTE THE ROOTS OF THE FILTER POLYNOMIAL
50
CALCULATE THE DECOMPOSITION COEFFICIENTS USING THE COMPUTED ROOTS
52
CALCULATE THE GRADIENT VECTOR BASED ON THE COMPUTED ROOTS AND THE DECOMPOSITION COEFFICIENTS
54
ESTIMATE THE NEW ROOTS OF THE POLYNOMIAL USING THE GRADIENT VECTOR
56
NO
SEARCH COMPLETE?
58
YES
EXIT THE SEARCH

FIG.4

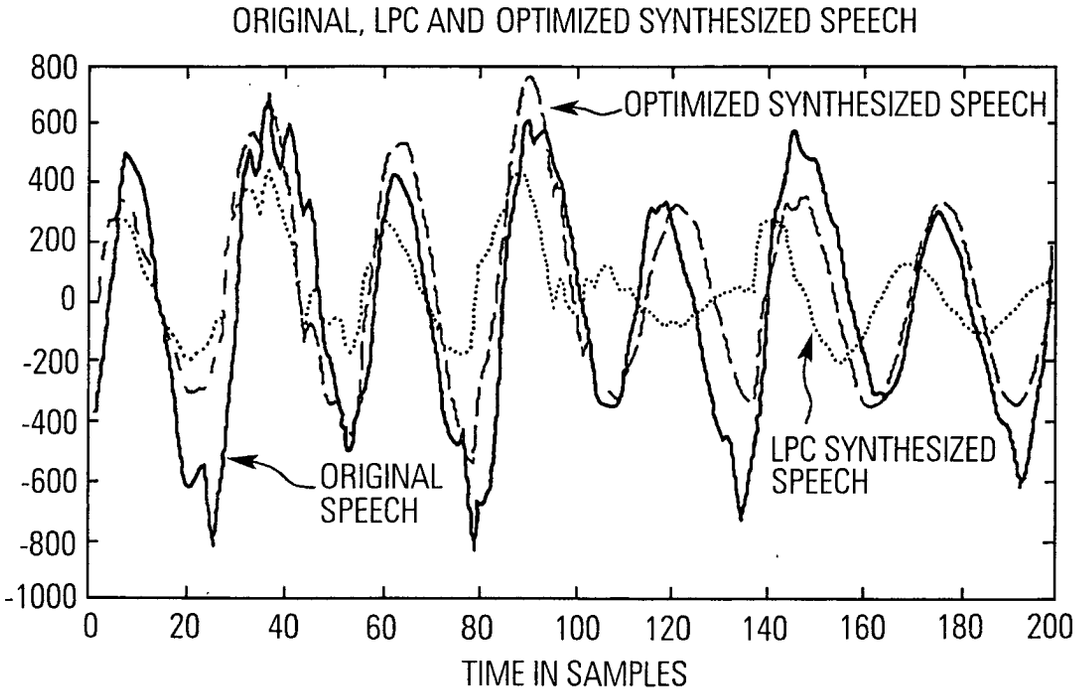

ORIGINAL, LPC AND OPTIMIZED SYNTHESIZED SPEECH
800
600
400
200
0
-200
-400
-600
-800
-1000
0
20
40
60
80
100
120
140
160
180
200
TIME IN SAMPLES
OPTIMIZED SYNTHESIZED SPEECH
ORIGINAL SPEECH
LPC SYNTHESIZED SPEECH

FIG.5

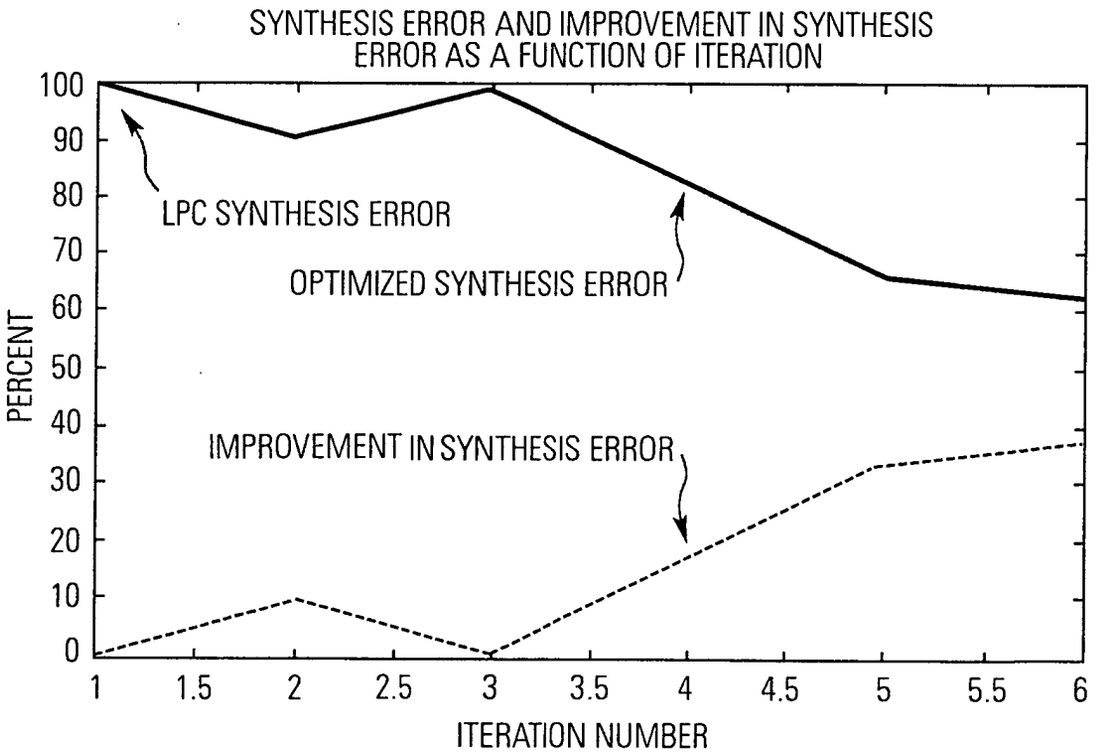

SYNTHESIS ERROR AND IMPROVEMENT IN SYNTHESIS ERROR AS A FUNCTION OF ITERATION
100
90
80
70
60
50
40
30
20
10
0
PERCENT
LPC SYNTHESIS ERROR
OPTIMIZED SYNTHESIS ERROR
IMPROVEMENT IN SYNTHESIS ERROR
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
ITERATION NUMBER

FIG.6

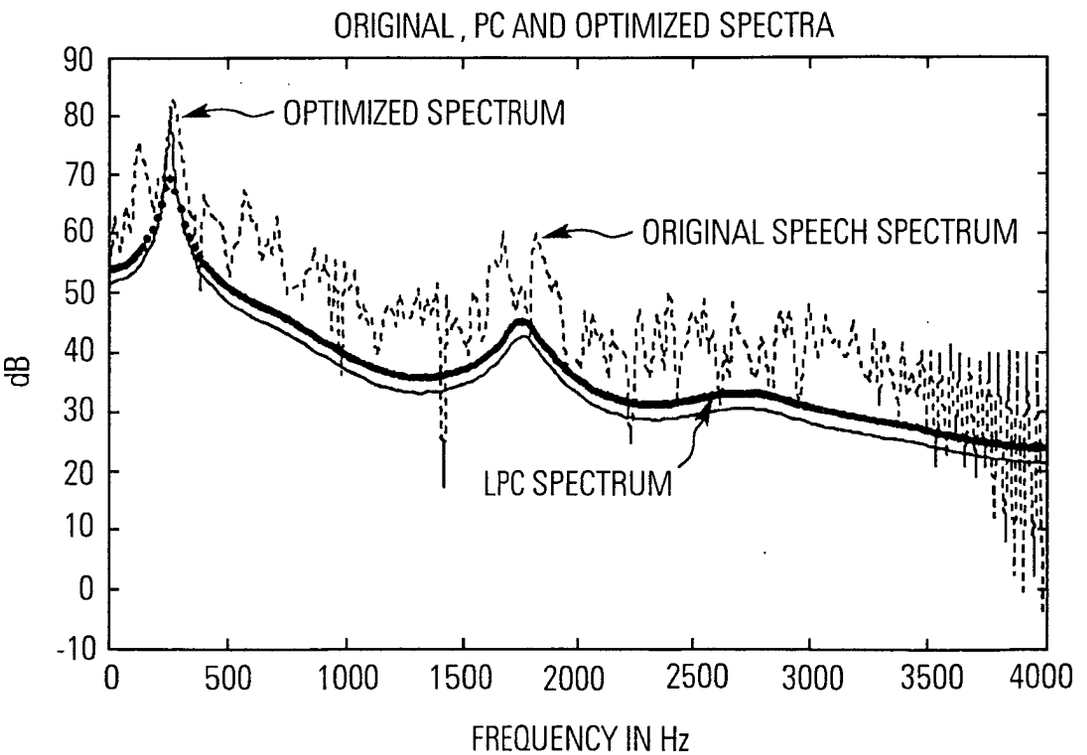

ORIGINAL , PC AND OPTIMIZED SPECTRA
OPTIMIZED SPECTRUM
ORIGINAL SPEECH SPECTRUM
LPC SPECTRUM
dB
90
80
70
60
50
40
30
20
10
0
-10
0
500
1000
1500
2000
2500
3000
3500
4000
FREQUENCY IN Hz

FIG. 7

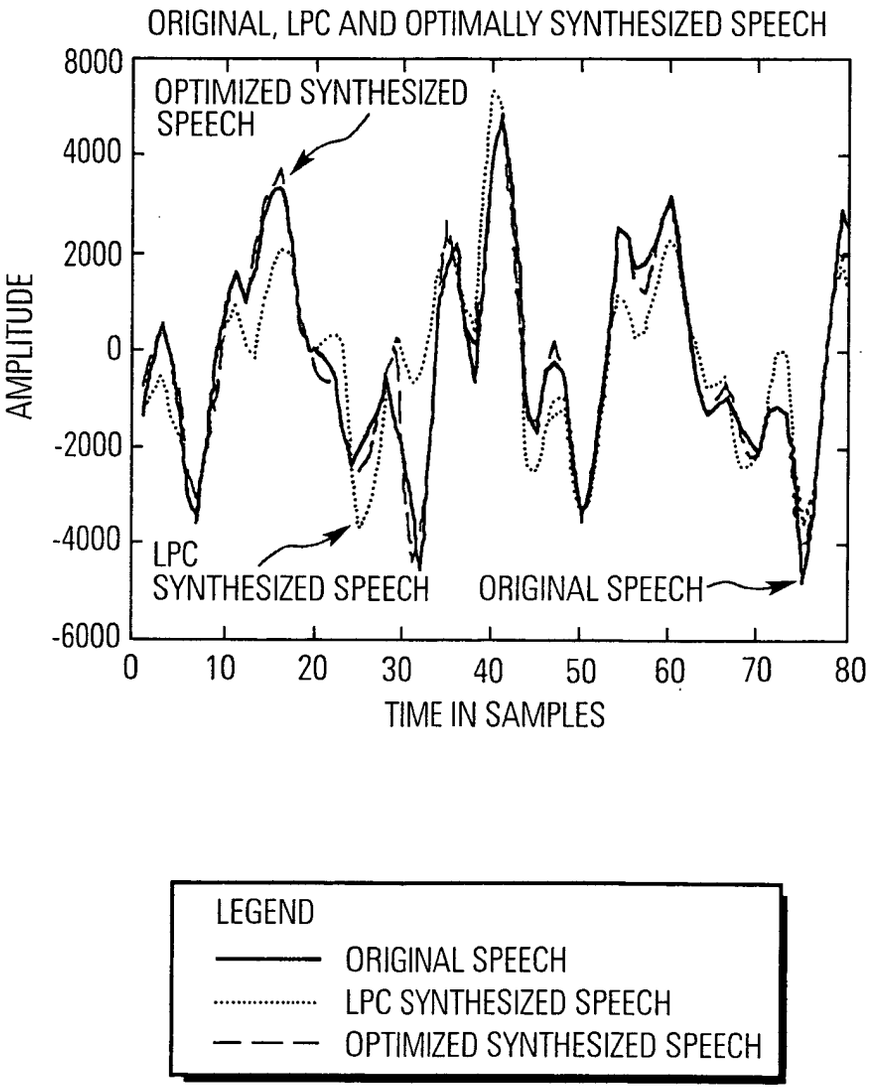

ORIGINAL, LPC AND OPTIMALLY SYNTHESIZED SPEECH
8000
4000
2000
0
-2000
-4000
-6000
AMPLITUDE
OPTIMIZED SYNTHESIZED SPEECH
LPC SYNTHESIZED SPEECH
ORIGINAL SPEECH
0
10
20
30
40
50
60
70
80
TIME IN SAMPLES
LEGEND
ORIGINAL SPEECH
LPC SYNTHESIZED SPEECH
OPTIMIZED SYNTHESIZED SPEECH

FIG.8

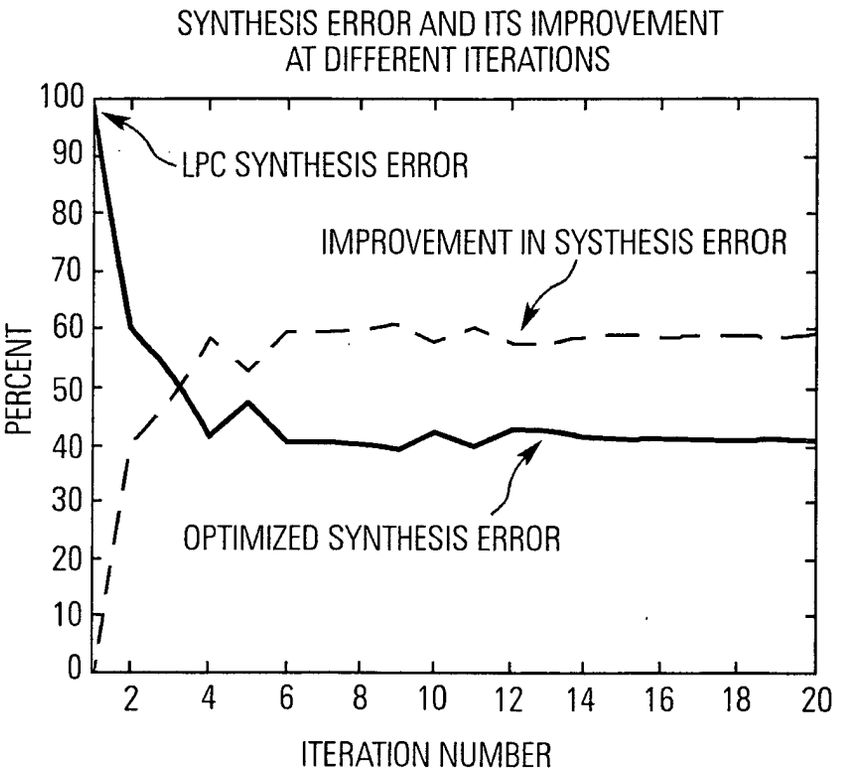
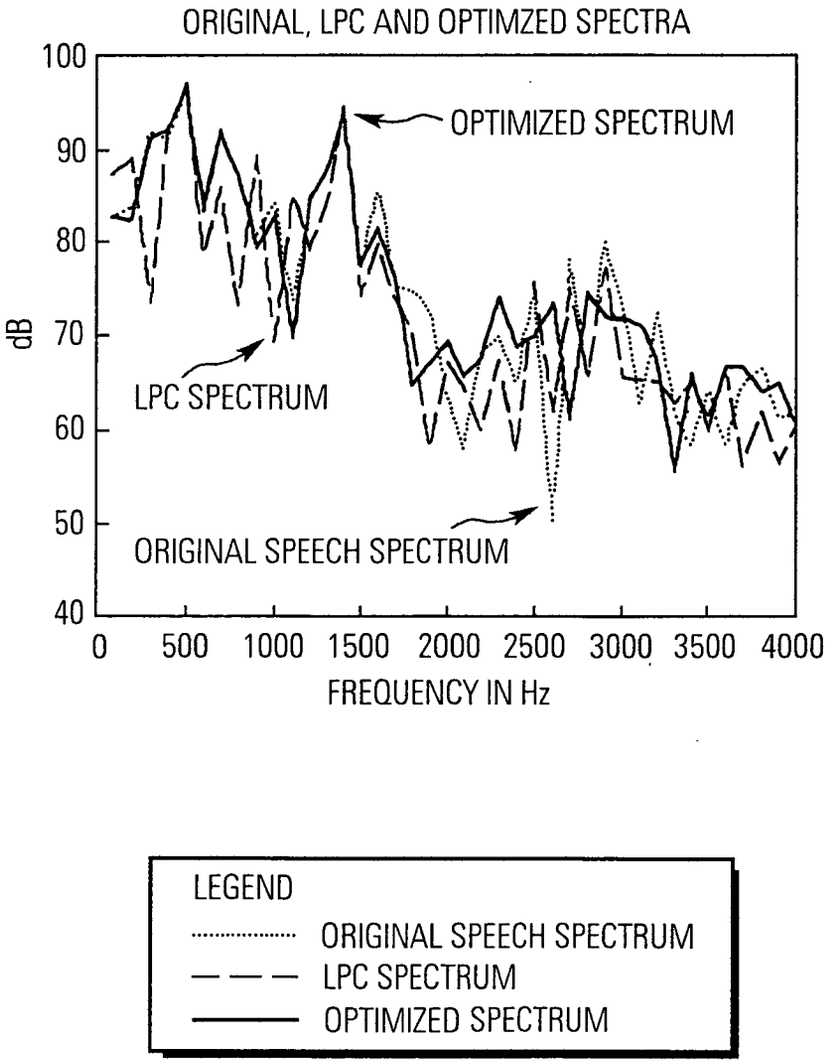

SYNTHESIS ERROR AND ITS IMPROVEMENT
AT DIFFERENT ITERATIONS
LPC SYNTHESIS ERROR
IMPROVEMENT IN SYSTHESIS ERROR
OPTIMIZED SYNTHESIS ERROR
PERCENT
ITERATION NUMBER
100
90
80
70
60
50
40
30
20
10
0
2
4
6
8
10
12
14
16
18
20

FIG.9


ORIGINAL, LPC AND OPTIMZED SPECTRA
100
90
80
70
60
50
40
dB
0
500
1000
1500
2000
2500
3000
3500
4000
FREQUENCY IN Hz
OPTIMIZED SPECTRUM
LPC SPECTRUM
ORIGINAL SPEECH SPECTRUM
LEGEND
ORIGINAL SPEECH SPECTRUM
LPC SPECTRUM
OPTIMIZED SPECTRUM