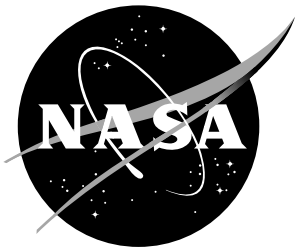
NASA/TM–2000–209845



Improved Speech Coding Based on Open-Loop Parameter Estimation

Jer-Nan Juang
NASA Langley Research Center, Hampton, Virginia

Ya-Chin Chen and Richard W. Longman
Institute for Computer Applications in Science and Engineering (ICASE), Hampton, Virginia

February 2000

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart and peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

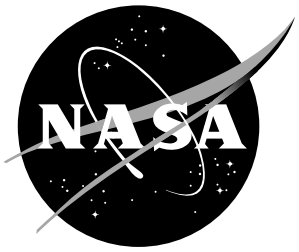
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621–0134

- Phone the NASA STI Help Desk at (301) 621–0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076–1320

NASA/TM–2000–209845



Improved Speech Coding Based on Open-Loop Parameter Estimation

Jer-Nan Juang
NASA Langley Research Center, Hampton, Virginia

Ya-Chin Chen and Richard W. Longman
Institute for Computer Applications in Science and Engineering (ICASE), Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681–2199

February 2000

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076–1320
(301) 621–0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161–2171
(703) 605–6000

Improved Speech Coding Based on Open-Loop Parameter Estimation

Jer-Nan Juang *
NASA Langley Research Center
Hampton, VA 23681

Ya-Chin Chen †and Richard W. Longman‡
ICASE/NASA Langley Research Center
Hampton, VA 23681

Abstract

A nonlinear optimization algorithm for linear predictive speech coding was developed early that not only optimizes the linear model coefficients for the open loop predictor, but does the optimization including the effects of quantization of the transmitted residual. It also simultaneously optimizes the quantization levels used for each speech segment. In this paper, we present an improved method for initialization of this nonlinear algorithm, and demonstrate substantial improvements in performance. In addition, the new procedure produces monotonically improving speech quality with increasing numbers of bits used in the transmitted error residual. Examples of speech encoding and decoding are given for 8 speech segments and signal to noise levels as high as 47 dB are produced. As in typical linear predictive coding, the optimization is done on the open loop speech analysis model. Here we demonstrate that minimizing the error of the closed loop speech reconstruction, instead of the simpler open loop optimization, is likely to produce negligible improvement in speech quality. The examples suggest that the algorithm here is close to giving the best performance obtainable from a linear model, for the chosen order with the chosen number of bits for the codebook.

1 Introduction

Linear prediction speech coding (LPC) techniques were first used for speech analysis and synthesis by Itakura and Saito [1], and Atal and Schroeder[2]. Conventional LPC requires

*Principal Scientist, Structural Dynamics Branch
†Graduate Student Visitor
‡Also Professor of Department of Mechanical Engineering, Columbia University, New York, NY 20017

two computational steps which are coefficient estimation of an all-pole model and quantization of the prediction residual [3,4]. Typically, the model is developed or optimized without regard for the fact that the residual will be quantized before it is transmitted to a receiver for reconstruction, and in addition the quantization is not optimized with respect to each speech segment transmitted.

An algorithm was introduced in [5] which starts from the basic LPC framework, but optimizes the coefficients of the model taking into account the fact that the transmitted error residual is simultaneously quantized into a specified number of levels. In other words the coefficients are optimized with knowledge of precisely what information will be made available for the speech synthesis process. The algorithm simultaneously optimizes the levels chosen for each speech segment rather than using some a priori choice. The fact that this algorithm supplies these two extra aspects to the usual open loop optimization suggests that better performance is achievable by comparison to typical LPC approaches. It is the purpose of this paper to present an improved initialization procedure for the algorithm of [5]. The optimization involved in the algorithm is nonlinear, and hence it can converge to a local minimum, and fail to realize the full potential. Hence, having good initialization for the optimization can substantially improve performance, and this is demonstrated here.

Although the algorithms in [5] and in this paper build on the LPC framework, historically they were developed after observing the attempt to use blind equalization in speech encoding in [6]. Reference [6] uses just two quantization levels for the error residual. In blind equalization of a corrupted binary bit stream, decisions are made each time step about which of the two possible bits was sent. The procedure is "blind" in the sense that it does not know what the input sequence was. If the corruption is not too large the decision process results in making the output equal (or "equalized") to the input bit stream. It is conceivable that when one uses only two quantization levels in the transmitted error residual in speech encoding, a similar binary decision could be made in the speech reconstruction or synthesis step, and this would then avoid the need to transmit the error residual. Numerical experience gave poor results using blind equalization in the closed loop reconstruction necessary for speech encoding, and hence [6] only treats open loop prediction. Here we do not attempt to use blind equalization. We transmit the information necessary for the reconstruction of the residual. The one aspect of the present algorithm in common with [6] and not part of typical LPC, is that the LPC coefficients are optimized with knowledge that the residual is quantized. This time we allow an arbitrary number of quantization levels (among powers of two) rather than just two levels, and furthermore we let the levels be optimized for each

speech segment.

2 Basic Concepts in Linear Predictive Speech Encoding

Here we summarize some basic formulation for LPC as a framework for later discussion [3,4]. Let $x(k)$, $k = 1, 2, \ldots, N$ be the sampled time history of a segment of speech signal (denote the segment by S). Then typical encoding, transmission, and decoding steps are as follows.

2.1 Encoding:

The encoding or speech analysis uses an open loop prediction $\hat{x}_o(k)$ satisfying

$$\hat{x}_o(k) = -\alpha_1 x(k-1) - \alpha_2 x(k-2) - \cdots - \alpha_n x(k-n) \tag{1}$$

where the coefficients α_i are chosen to make the open loop prediction error $\epsilon_o(k)$ minimize the optimization criterion

$$J_o = \sum_S \epsilon_o^2(k) \tag{2}$$

where

$$\epsilon_o(k) = x(k) - \hat{x}_o(k) \tag{3}$$

Note that by substituting Eq. (1) into Eq. (3), the speech sequence $x(k)$ exactly satisfies the finite-difference model

$$x(k) + \alpha_1 x(k-1) + \alpha_2 x(k-2) + \cdots + \alpha_n x(k-n) = \epsilon_o(k) \tag{4}$$

By choosing the α_i to minimize the equation error in Eq. (4), one minimizes the one step ahead prediction error, i.e. the open loop prediction error. The sequence of values of the input $\epsilon_o(k)$ are now quantized in some way to represent $\epsilon_o(k)$ by an approximate signal $\hat{\epsilon}_o(k)$, requiring fewer number of bits to transmit than the full number in $x(k)$. This accomplishes compression of the signal.

2.2 Transmission:

The values of the α_i and initial conditions of $x(k)$ for n time steps are transmitted, and the sequence of $\hat{\epsilon}_o(k)$ for all time steps are transmitted in some form. For an appropriately chosen order n, the left hand side of Eq. (4) captures the majority of the signal, so the error in the finite-difference representation, $\epsilon_o(k)$, should be substantially smaller than the signal $x(k)$ itself. This indicates that using fewer bits to form $\hat{\epsilon}_o(k)$ need not result in degraded quality in the reconstructed signal.

2.3 Decoding:

In the speech synthesis step, the signal is reconstructed by the receiver, using the closed loop formula

$$\hat{x}_c(k) = -\alpha_1 \hat{x}_c(k-1) - \alpha_2 \hat{x}_c(k-2) - \cdots - \alpha_n \hat{x}_c(k-n) + \hat{\epsilon}_o(k) \tag{5}$$

starting with the transmitted initial conditions $\hat{x}_c(k) = x(k)$. Comparing to equation (4), the only error in this reconstruction is the quantization used in the transmitted values of $\hat{\epsilon}_o(k)$.

By using the open loop equation for encoding one obtains a relatively simple linear problem to find the coefficients α_i. Since the reconstruction is necessarily closed loop because the receiver does not know the previous n values of $x(k)$, it would yield better reconstructed values if the encoding optimization was done for the closed loop prediction equation, but this is a nonlinear optimization problem which is substantially more difficult to solve.

3 Encoding Scheme

In [5], an encoding scheme is introduced which makes the choice of the quantization levels for $\hat{\epsilon}_o(k)$ part of the optimization. The coefficients α_i are optimized simultaneously with the choice of these levels.

3.1 Codebook:

The input $\epsilon_o(k)$ is constrained to be a linear combination of the entries in the vectors of a binary codebook. To form a codebook, first pick the number of bits r to be used. Then form the column vectors of the codebook as all possible vectors of length r with each entry either $+1$ or -1. For example, for $r = 4$ there are 16 vectors in the codebook. Denote the ith entry of the jth vector in the codebook as δ_{ji}.

3.2 Encoding:

The encoding in Eq. (1) is modified as follows for the jth codebook entry

$$\hat{x}_o(k, j(k)) = -\alpha_1 x(k-1) - \alpha_2 x(k-2) - \cdots - \alpha_n x(k-n) + u(j(k)) \tag{6}$$

where the forcing function is taken as a linear combination

$$u(j(k)) = \beta_1 \delta_{j1} + \beta_2 \delta_{j2} + \cdots + \beta_r \delta_{jr} \tag{7}$$

of the jth codebook vector entries. The objective is then to determine constant values for α_i and β_i for all time steps of the speech segment, and determine codebook entries $j(k)$ for every time step k, in order to achieve the following minimization

$$J_o = \min_{\alpha_i,\beta_i,j(k)} \sum_{k=1}^{N} \lambda^{N-k} \epsilon_o^2(k, j(k)) \tag{8}$$

$$\epsilon_o(k, j(k)) = x(k) - \hat{x}_o(k, j(k)) \tag{9}$$

The λ is a positive number less than or equal to one, representing a forgetting factor.

To accomplish this minimization, Ref. [5] formulates the recursive least squares equations for finding the values of the coefficients α_i and β_i that minimize the weighted (by λ) Euclidean norm of the equation errors for all k and any choice of j for the equation

$$x(k) + \alpha_1 x(k-1) + \alpha_2 x(k-2) + \cdots + \alpha_n x(k-n) = \beta_1 \delta_{j1} + \beta_2 \delta_{j2} + \cdots + \beta_r \delta_{jr} \tag{10}$$

As noted earlier for LPC, this process minimizes the (weighted) open loop prediction error of Eq. (6). Such a recursive computation produces running estimates $\hat{\alpha}_i(k)$, $\hat{\beta}_i(k)$. The desired solutions for these coefficients minimizing the least squares error are obtained when k reaches N. However, Ref. [5] also incorporates the choice of j in this running estimation, picking its value each time step to minimize the current estimation error before progressing to the next step. The result is that for sufficiently long data sets, the recursively updated values of $\hat{\alpha}_i(k)$, $\hat{\beta}_i(k)$ converge to constant values along with a computed set of $j(k)$ for the speech block. The value of λ can be adjusted to influence the number of data points needed to reach constant values.

3.3 Transmission:

The transmission of the coded signal can be done by sending the final minimizing values for α_i and β_i, the initial conditions, and the code vector entry number $j(k)$ identifying the minimizing code vector for each time step. Since the choice of code vector typically will not change every time step, one can compress the amount of data further by simply transmitting changes in the code vector when they occur.

3.4 Reconstruction:

The speech synthesis uses the transmitted information to determine $u(j(k))$ according to Eq. (7), and recursively computes

$$\hat{x}_c(k) = -\alpha_1 \hat{x}_c(k-1) - \alpha_2 \hat{x}_c(k-2) - \cdots - \alpha_n \hat{x}_c(k-n) + u(j(k)) \tag{11}$$

starting by using the transmitted initial values of $x(k)$ for the initial conditions on $\hat{x}_c(k)$.

3.5 Initialization:

The initialization for the minimization process starts with the choice of the number of codebook entries, i.e. the number of bits r, and then needs initial guesses for the coefficients $\hat{\alpha}_i(0)$, $\hat{\beta}_i(0)$ and an initial value for the covariance function $P(0)$ in the least squares update formula. As is typically done in recursive least squares, Ref. [5] sets the $\hat{\alpha}_i(0)$, $\hat{\beta}_i(0)$ to zero, and $P(0)$ to be a large number ([5] uses 100,000 in its examples) times the identity matrix of appropriate dimension.

The set of possible values of $u(j(k))$ achievable are given by picking all possible signs in $u(j(k)) = (\pm)_1\beta_1 + (\pm)_2\beta_2 + \cdots + (\pm)_r\beta_r$, producing 2^r levels. The optimization achieved here differs from that in LPC because the discretization levels are now optimized for each speech block, and in addition, the coefficients α_i are optimized with knowledge of these levels. Hence, for a given number of quantization levels, if a global minimum is achieved in Eq. (8), then the method of Ref. [5] would necessarily out perform typical LPC with the same number of levels. The problem addressed here is a nonlinear problem, and hence it is possible to converge to a local minimum. Whether or not one reaches a good minimum can depend on the starting conditions in the minimization process, i.e. the initialization. The objective of this paper is to present improved starting conditions for the algorithm, and to demonstrate the resulting improved error levels upon convergence.

4 Improved Starting Conditions

Instead of starting with the desired bit number and performing the optimization, we first optimize for bit number $r = 1$, and use the results to optimize bit number $r = 2$, continuing until the desired bit number (or speech quality) is reached.

For bit number $r = 1$, we need initial values for α_i and β_1, as well as the initial value for the $(n+1) \times (n+1)$ dimensional covariance matrix P. The quantity $\hat{\beta}_1(0) = 0$ is used, but $\hat{\alpha}_i(0)$ are estimated by minimizing the sum of the squares of the $\epsilon_o(k)$ in Eq. (4) over the speech segment. Just as in LPC, it is desirable to have the left hand side capture as much of the behavior of the signal as possible, leaving as little as possible for the $u(j(k))$ to capture its resulting residual. Write Eq. (4) in matrix form including each time step of the N length speech segment, and using the first n points as the initial conditions

$$\underline{x} = -A\underline{\alpha} + \underline{\epsilon} \tag{12}$$

where

$$\begin{aligned}
\underline{x} &= \left[\begin{array}{cccc} x(n+1) & x(n+2) & \cdots & x(N) \end{array}\right]^T \\
\underline{\epsilon} &= \left[\begin{array}{cccc} \epsilon_o(n+1) & \epsilon_o(n+2) & \cdots & \epsilon_o(N) \end{array}\right]^T \\
\underline{\alpha} &= \left[\begin{array}{cccc} \alpha_1 & \alpha_2 & \cdots & \alpha_n \end{array}\right]^T \\
A &= \left[\begin{array}{cccc} x(n) & x(n-1) & \cdots & x(1) \\ x(n+1) & x(n) & \cdots & x(2) \\ \vdots & \vdots & \ddots & \vdots \\ x(N-1) & x(N-2) & \cdots & x(N-n) \end{array}\right]
\end{aligned} \tag{13}$$

Then the value of $\underline{\alpha}$ that minimizes $\underline{\epsilon}^T\underline{\epsilon}$, i.e. the desired starting values $\underline{\hat{\alpha}}(0)$, satisfies $A^T\underline{x} = (A^TA)\underline{\hat{\alpha}}(0)$ which can be rewritten as

$$\underline{\hat{\alpha}}(0) = -P_0X \tag{14}$$

where

$$\begin{aligned}
X = A^Tx &= \left[\begin{array}{cccc} C_{n,n+1} & C_{n-1,n+1} & \cdots & C_{1,n+1} \end{array}\right]^T \\
P_0 = \left[A^TA\right]^\dagger &= \left[\begin{array}{cccc} C_{n,n} & C_{n,n-1} & \cdots & C_{n,1} \\ C_{n-1,n} & C_{n-1,n-2} & \cdots & C_{n-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1,n} & C_{1,n-1} & \cdots & C_{1,1} \end{array}\right]^\dagger
\end{aligned} \tag{15}$$

and superscript † indicates the inverse, or Moore-Penrose pseudo inverse if appropriate. The $C_{i,j}$ represents the correlation between the values of the data sequence $x(k)$ and the sequence shifted by $i-j$ time steps. Thus, P_0 is the inverse (or pseudo inverse) of the data correlation matrix.

The weighted recursive least squares algorithm is a recursive version of a least squares equation like Eq. (14) but including the βi and a forgetting factor. It computes the change needed in the coefficient estimates each time a new data point is added to the data set. Part of the recursive formula is a recursive version of the matrix $P_0 = (A^TA)^\dagger$ above, generalized to include the β_i terms and denoted by P. For bit number $r = 1$, the $P(0)$ of the recursive formula is the inverse of the correlation matrix for $\left[\begin{array}{cc} \underline{\alpha}^T & \beta_1 \end{array}\right]^T$, and hence we use P_0 from Eq. (15) for the upper left $n \times n$ partition, and need to assign values for one more row and column. All these new elements are set to zero, except for the final diagonal element associated with knowledge of β_1 which is chosen as 10^6. Such a large number represents essentially no a priori knowledge about this coefficient.

Once the solution for bit number $r = 1$ is obtained, then we progress to bit number $r = 2$, etc. In general, when going from r to $r+1$ for any r, the inital values are set as follows:

1. The final values with bit number r, obtained for the coefficients α_i after running the recursive least squares until stabilized values are obtained, are used as starting values $\underline{\hat{\alpha}}(0)$ for the new problem using $r+1$ bits.

2. The corresponding procedure is also used for the r initial values for β_1, $\beta_2, \ldots, \beta_r$. The initial value for β_{r+1} is set to zero.

3. The $(n+r+1) \times (n+r+1)$ dimensional $P(0)$ for the problem using $r+1$ bits takes the form of a block diagonal matrix

$$P(0) = \text{diag}(P_{11}(0), P_{22}(0), P_{33}(0)) \tag{16}$$

4. The $P_{11}(0)$ for the problem with bit number $r+1$ is of dimension $n \times n$, and is taken as the final value of the upper left $n \times n$ partition of the $(n+r) \times (n+r)$ dimensional matrix P for bit number r. after finishing the recursive computation.

5. The $P_{22}(0)$ for bit number $r+1$ is of dimension $r \times r$, and is the product of $r \times r$ identity matrix and the norm, or maximum singular value, of $P_{11}(0)$.

6. The $P_{33}(0)$ for bit number $r+1$ is a scalar set to 10^6 .

This procedure for initializing makes full use of available information for the α_i . The initialization for the β_i is somewhat ad hoc, and is made with the following considerations in mind. Numerical experiments showed that using the full $(n+r) \times (n+r)$ final matrix P for bit level r, in place of the first two partiitions of the block diagonal $P(0)$ for the next bit number, results in rather small adjustments of the model coefficients in the next level, and in corresponding small improvements in speech quality with each bit number. On the other hand, replacing the $P_{22}(0)$ of item 5 by 10^6 times the $r \times r$ identity matrix, i.e. using essentially no a priori information about the first r coefficients among the β_i, did not achieve good results either. It appears to converge to a local minimum solution with poor speech quality. The choice described above allows these r coefficients β_i to be adjusted about as much as the α_i's, and this appears to be a good compromise. There is no a priori information on the remaining coefficient, β_{r+1}, and using 10^6 leaves it totally free to be adjusted.

5 Performance of the Modified Algorithm

Eight speech segments from two speakers are used to demonstrate the performance of the modified algorithm. The first four are from a female speaker, and correspond to the words:

The pipe / be-gan / to rust / while new. The remaining four are from a male speaker saying: Oak is / strong / and also / gives shade. The lengths of these eight segments are $3100, 3550, 4720, 6650, 4300, 3700, 4500, 5450$ data points, respectively. The length of the filter is chosen to be $n = 10$ which is a commonly used order for LPC speech modeling. The forgetting factor is set to $\lambda = 0.999$.

Two measures of the speech quality of the reconstructed signal are considered, the Euclidean norm of the error, err, and the signal to noise ratio, SNR, i.e. the norm of the signal divided by the norm of the error, in dB

$$\|err\| = \left[\sum\nolimits_k (x(k) - \hat{x}_c(k))^2\right]^{1/2}$$

$$SNR = 20\log(\|x\| \,/\, \|err\|); \|x\| = \left[\sum\nolimits_k x^2(k)\right]^{1/2} \tag{17}$$

Tables 1 and 2 give these measures for the algorithm of Ref. [5] used on the eight speech segments, for bit numbers ranging from $r = 1$ to 10. To evaluate the amount of compression obtained at each bit level, we comment that the unencoded signal uses 16 bits. The SNR's for 10 bits tend to be in the range from 8 to 11 dB. The SNR tends to saturate as the bit number increases, with only small improvements obtained with increasing the bit number beyond 4 or 5. However, an important property is that the speech quality does not necessarily improve each time the number of bits is increased. This property would not occur if we were able to obtain a global minimum each time.

Tables 3 and 4 give the corresponding results using the modified algorithm with the improved starting conditions. The average of the SNR's with bit number $r = 10$ for the female speaker is 35 dB, and for the male speaker is 28 dB, which represents a very substantial improvement. By making use of the results for bit number r to start the algorithm for bit number $r + 1$, the resulting SNRs now exhibit monotonic improvement with increasing bit number. There appears to be a relationship between how good the bit number 1 result is, and how good higher bit numbers are. For example, segment number 3 starts with the highest SNR at bit number one, and for bit number 10 it is still the highest with an impressive SNR of 47 dB. Similarly, segment number 8 starts with the lowest SNR and ends with the lowest for bit number 10.

The use of the result from the previous bit number makes the computation for the next bit number take less time than starting from the initialization for that bit number used in Ref. [5]. In the case of speech segment 4 which is the longest segment, the solutions for bit levels $r = 4$ through 7 took about 48% less time than using [5], and for bit levels $r = 8, 9$, and 10 it took $43\%, 34\%$, and 27% less time, respectively. However, to get the initialization

for a given bit number we need to run all lower bit numbers first, and this means that using the new initialization take somewhat longer. For segment 4, the total computation time for bit level 4 takes approximately twice as long as in Ref. [5], and for bit level 7 somewhat less than three times as long. For this extra computation time the signal to noise ratios improve from 8.9 and 9.5 to 16 and 27 dB respectively. These computation times using code written in Matlab and run on a work station are near real time.

The mean opinion score (MOS) is the most commonly used measure for the subjective quality of coded speech. It is extracted from the results of a category-rated test performed by 20 to 60 untrained listeners. Reference [3] describes a curve fitting procedure used to convert MOS to equivalent Q values (EQ), or dB levels which we can compare to our SNRs. The dB values are categorized in increments of 5 dB starting from 5 dB (bad) to 35 dB (good). Table 5 reproduced from [3] gives such evaluations for some existing coders. The flat condition in the table refers to unfiltered speech recorded with a high quality microphone, and the IRS condition refers to speech filtered through an IRS transmitting filter, such as speech that would be recorded from a typical telephone handset. The line labeled "source" represents the error between the original signal and the signal using 16 bits which is then used for the encoding. Among the coding methods listed, the conventional LDCELP employs a 10-bit codebook with a 50th order LPC predictor and a 10th order adaptive linear predictor. VSELP uses two 7 bit codebooks and a long term filter state, which is also a 7 bit codebook (together requiring 14 bits for index delivery), with a 10th order LPC predictor to carry out speech coding. Together this requires 14 bits for index delivery, so that for comparison purposes one must compare to the performance using a 14 bit code book in the method presented here (beyond the last entry for 10 bits in our table). Table 5 gives a rough understanding of what we might expect if MOS tests were run on the current method, and it is clear that the present method is competitive. However, true MOS tests under uniform testing conditions for each vocoder (voice encoder) are needed to actually determine the potential performance advantages of the new method.

As in LPC, the information transmitted in the vocoder proposed here is optimized for reconstruction using a open loop predictor, but the receiver necessarily reconstructs with a closed loop predictor. It is of interest to see how much signal is lost in the open loop encoding and how much is lost in the closed loop reconstruction. This information is given in Tables 6 through 9. The column labeled SNRc is the signal to noise ratio given previously for the reconstructed signal using the closed loop formula (11), and SNRo is the signal to noise ratio of the open loop prediction of equation (6). The third column gives the percent

of signal to noise ratio of SNRc compared to SNRo. The best that the reconstruction could possibly do is to reproduce the open loop encoding, which corresponds to 100%. A smaller percentage indicates the amount of SNR lost by going from open to closed loop for performing the speech reconstruction. By bit number 10 the amount of SNR lost is about one fourth, with the percentages ranging from 68.9% to 78.2% for the 8 speech segments. Again, the speech segment with the best percentage for bit 1, has the best percentage for bit 10.

6 Potential Improvement with Closed Loop Optimization

What matters in any vocoder is the quality of the reconstructed speech. LPC optimizes the quality of the speech encoded with the open loop equation (6) because this optimization is relatively simple, and the same is done here. Presumably, improved open loop encoding is reflected in improved closed loop reconstruction. In this section we address the question of how much improvement might be obtainable if we optimized the error in the reconstruction. This means that we replace Eqs. (2) and (3) by

$$J_c = \sum_S \epsilon_c^2(k) \tag{18}$$

$$\epsilon_c(k) = x(k) - \hat{x}_c(k, j(k)) \tag{19}$$

$$\hat{x}_c(k, j(k)) = -\alpha_1 \hat{x}_c(k-1, j(k-1)) - \alpha_2 \hat{x}_c(k-2, j(k-2)) - \cdots - \alpha_n \hat{x}_c(k-n, j(k-n)) + u(j(k)) \tag{20}$$

with the closed loop output $\hat{x}_c(k)$ of Eq. (11) substituted, and then develop an algorithm to minimize Eq. (18) over the α_i, β_i, and $j(k)$. In order to minimize J_c, we develop a nonlinear least squares algorithm using analytical gradient and Hessian information, and setting any negative eigenvalues to zero for that portion of the Hessian that comes from the second derivative terms [7]. These iterations are started for each bit level using the results of the vocoder developed here. Thus, the nonlinear least squares algorithm of this section could be made the second part of the total speech algorithm, aiming to reach speech encoding whose reconstruction is the best possible for the chosen model order and bit number.

Table 10 gives the results of this optimization. For bit number 10 the amount of improvement over Tables 3 and 4 is always less than 1 dB, and often substantially less. Thus, we conclude that the extra complexity in optimizing the reconstructed speech signal error as an extra step after optimizing the open loop encoding, is not justified. Of course optimizing the reconstructed speech signal is a nonlinear optimization. There is no way to know

whether we have found the global minimum by use of the nonlinear least squares algorithm here, initialized from the open loop optimization results. Nevertheless, the consistency of all of these results for the 8 speech segments suggests that there is only a very small amount of improvement available by doing the closed loop optimization in place of the open loop. This suggests that the vocoder developed here easily captures essentially all of the potential speech quality available by the chosen filter order and bit number (or codebook vectors).

7 Conclusions

Here we have developed an initialization process for the vocoder developed earlier that very substantially improves its performance. It also consistently gives improved performance when the number of bits used is increased. Although we optimize the open loop predictor as does LPC, the amount of improvement is quite small that could be obtained by actually directly optimizing the closed loop reconstructed speech signal quality. It is sufficiently small that any significant extra computational effort would not be justified. Rough comparisons indicate that the proposed vocoder performance could be competitive. The next step is to actually evaluate the potential performance advantages using MOS tests comparing to existing methods.

References

[1] F. Itakura and S. Saito, "Analysis Synthesis Telephony Based on the Maximum Likelihood Method," *Proceedings of the 6th International Congress on Acoustics*, C-5-5, 1968.

[2] B. S. Atal and M. R. Schroeder, "Predictive Coding of Speech Signals," *Proceedings of the 6th International Congress on Acoustics*, C-5-4, 1968.

[3] W. B. Kleijn and K. K. Paliwal, *Speech Coding and Synthesis*, Elsevier, Amsterdam, 1995.

[4] J. R. Deller, Jr., J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*, Prentice-Hall, Inc., NJ., 1987.

[5] Jer-Nan Juang and Ya-Chin Chen, *Signal Prediction with Input Identification*, NASA/TM-1999-209705, October 1999.

[6] Y. C. Chen, T. Stathaki and A. G. Constantinides, "Adaptive Prediction Based on Blind Equalisation Principles," *ISMIP-96*, Taiwan, pp. 301–307, December 1996.

[7] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, John Academic Press, London, 1981.

	seg#1		*seg*#2		*seg*#3		*seg*#4	
bit#	$\|err\|$	*SNR*	$\|err\|$	*SNR*	$\|err\|$	*SNR*	$\|err\|$	*SNR*
1	10.6900	2.1020	10.2170	3.1198	10.2260	3.0743	10.0250	2.5206
2	7.5088	5.1704	7.2539	6.0948	6.9625	6.4135	6.0034	6.9742
3	6.1292	6.9338	6.2971	7.3234	5.5346	8.4070	5.0588	8.4613
4	5.7824	7.4397	6.4906	7.0606	4.9370	9.3995	4.8093	8.9005
5	5.6382	7.6591	6.1148	7.5786	5.2342	8.8918	4.5001	9.4777
6	5.5952	7.7256	7.8297	5.4313	4.8731	9.5128	4.4749	9.5266
7	5.6201	7.6870	6.5351	7.0012	4.6804	9.8631	4.4726	9.5310
8	6.1560	6.8958	8.6430	4.5730	4.9920	9.3034	4.3559	9.7606
9	6.1446	6.9119	7.1184	6.2586	4.7546	9.7266	4.4212	9.6313
10	5.4031	8.0289	5.3697	8.7072	4.6582	9.9045	4.3060	9.8608

Table 1: The Euclidean norm and the signal to noise ration for segments #1, #2, #3, and #4 using the original initialization in [5].

	$seg\#5$		$seg\#6$		$seg\#7$		$seg\#8$	
$bit\#$	$\|err\|$	SNR	$\|err\|$	SNR	$\|err\|$	SNR	$\|err\|$	SNR
1	9.2846	2.2080	10.5040	2.5518	8.8284	2.5670	8.9596	1.0189
2	5.7972	6.2990	7.6989	5.2505	4.9202	7.6451	6.8906	3.2996
3	4.5242	8.4525	6.8499	6.2653	3.5340	10.5190	6.4524	3.8702
4	4.0774	9.3557	6.5148	6.7010	3.3269	11.0439	6.3875	3.9580
5	4.0296	9.4580	6.0687	7.3171	3.2257	11.3123	9.8403	0.2045
6	3.9711	9.5852	6.3260	6.9565	2.8116	12.5057	9.7462	0.2879
7	3.8738	9.8006	5.9694	7.4604	2.7830	12.5944	7.5068	2.5556
8	3.9596	9.6104	6.2548	7.0548	2.7642	12.6532	8.1778	1.8120
9	3.8545	9.8440	5.8460	7.6419	2.7472	12.7067	7.2672	2.8373
10	3.5071	10.6643	4.6599	9.6115	2.7996	12.5428	10.1554	-0.0693

Table 2: The Euclidean norm and the signal to noise ratio for segments #5, #6, #7, and #8 using the original initialization in [5].

	$seg\#1$		$seg\#2$		$seg\#3$		$seg\#4$	
$bit\#$	$\|err\|$	SNR	$\|err\|$	SNR	$\|err\|$	SNR	$\|err\|$	SNR
1	10.5016	2.2567	12.6157	2.9729	10.7945	3.1509	13.2577	2.6584
2	6.0716	7.0158	7.5711	7.4079	7.8669	5.8988	7.8404	7.2209
3	3.5933	11.5721	4.4410	12.0415	4.3603	11.0247	4.7244	11.6208
4	2.3918	15.1072	2.9581	15.5708	2.2975	16.5897	2.8488	16.0145
5	1.6356	18.4081	1.9893	19.0172	1.3518	21.1970	1.8872	19.5915
6	1.2033	21.0746	1.3757	22.2205	0.7285	26.5661	1.1658	23.7752
7	0.9633	23.0066	0.9821	25.1480	0.3593	32.7065	0.8148	26.8870
8	0.8184	24.4221	0.7491	27.5006	0.2071	37.4924	0.5637	30.0868
9	0.6744	26.1030	0.6171	29.1840	0.1265	41.7704	0.4312	32.4149
10	0.5545	27.8036	0.5056	30.9153	0.0717	46.7031	0.3596	33.9909

Table 3: The Euclidean norm of the error and the signal to noise ratio for segments #1, #2, #3, and #4 using the new initialization procedure.

	seg#5		*seg*#6		*seg*#7		*seg*#8	
bit#	$\|err\|$	*SNR*	$\|err\|$	*SNR*	$\|err\|$	*SNR*	$\|err\|$	*SNR*
1	10.0058	2.5381	10.4803	2.5715	9.8475	2.6128	11.7792	1.4402
2	5.2222	8.1861	6.2075	7.1206	5.6635	7.4177	7.9410	4.8650
3	3.2469	12.3137	4.3628	10.1836	3.4512	11.7199	5.9079	7.4338
4	2.1116	16.0508	2.6805	14.4147	2.1235	15.9383	3.9910	10.8409
5	1.7588	17.6387	1.6751	18.4982	1.3870	19.6379	2.7326	14.1309
6	1.4205	19.4940	1.0785	22.3223	1.0235	22.2772	2.0632	16.5717
7	1.1707	21.1743	0.6937	26.1559	0.7594	24.8695	1.6376	18.5785
8	0.9815	22.7050	0.4903	29.1693	0.6025	26.8805	1.2866	20.6735
9	0.8204	24.2622	0.3588	31.8814	0.5171	28.2075	1.0075	22.7978
10	0.7044	25.5872	0.2852	33.8777	0.4622	29.1828	0.7898	24.9122

Table 4: The Euclidean norm of the error and the signal to noise ratio for segments #5, #6, #7, and #8 using the new initialization procedure.

		IRS		*Flat*	
Vocoder Type	*kb/s*	*MOS*	*EQ*	*MOS*	*EQ*
G.726(ADPCM)	32	3.77	27.87	3.70	35.00
G.728(LDCELP)	16	3.88	30.38	3.77	35.00
GSM(RPE-LTP)	13	3.63	25.58	3.56	33.25
IS54(VSELP)	8	3.49	23.79	3.47	31.89
source	128	4.10	35.00	4.03	35.00

Table 5: MOS test results for several existing vocoder types [3]

	seg#1			*seg*#2		
bit#	SNR_o	SNR_c	%	SNR_o	SNR_c	%
1	11.7041	2.2567	19.2816	14.1828	2.9729	20.9609
2	16.2364	7.0158	43.2104	18.1943	7.4079	40.7158
3	20.2139	11.5721	57.2481	22.0268	12.0415	54.6675
4	24.0978	15.1072	62.6911	25.3486	15.5708	61.4267
5	27.1619	18.4081	67.7717	28.5015	19.0172	66.7236
6	30.1759	21.0746	69.8391	31.6454	22.2205	70.2171
7	32.4094	23.0066	70.9875	34.3546	25.1480	73.2013
8	34.3550	24.4221	71.0875	36.7348	27.5006	74.8627
9	36.0308	26.1030	72.4464	38.6917	29.1840	75.4270
10	37.6211	27.8036	73.9041	40.4305	30.9153	76.4654

Table 6: The signal to noise ratios for the open loop encoding and for the closed loop reconstructed signal, and the ratio of the latter to the former given in percent. Speech segments #1 and #2.

	$seg\#3$			$seg\#4$		
$bit\#$	SNR_o	SNR_c	%	SNR_o	SNR_c	%
1	14.7564	3.1509	21.3526	15.1423	2.6584	17.5558
2	18.7843	5.8988	31.4029	19.1231	7.2209	37.7602
3	23.5851	11.0247	46.7443	23.3566	11.6208	49.7536
4	28.8222	16.5897	57.5588	27.3273	16.0145	58.6025
5	33.8082	21.1970	62.6978	31.4017	19.5915	62.3898
6	39.2141	26.5661	67.7462	35.1828	23.7752	67.5762
7	45.2169	32.7065	72.3325	38.9523	26.8870	69.0254
8	50.0510	37.4924	74.9084	41.9132	30.0868	71.7838
9	54.7252	41.7704	76.3276	44.2857	32.4149	73.1949
10	59.7176	46.7031	78.2065	46.0070	33.9909	73.8820

Table 7: The signal to noise ratios for the open loop encoding and for the closed loop reconstructed signal, and the ratio of the latter to the former given in percent. Speech segments #3 and #4.

	$seg\#5$			$seg\#6$		
$bit\#$	SNR_o	SNR_c	%	SNR_o	SNR_c	%
1	14.4451	2.5381	17.5703	16.0680	2.5715	16.0040
2	18.3325	8.1861	44.6533	20.2947	7.1206	35.0862
3	21.6119	12.3137	56.9767	23.8674	10.1836	42.6677
4	24.4126	16.0508	65.7480	28.2908	14.4147	50.9517
5	26.6311	17.6387	66.2333	32.5703	18.4982	56.7946
6	28.8033	19.4940	67.6798	36.3803	22.3223	61.3581
7	31.0721	21.1743	68.1456	39.9759	26.1559	65.4293
8	33.1476	22.7050	68.4966	43.1481	29.1693	67.6027
9	35.0450	24.2622	69.2316	45.5918	31.8814	69.9278
10	36.7098	25.5872	69.7013	47.5695	33.8777	71.2172

Table 8: The signal to noise ratios for the open loop encoding and for the closed loop reconstructed signal, and the ratio of the latter to the former given in percent. Speech segments #5 and #6.

	$seg\#7$			$seg\#8$		
$bit\#$	SNR_o	SNR_c	%	SNR_o	SNR_c	%
1	13.9421	2.6128	18.7404	11.0268	1.4402	13.0606
2	18.2153	7.4177	40.7222	14.5319	4.8650	33.4780
3	22.9482	11.7199	51.0712	18.2763	7.4338	40.6746
4	26.6642	15.9383	59.7741	22.0020	10.8409	49.2725
5	30.9059	19.6379	63.5411	24.9070	14.1309	56.7344
6	34.0348	22.2772	65.4542	27.2988	16.5717	60.7049
7	36.9603	24.8695	67.2869	29.5096	18.5785	62.9577
8	39.3135	26.8805	68.3746	31.4522	20.6735	65.7299
9	41.1298	28.2075	68.5818	33.0869	22.7978	68.9027
10	42.3422	29.1828	68.9214	34.9470	24.9122	71.2856

Table 9: The signal to noise ratios for the open loop encoding and for the closed loop reconstructed signal, and the ratio of the latter to the former given in percent. Speech segments #7 and #8.

	SNR_{opt}							
$bit\#$	$seg\#1$	$seg\#2$	$seg\#3$	$seg\#4$	$seg\#5$	$seg\#6$	$seg\#7$	$seg\#8$
1	2.4679	3.0863	3.3356	2.7293	2.7171	2.5986	2.6743	1.6407
2	7.5290	8.0253	6.6427	7.3882	8.2111	7.3181	7.5183	5.0847
3	12.2880	12.5018	11.3505	11.8651	12.5419	10.3847	11.7578	7.8394
4	15.7835	16.0819	16.7848	16.2313	16.2452	15.0043	15.9949	11.4344
5	19.0602	19.6372	21.2640	19.8404	17.9537	19.2594	19.7424	14.6990
6	21.8042	22.6536	26.6003	23.9664	19.8226	23.0186	22.3635	17.3232
7	23.7572	25.5528	32.7834	27.0377	21.6392	26.8645	24.9577	19.6836
8	25.1132	27.8093	37.5280	30.1814	23.1348	29.6485	27.0550	21.5921
9	26.8069	29.4403	41.8666	32.5186	24.6874	32.2450	28.3679	24.0594
10	28.4514	31.1397	46.8472	34.0583	25.9113	34.1923	29.4471	25.8977

Table 10: The SNR for all segments when the norm of the error in the closed loop reconstruction is minimized.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 2000	Technical Memorandum

4. TITLE AND SUBTITLE
Improved Speech Coding Based on Open-Loop Parameter Estimation

5. FUNDING NUMBERS
WU 632-02-00-03

6. AUTHOR(S)
Jer-Nan Juang, Ya-Chin Chen, and Richard W. Longman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681–2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L–17913

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TM–2000–209845

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 39 Distribution: Standard
Availability: NASA CASI (301) 621–0390

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

A nonlinear optimization algorithm for linear predictive speech coding was developed early that not only optimizes the linear model coefficients for the open loop predictor, but does the optimization including the effects of quantization of the transmitted residual. It also simultaneously optimizes the quantization levels used for each speech segment. In this paper, we present an improved method for initialization of this nonlinear algorithm, and demonstrate substantial improvements in performance. In addition, the new procedure produces monotonically improving speech quality with increasing numbers of bits used in the transmitted error residual. Examples of speech encoding and decoding are given for 8 speech segments and signal to noise levels as high as 47 dB are produced. As in typical linear predictive coding, the optimization is done on the open loop speech analysis model. Here we demonstrate that minimizing the error of the closed loop speech reconstruction, instead of the simpler open loop optimization, is likely to produce negligible improvement in speech quality. The examples suggest that the algorithm here is close to giving the best performance obtainable from a linear model, for the chosen order with the chosen number of bits for the codebook.

14. SUBJECT TERMS
Signal Processing, Speech Coding, Data Compression, Signal Prediction

15. NUMBER OF PAGES
22

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	