An extended Levinson-Durbin algorithm and its application in mixed excitation linear prediction

**Dong Xiao**<sup>a,b,*</sup>, **Fuyuan Mo**<sup>b</sup>, **Yan Zhang**<sup>a,b</sup>, **Min Zhao**<sup>c</sup>, **Li Ma**<sup>a,b</sup>

<sup>a</sup>Key Laboratory of Underwater Acoustic Environment, Institute of Acoustics, Chinese Academy of Sciences, China
<sup>b</sup>Institute of Acoustics, Chinese Academy of Sciences, China
<sup>c</sup>Economic Research Institute, State Grid Jibei Electric Power Company Limited, China

*Corresponding author.
E-mail address: xiaodong@mail.ioa.ac.cn (D. Xiao).

**Abstract**

Ten order all-pole model is used in the 2400 bit/second mixed excitation linear prediction to describe human vocal tract. Traditional Levinson-Durbin algorithm is one of the methods to solve the Yule-Walker equations conducted by the ten order linear prediction model. Taking the iteration step of traditional Levinson-Durbin algorithm as 1, an extended algorithm with any positive integer iteration step which is no larger than the order of Toeplitz matrix is proposed. The extended algorithm considers interaction between the adjacent substracts. A hybrid algorithm of the extended algorithm and traditional algorithm has been applied to solve the 2400 bit/second mixed excitation linear prediction under some conditions. The perceptual evaluation of speech quality mean opinion score of nasal syllable is improved in some degree.

Keywords: Acoustics, Computer science

**1. Introduction**

Linear predictive coding has been widely used in low-bit-rate speech coders. Desired results are achieved by the use of an all-pole model in this algorithm to simulate the
vocal tract in producing voiced sounds. However, as the nasal cavity opens and the end of the vocal tract branches during nasal sound production, a pole-zero or higher-order all-pole model is necessary to model the vocal tract. The lower-order all-pole model used in low-bit-rate speech coding cannot accurately simulate the vocal tract, making the sound quality of synthesized nasal syllables less ideal [1]. Liu et al. developed an improved split-vocal-tract model to analyze nasal sounds in a better manner but did not apply it to low-bit-rate speech coders [1].

The Levinson-Durbin algorithm, in addition to being used in linear predictive coding [2], is widely adopted in many signal processing situations, such as active noise control [3], autoregressive model estimation [4], wave propagation modeling in layered media [5], acoustic echo cancelling [6], maximum entropy spectrum estimation [7], and minimum mean-square error equalizer [8], etc. The underlying concept is to solve the Yule-Walker equation recursively.

\[
\begin{bmatrix}
R_0 & R_1 & R_2 & \cdots & R_{M-1} \\
R_1 & R_0 & R_1 & \cdots & R_{M-2} \\
R_2 & R_1 & R_0 & \cdots & R_{M-3} \\
\vdots & \vdots & \vdots & \cdots & \vdots \\
R_{M-1} & R_{M-2} & R_{M-3} & \cdots & R_0
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
a_3 \\
\vdots \\
a_M
\end{bmatrix}
= 
\begin{bmatrix}
b_1 \\
b_2 \\
b_3 \\
\vdots \\
b_M
\end{bmatrix}
\]  

(1)

Without losing generality, the Yule-Walker equation can be written as in (1). The first matrix on the left is the Toeplitz matrix, which is written in vector form as in (2), with bold letters having arrow above representing vectors, and those without arrows indicating the matrices.

\[
R_M \cdot \overrightarrow{A}_M = \overrightarrow{B}_M
\]  

(2)

There are also many improved versions of the Levinson-Durbin algorithm. Yu et al. proposed a multi-stage Levinson-Durbin algorithm, which gives more numerically robust results for input signals with high spectral dynamics [9]. Delsarte et al. generalized the Levinson-Durbin algorithm for Hermitian Toeplitz matrices with any rank profile [10]. Frakt et al. generalized the algorithm for covariance extension and applied the method to multiscale autoregressive modeling [11].

The step size of the conventional Levinson-Durbin algorithm is 1 during iterations, while that for the direct computation of the inverse matrix is \(M\), the order of the matrix. Based on these, the conventional Levinson-Durbin algorithm is generalized here to change the recursion step size to a positive integer \(s\) \((0 < s < M)\), which is variable during recursion. This generalized algorithm is then applied to linear predictive coding, and found to enhance the sound quality of nasal syllables to a certain extent. The subjective perception of the sounds and the PESQ MOS score are both improved [12].
2. Theory

In the following case, a known column vector \( \overrightarrow{A}_m = [a_{m,1}, a_{m,2}, \ldots, a_{m,m}]^H \) (\( m < M \), \( H \) represents the transpose operation) satisfies (3):

\[
\begin{bmatrix}
R_0 & R_1 & R_2 & \cdots & R_{m-1} \\
R_1 & R_0 & R_1 & \cdots & R_{m-2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
R_{m-1} & R_{m-2} & R_{m-3} & \cdots & R_0
\end{bmatrix}
\begin{bmatrix}
a_{m,1} \\
a_{m,2} \\
\vdots \\
a_{m,m}
\end{bmatrix}
= 
\begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_m
\end{bmatrix}
\]

in which the first \( m \) in the subscript of \( a \) is the number of iterations, and is increased by 1 in conventional Levinson-Durbin to make the next prediction. In generalized Levinson-Durbin algorithm however, \( m \) increases by \( s \) (\( s \) is a positive integer). For the convenience of representation, \( n = m + s \leq M \), then (3) is expanded to:

\[
\begin{bmatrix}
R_0 & R_1 & \cdots & R_{m-1} & R_m \\
R_1 & R_0 & \cdots & R_{m-2} & R_m \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
R_{m-1} & R_{m-2} & \cdots & R_0 & R_1 \\
R_m & R_{m-1} & \cdots & R_1 & R_0
\end{bmatrix}
\begin{bmatrix}
a_{n,1} \\
a_{n,2} \\
\vdots \\
a_{n,m} \\
a_{n+1,1} \\
\vdots \\
a_{n,n}
\end{bmatrix}
= 
\begin{bmatrix}
b_{n,1} \\
b_{n,2} \\
\vdots \\
b_{n+1,1} \\
b_{n,n}
\end{bmatrix}
\]

Eq. (4) is split at the straight line and written in vector form.

\[
\begin{bmatrix}
R_m \\
I_m \\
A_m^H \\
r_m
\end{bmatrix}
\begin{bmatrix}
\overrightarrow{A}_m \\
\overrightarrow{a}_m
\end{bmatrix}
= 
\begin{bmatrix}
\overrightarrow{B}_m \\
\overrightarrow{b}_m
\end{bmatrix}
\]

In (5), if we let

\[
L_m = 
\begin{bmatrix}
R_1 & R_2 & \cdots & R_s \\
R_2 & R_3 & \cdots & R_{s+1} \\
\vdots & \vdots & \ddots & \vdots \\
R_m & R_{m+1} & \cdots & R_{n-1}
\end{bmatrix}
\]

and denote the matrix, which has the columns of \( L_m \) reversed as \( I_m \), (5) can be decomposed into:

\[
\begin{cases}
R_m \cdot \overrightarrow{A}_m + I_m \cdot \overrightarrow{a}_m = \overrightarrow{B}_m \\
I_m^H \cdot \overrightarrow{A}_m + r_m \cdot \overrightarrow{a}_m = \overrightarrow{b}_m
\end{cases}
\]

It is known from the previous iteration result, i.e. (3) that \( R_m \cdot \overrightarrow{A}_m = \overrightarrow{B}_m \). If \( R_m \cdot \overrightarrow{A}_m = \overrightarrow{B}_m \) and the matrix with the columns of \( Z_m \) reversed is \( S_m \), then \( R_m \cdot S_m = I_m \). After simple transformation, the following is obtained:
\begin{align}
\begin{cases}
\mathbf{A}_m &= \mathbf{A}_m - \mathbf{S}_m \cdot \mathbf{a}_m \\
\mathbf{a}_m &= (r_m - r_{m,1} \mathbf{S}_m)^{-1} (b_m - r_{m,1} \mathbf{A}_m)
\end{cases}
\end{align}
(6)

It follows that \( \mathbf{A}_{m+s} = [\mathbf{A}_m^H \mathbf{a}_m^H]^H \). The \((\bullet)^{-1}\) in (6) represents the computation of the inverse matrix of \((\bullet)\). The same is true below. If the matrix involved has a full rank, then the scenario of matrices not having a full rank and thus not being invertible is not part of the discussion here.

With \( \mathbf{S}_m \) still unknown in the equation, iteration is used again to solve for it. For clarity of description, \( \mathbf{L}_m \) and \( \mathbf{Z}_m \) are both decomposed into \( s \) number of column vectors, which are \( \mathbf{L}_m = [\mathbf{L}_{m,0} \mathbf{L}_{m,1} \ldots \mathbf{L}_{m,s-1}] \) and \( \mathbf{Z}_m = [\mathbf{Z}_{m,0} \mathbf{Z}_{m,1} \ldots \mathbf{Z}_{m,s-1}] \). With \( \mathbf{R}_m \cdot \mathbf{Z}_m = \mathbf{L}_m \) can then be decomposed into \( s \) independent equations.

\[ \mathbf{R}_m \cdot \mathbf{Z}_{m,i} = \mathbf{L}_{m,i} \]

\( i = 0, 1, \ldots, s-1 \) in the equation represents the \( i^{th} \) column. Matrix segmentation and variable naming as in (4), yield the following:

\[ \begin{cases}
\mathbf{R}_{m-s} \cdot \mathbf{Z}_{m-s,i} + \mathbf{I}_{m-s} \cdot \mathbf{z}_{m-s,i} = \mathbf{L}_{m-s,i} \\
\mathbf{I}_{m-s}^H \cdot \mathbf{z}_{m-s,i} + \mathbf{r}_{m-s} \cdot \mathbf{z}_{m-s,i} = \mathbf{I}_{m-s,i}^H
\end{cases} \]

Upon completion of the previous iteration, it is known that \( \mathbf{R}_{m-s} \cdot \mathbf{Z}_{m-s} = \mathbf{L}_{m-s} \), the following can be solved:

\[ \begin{cases}
\mathbf{z}_{m-s,i} = \mathbf{z}_{m-s,i} - \mathbf{S}_{m-s} \cdot \mathbf{z}_{m-s,i} \\
\mathbf{z}_{m-s,i} = (\mathbf{r}_{m-s} - \mathbf{I}_{m-s}^H \mathbf{S}_{m-s})^{-1} (\mathbf{I}_{m-s,i} - \mathbf{I}_{m-s}^H \mathbf{z}_{m-s,i})
\end{cases} \]

and \( \mathbf{Z}_{m,i} = [\mathbf{Z}_{m,i}^H \mathbf{Z}_{m,i}^H]^H \), \( i = 0, 1, \ldots, s-1 \). The iteration on \( \mathbf{Z}_m \) can also be carried out without decomposing into \( s \) columns, but the explanation is relatively lengthy. One final point is that for constant \( s \), \( \mathbf{r}_m = \mathbf{r}_{m-1} = \cdots = \mathbf{r}_0 \).

For varying \( s \), such as in \( m + ks = M \), in which \( m \) and \( K \) are both non-negative integers and \( s \) is a positive integer, the first \( m \) steps can be regarded as conventional Levinson-Durbin algorithm with \( s = 1 \), and the next \( k \) steps as generalized Levinson-Durbin algorithm with \( s \neq 1 \). The iteration is briefly described as follows:

1. Initialization: for a given \( M \), \( R(i), i = 0, 1, 2, \ldots, M \) is computed, as well as \( m + ks = M \), where \( m \) and \( k \) are non-negative integers and \( s \) is a positive integer. The first \( m \) steps are calculated by conventional Levinson-Durbin algorithm to give the results \( \mathbf{Z}_m \) and \( \mathbf{A}_m \). The number of iterations is set to be \( m \).
2. \( \mathbf{A}_{m+s} \) is calculated as in (6).
3. For \( m + s < M \), \( \mathbf{Z}_{m+s,i}, i = 0, 1, 2, \ldots, s-1 \) is computed according to Eq. (7) to give \( \mathbf{Z}_{m+1} \). Attention needs to be paid on the differences among the subscripts.
4. For $m = m + s$, if $m < M$, the iteration is repeated from step 2. Otherwise, it is ended and the condition $m = M$ should be met by now.

5. $\overrightarrow{A}_M$ is given as the output.

The generalized Levinson-Durbin algorithm is equivalent to the conventional one when $s = 1$ or $k = 0$. When $m = 0$, $k = 1$ and $s = M$ simultaneously, it is a just a calculation for the inverse matrix. As $s$ increases, the inverse matrix operation in (7) makes the iteration result of the ensuing $k = (M - m)/s$ steps less sophisticated and, to a certain extent, affects the result of the first $m$ iterations. It is also likely that the inverse matrix would not exist because the matrix does not have a full rank, in which case the singular-value decomposition could be used to calculate the pseudo inverse matrix, with greater computational effort. If the generalized Levinson-Durbin algorithm must be used, whitening could be performed on $R_M$ and $\overrightarrow{B}_M$ to reduce the singularity of the matrix.

3. Experimental

Generally, due to the inverse matrix operation, the prediction error of generalized Levinson-Durbin algorithm is larger than the conventional one. The prediction error in this case is the sum of the absolute values of the differences between the linear predictive coding results and the data in their original dimensions. Nevertheless, after many trials and parameter adjustments, it is found that the objective rating scored by Levinson-Durbin algorithm on nasal sound synthesis is improved at $M = 10$, $m = 6$, $s = 2$ and $k = 2$.

Experimentations with the mixed excitation linear prediction (MELP) algorithm [13] show that if one of the following three conditions is satisfied, the conventional Levinson-Durbin algorithm is used. For all other cases, the generalized Levinson-Durbin algorithm is used.

1. The prediction error of the generalization algorithm is larger than that of the conventional one multiplied by $2.0 \times 10^5$.
2. The generalized algorithm cannot solve for the 10-dimensional vector Line Spectrum Frequency (LSF).
3. Matrix inversion is not possible in the generalized algorithm as the matrix does not have a full rank.

This new algorithm combining the generalized and conventional Levinson-Durbin recursion is named the hybrid Levinson-Durbin algorithm here. Its comparison with the conventional Levinson-Durbin in terms of performance comparison, is shown in Table 1. The voice samples in Table 1 are described below.
Table 1. PESQ MOS comparison between hybrid and conventional Levinson-Durbin recursion.

|                      | Chinese male voice | Chinese female voice | English male voice | English female voice | Recorded voice of the author |
|----------------------|--------------------|----------------------|--------------------|----------------------|-----------------------------|
| Conventional         | 3.125              | 2.897                | 3.216              | 3.172                | 3.217                       |
| Hybrid               | 3.234              | 2.884                | 3.215              | 3.178                | 3.212                       |

1. The Chinese male voice “zhe xiao fu qì lia’er zheng zai náo bie niú (the young couple is in bad mood with each other (with erhua sound))” running for 1.74s.

2. The Chinese female voice “ah ai an ao bai ban biao bie bian bo bu (sounds with no clear meaning in English) running for 6.18s.

3. The English male voice “They took the cross town bus. I use ketchup on fish. The goose laid an odd egg. That quiz was much too hard.” Running for 13.14s.

4. The English female voice “That is the oldest wine. Line up at the screen door. Tom and Thomas discussed. We watch the new program.” Running for 9.09s.

5. The 10.128s recorded voice of the author “yi er san si wu liu qi ba jiu shi (one two three four five six seven eight nine ten).”

The English voice samples are test sentences complying with the G.723 standard and contain 60% voice signals and 40% silence or low-level noise. These sound materials are representative in coverage and reflect the performance of the coding algorithm reliably. The Chinese male and female voices include plosives, nasal sounds, unpredicted syllables, and erhua sounds and have almost no pause. The performance of most low-bit-rate coding algorithms with these samples is not good. The recorded voice of the author has higher noise level and longer pause. The above sound materials are all representative on their own account. The perceptual evaluation of speech quality mean option score (PESQ MOS) mentioned in Table 1 is based on the ITU-T P.826 standard. It is not difficult to see in the table that in terms of the PESQ MOS, the Chinese male voice “zhe xiao fu qi lia’er zheng zai náo bie niú (the young couple is in bad mood with each other)” with more nasal and erhua sounds receives a score 0.1 point higher when the hybrid algorithm is used, as compared to the conventional one. The other voice samples do not score differently under the two algorithms, which is also true for the auditory perception.

Following the comparison in Table 1, the conventional and hybrid algorithms are each used to process and synthesize voices. The speech spectra of the synthesized voices are shown in Fig. 1. The voice materials used in the figure are Chinese male voices. The upper panel is the original speech, while the middle and lower panels are the synthesized voice using conventional and hybrid recursion algorithms. The darker colored parts of the speech spectra are spectral lines with higher energy. Ten regions, where the energy tends to concentrate, are clearly identified along the
Fig. 1. Comparison of the speech spectra for voices synthesized using the conventional and hybrid algorithm.
time axis (horizontal axis) of the spectral lines, which correspond to the ten words in the Chinese male voice. Through a detailed comparison of the three panels, it is noticed that in the two words “nao” and “niu,” the sound synthesized by the hybrid algorithm is more similar to the original speech than that by conventional algorithm, particularly for the low frequencies.

To validate the strength of the hybrid algorithm further, the Chinese male voice is divided into two parts. The first “zhe xiao fu qi lia’er (the young couple (with erhua sound))” is denoted as “erhua sound,” which emphasize the synthesis of the erhua. The second part “zheng zai nao bie niu (in bad mood with each other)” is the “nasal sound” part, with emphasis on the synthesis of the nasal sounds. The processing results are shown in Table 2. Additionally, the results of the hybrid algorithm tested on some nasal words and sentences selected from TIMIT corpus have been also listed in Table 2. The selected nasal words are “mood, the number of, rain, made, my, traffic, bomb, more, not, corner, major, the manner, unknown, meanwhile, map, nearby, nuclear, now, seven, museum, on, bone, in the winter, given, equipment, train, from, mold, and, down, make, needle, problems, nature, single, contained, unit, begin, examination, chemical, name, market, planning, milk, noun, overnight, farmer, routinely, much, not, only, knew, bring, me, time, move, around, American, newspaper, season, between, summer, belong.”. And the selected sentences are “Frequently has failed to measure up to engineer’s rosy estimates. The desire and ability to read are important aspects of our cultural life. So rules we made, in unabashed collusion. The long and ever-increasing column of sportsmen is now moving into a new era. Naturally, no woman can ever completely monopolize the sexual initiative. It was the story of the rhinoceros fight all over again.”

It can be seen from the table that the hybrid algorithm is slightly weaker in the processing of erhua sound, but clearly advantageous in the generation of nasal sounds. The PESQ MOS of corpus selected from TIMIT is not as good as Chinese corpus. The reason may be that the nasal syllables in English last shorter time than in Chinese, which makes the hybrid algorithm harder to recognize the nasal syllables.

As specifically for the word “nao,” a comparison of the synthesized speech spectrum for conventional and hybrid algorithms is shown in Fig. 2. Though not obvious in the image, the variances between the spectrum of the sounds synthesized by the two algorithms and the original sound material are 64.54%

|                  | Erhua sound | Nasal sound | Selected TIMIT |
|------------------|-------------|-------------|----------------|
| Conventional     | 3.170       | 3.289       | 3.295          |
| Hybrid           | 3.130       | 3.361       | 3.315          |
The correlation coefficients between the synthesized amplitude spectrum by two Levinson-Durbin algorithms and the original amplitude spectrum is 0.8957 (conventional), 0.9010 (hybrid) respectively. The hybrid algorithm gives results closer to the original sound in terms the speech amplitude spectrum.

From the above experimental results in Tables 1 and 2, and Figs. 1 and 2, though the hybrid Levinson-Durbin algorithm has orders of magnitude higher in computational workload than conventional one and the quality of the synthesized speech does not change much, some improvement is seen in the processing of nasal sounds as compared with conventional linear predictive model.

This situation may be explain by the followings:

1. When $s = 1$, the traditional voice tract which introduces to linear prediction model does not consider the interaction between the adjacent subtructs. When $s > 1$, the inverse matrix computation makes the interaction between the adjacent subtructs to be considered, which brings non-linear component in and breaks the all-pole model someway.

2. Although it’s not accuracy, the parameters “$M = 10$, $m = 6$, $s = 2$ and $k = 2$” and the 3 conditions above pick out the speech frames which can be described by the hybrid algorithm. They describes the vocal tract with nasal branches to some extent and makes the hybrid algorithm adapt to the nasal sound.

**Fig. 2.** Comparison of speech spectrum generated by conventional and hybrid algorithm.
4. Conclusion

While the all-pole model used in low-bit-rate linear predictive coding describes the spectra of voiced sounds well, the branching of the vocal tract during the formation of nasal sounds makes it difficult for the lower-order (<20) all-pole model to simulate the spectra of nasal voices accurately. A generalized Levinson-Durbin algorithm is proposed in this paper, which extends the iteration step size of the conventional Levinson-Durbin algorithm from 1 to any positive integer not greater than the order of the matrix. Relevant derivations are then carried out. When the iteration step is 1, the generalized algorithm is the conventional Levinson-Durbin algorithm. When the iteration step equals the matrix order, the generalized algorithm is only a matrix inversion procedure. The generalized and conventional algorithms are used together in linear predictive coding. With the size of the first six iterations set at 1, and the next two iterations set at 2, the quality of nasal syllables in synthesized speech is improved to a certain extent.

Declarations

Author contribution statement

Dong Xiao: Performed the experiments; Analyzed and interpreted the data; Wrote the paper.

Fuyuan Mo: Conceived and designed the experiments.

Yan Zhang: Performed the experiments; Analyzed and interpreted the data.

Min Zhao, Li Ma: Contributed reagents, materials, analysis tools or data.

Funding statement

This work was supported by National Natural Science Foundation of China (61302109) and Youth Talent Project of Institute of Acoustics (QNYC201702).

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] M.S. Liu, A. Lacroix, Improved vocal tract model for the analysis of nasal speech sounds, in: Proc. ICASSP, Atlanta, Georgia, USA, 1996, pp. 801–804.
[2] B. Fazlali, M. Eshghi, A pipeline design for implementation of LPC feature extraction system based on Levinson-durbin algorithm, in: Proc. ICEE, Tehran, Iran, 2011, pp. 1–5.

[3] S. Tyagi, V. Katre, N.V. George, Online estimation of secondary path in active noise control systems using Generalized Levinson Durbin algorithm, in: Proc. DSP, Hong Kong, China, 2014, pp. 552–555.

[4] L.M. Lee, H.C. Wang, An extended Levinson-Durbin algorithm for the analysis of noisy autoregressive process, IEEE Lett. Signal Process. 3 (1) (1996) 13–15. S 0093-9994(95)12784-12789.

[5] Y. Bistritz, Y. Segalov, Integer Levinson algorithms for toeplitz and certain toeplitz-like matrices, in: Proc. ACC, Baltimore, Maryland, USA, 2010, pp. 5720–5725.

[6] S. Maddala, Y. Maheswar, Acoustic echo canceller for teleconferencing systems using Levinson algorithm, in: Proc. INTERACT, 2010, pp. 92–95. Chennai, India.

[7] C. Qing, C. K. Liu. Properties and recursive algorithms of toeplitz generalized inverse, J. Tianjin Univ. Commer. 1 25–36.

[8] M. Kim, J.Y. Lee, Y. Kim, Implementation of the Levinson algorithm for MMSE equalizer, in: Proc. ISOCC, 2008. Busan, Korea (south), pp.III-15 – III-16.

[9] R. Yu, X. Lin, C.C. Ko, A multi-stage Levinson-Durbin algorithm, in: Proc. ASILOMAR, 2002, pp. 218–221. Pacific Grove, CA, USA.

[10] P. Delsarte, Y. Genin, Y. Kamp, A generalization of the Levinson algorithm for Hermitian Toeplitz matrices with any rank profile, IEEE Trans. ICASSP 33 (4) (1985) 964–971.

[11] A.B. Frakt, H. Lev-Ari, A.S. Willsky, A generalized Levinson algorithm for covariance extension with application to multiscale autoregressive modeling, IEEE Trans. Inf. Theory 49 (2) (2003) 411–424.

[12] ITU-T standard P.826. Perceptual Evaluation of Speech Quality (PESQ): an Objective Method for End-to-end Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs, 2001.

[13] ITU-T standard, Federal Information Processing Standards Publication, Analog to Digital Conversion of Voice by 2,400 Bit/second Mixed Excitation Linear Prediction (MELP), 1997.