An Analysis of Finite Precision Effects for the Autocorrelation Method of Linear Prediction

Zongmyung Rhee

S. T. Alexander

Center for Communications and Signal Processing

N. C. State University

Raleigh, NC 27695 - 7914

CCSP-TR-85/10
Abstract

This paper derives new analytical results for predicting the performance degradation induced by finite precision arithmetic upon the autocorrelation method for linear prediction. The primary result is an analytical derivation of the resulting error in the reflection coefficient computation due to finite precision (FP) arithmetic implementation. The result shows that the error in the $i$th reflection coefficient, $\Delta k(i)$, is inversely proportional to the mean square prediction error at the previous ($i-1$)st stage and is proportional to the AR coefficient at the previous ($i-1$)st stage and the roundoff error.

From analysis of the 2nd order system, it is shown that FP effects are more dominant for signals having spectral poles which are near the unit circle and the real axis in z plane. In this circumstance, analytical results show that the autocorrelation method has severe degradation due to FP implementation.

From the analytical computation of reflection coefficient error derived, the occurrence of the system instability may be predicted. Experimental results are presented which show very close agreement between the analytical prediction of the occurrence of the instability ($|k(i)| > 1$) and experimental results showing the same effect.

1. Introduction

During the past decade, methods of linear prediction have been widely used for speech analysis, synthesis, and recognition. There are at least three popular methods of linear prediction: (1) the autocorrelation method, (2) the covariance method and (3) Burg's method. The autocorrelation method is perhaps the most widely used, partly because of the resulting Toeplitz form of the
resulting matrix equation. Given the Toeplitz form, efficient algorithms such as Levinson's recursion [1] and Durbin's algorithm [2] exist for solving the resulting linear equations. Further and perhaps most important, the all-pole synthesis filter produced by the autocorrelation method may be shown to be unconditionally stable [3] in a theoretical sense. Another property of the autocorrelation method is that the Burg and covariance methods are asymptotically equivalent to the autocorrelation method as the number of speech samples in the analysis frame increases [8]. Another work by Gray and Wong shows that Burg's method is at best equal to the autocorrelation method [4].

Despite the frequent usages of the autocorrelation method, however, little analytical attention has been given to investigation of finite precision (FP) arithmetic effects on these related linear prediction algorithms. The main objective of this paper is to analytically examine the effects of using finite word length arithmetic in the autocorrelation method of linear prediction. Such analysis has substantial impact on high speed, real-time implementations of linear prediction (i.e., spectral analysis for radar or beamforming) in which shorter wordlength fixed point processors must be used.

Hertofore, simulation studies have been the primary tool for examining finite wordlength effects. Markel and Gray [5] examined the instability of the autocorrelation method for implementing a vocoder system using fixed point arithmetic simulation. These simulation results showed that implementing the autocorrelation method using finite precision arithmetic often results in instability. Other authors have mentioned [6] that, in practice, the autocorrelation method has been observed to occasionally produce instability. However, no analytical work to date has successfully examined the degradation due to the finite precision implementation of the autocorrelation method. Specifically,
no analytical results are available which quantify the degradation, and eventually, instability of the synthesis filter as a function of the wordlength of the arithmetic used.

As shown in [6], the condition that \(|k(i)| < 1\), the magnitude of the \(i\)th reflection coefficient, is less than one defines the necessary and sufficient condition for the stability of the resulting all-pole filter. For infinite precision computation, the autocorrelation method using Durbin's algorithm guarantees \(|k(i)| < 1\). However, degradation due to FP effects can produce \(|k(i)| > 1\) for situations in which the infinite precision filter was indeed stable. Therefore, examining the value of the reflection coefficients produced by Durbin's algorithm in the autocorrelation method makes it possible to check the resulting synthesis filter for stability. The examination in this paper is limited to fixed point arithmetic which is often used in high speed digital signal processing applications. The analytical investigation of potential instability due to combinations of wordlength and signal statistics are the main examinations of this paper.

In this paper the following results are presented:

1) Using fixed point arithmetic an expression for the \(i\)th stage reflection coefficient error due to FP effects is derived. Using the derived results, the degree of degradation and the possibility of system instability are then calculable.

2) Examples which make the FP implementation of Durbin's algorithm unstable are presented. The interaction between signal statistics and number of bits in the wordlength is derived and examined.
Simulations are used to check the theoretical predictions. Examples are presented which display very close agreement between the analytical prediction of the occurrence of the instability and the experimental result.

An outline of this paper is as follows. Durbin's algorithm and its quantized version are described in Section 2. The general reflection coefficient error [equations (8)] is derived in Section 3. Section 4 shows the comparison between the theoretical derivation and the experimental results to verify the theoretical analysis. Finally, summarizing comments on the analysis are presented in Section 5.

2. Durbin's Algorithm

Durbin's algorithm [2] is used quite frequently to solve the Toeplitz system of equations resulting from the autocorrelation method. In this section, Durbin's algorithm and its quantized version are described. Since the focus is upon degradation due to roundoff errors, the input to Durbin's algorithm will be assumed to be appropriately scaled such that overflow does not occur. Therefore with no loss of generality the problem may be examined using the normalized autocorrelation coefficient. This is beneficial since it removes the effect of the large numerical values due simply to data values, and thus more clearly displays numerical effects due to the algorithm itself. Therefore, dividing the autocorrelation coefficients, \( R(i) \), by \( R(0) \) produces the normalized autocorrelation coefficient \( r(i) \):

\[
r(i) = \frac{R(i)}{R(0)}
\]

(1)

which is useful for the fixed point arithmetic implementation since \( |r(i)| \leq 1 \) [6].
2.1. Infinite Precision

If the signal \( s(n) \) is assumed to be zero for \( n<0 \) and \( n>N \) (e.g., by multiplying it by a finite window), the error minimization of the \( p \)th order linear predictor results in the set of equations

\[
\sum_{k=1}^{p} a_k R(\mid i-k \mid) = R(i) \quad 1 \leq i \leq p
\]  

(2)

where

\[
R(i) = \sum_{n=0}^{N-1-i} s(n)s(n+i)
\]

(3)

The set of equations in (2) can be recursively solved for the predictor coefficients \( a_k \). Making use of the normalized autocorrelation coefficient \( r(i) \), this recursive solution, which is attributed to Durbin, is as follows [2]:

\[
E(0) = r(0) = 1
\]

(4a)

For \( 1 \leq i \leq p \):

\[
r(i) = r(i) - \sum_{j=1}^{i-1} a_j (i-1) r(i-j)
\]

\[
k(i) = \frac{r(i)}{E(i-1)}
\]

(4b)

\[
\alpha_i(i) = k(i)
\]

(4c)

For \( 1 \leq j \leq i-1 \):

\[
a_j(i) = a_j(i-1) - k(i)\alpha_{i-j}(i-1)
\]

(4d)

\[
E(i) = (1 - k(i))^2 E(i-1)
\]

(4e)

Equation (4b)-(4e) are solved recursively for \( i=1,2,\ldots,p \). The final solution is given by
\[ \alpha_j = \alpha_j(p) \quad 1 \leq j \leq p \quad (4f) \]

The filter \( H(z) \) with the predictor coefficients obtained from \( (4f) \) is always stable, i.e., the poles of \( H(z) \) lie inside the unit circle in the \( z \) plane. Hence, for exact computation, \( \left| k(i) \right| \) is always bounded by unity.

2.2. Finite Word Length Implementation

For purposes of realization in actual hardware, the sampled data signal and intermediate values for the algorithm computations must be stored in finite word length registers. In order to analyze this procedure all the parameters must be therefore quantized to the required precision. In the following, \( Q[x] \) represents a quantized version of \( x \), that is,

\[ \hat{x} = Q[x] = x + \eta \]

where \( \eta \) is the roundoff error.

Using this notation, \( (4) \) can be modified as a quantized version:

\[ \hat{E}(0) = r(0) = 1 \quad (5a) \]

For \( 1 \leq i \leq p \):

\[ \hat{y}(i) = Q[\sum_{j=1}^{i-1} \hat{\alpha}_j(i-1)r(i-j)] \quad (5b) \]

\[ \hat{k}(i) = Q[\frac{r(i) - \hat{y}(i)}{\hat{E}(i-1)}] = k(i) + \Delta k(i) \quad (5c) \]

\[ \hat{\alpha}_i(i) = \hat{k}(i) \quad (5d) \]

For \( 1 \leq j \leq i-1 \):
\[ \hat{a}_j(i) = \hat{a}_j(i-1) - Q[k(i)\hat{a}_{i-j}(i-1)] \]  

(5e)

\[ \hat{E}(i) = Q\left\{ \left[ \begin{array}{c} 2 \\ 1 \end{array} \right] k(i) \right\} E(i-1) \]  

(5f)

The inner product \( \hat{y}(i) \) is computed in the following manner: all scalar products are computed without quantization, summed, and the final result is quantized. This is equivalent to the use of a double precision accumulator, which is common in many processors.

In the following analysis, unity in (5f) is assumed to implemented exactly.

3. Derivation of Reflection Coefficient Error

Since we restrict our attention to fixed point arithmetic, addition and subtraction involve no roundoff. However multiplication and division do produce roundoff error. These are commonly modeled as [7]

\[ \hat{c} = Q[AB] = AB + \eta \]

\[ \hat{d} = Q[A/B] = A/B + \eta \]

where \(|\eta| \leq \frac{1}{2}2^{-b}\). Here it is assumed that the data representation is \(b\)-bit (including sign bit) fixed point arithmetic. The exact product of two \(b\)-bit numbers lying in the interval \([-1,1]\) is, in general, a number requiring \(2b\) bits for its representation. This exact product is then rounded, such that a \(b\)-bit word is restored, and this mechanism induces the roundoff error. A similar phenomenon occurs in modeling the division operation. In the case of algorithm having multiple iterations, such as Durbin's algorithm, it will be shown that the accumulation of roundoff error can result in instability of a syn-
thesis filter which would otherwise be stable for infinite precision arith-
metic.

In the following work, quantized parameter values are denoted as the sum of the true value and the roundoff error; for example,

\[
\hat{k}(i) = k(i) + \Delta k(i)
\]

where \(\hat{k}(i)\) is the quantized value and \(\Delta k(i)\) is the error due to the FP effects. As a matter of notation, instantaneous roundoff errors due to quantization will be represented by the symbol \(\eta\) and the resulting parameter error terms will be represented by the symbol \(\Delta\). With a negligible loss in accuracy, products of the form \(\Delta(\cdot)\eta(\cdot), \Delta(\cdot)\Delta(\cdot),\) and \(\eta(\cdot)\eta(\cdot)\) may therefore be neglected, since they will be much smaller than the terms which dominate the degradation phenomena. The resulting simplicity in the analysis however is substantial.

From the first few iterations of the quantized version of Durbin's algo-
rithm, we can obtain by induction a general formulation for reflection coeffi-
cient error at each stage. The derivations are shown in Appendix A, and the final results are

\[
\Delta k(1) = 0 \quad \text{(6a)}
\]

For \(2 \leq i \leq p\),

\[
\Delta k(i) = \frac{\Delta a(i-1)}{E(i-1)} \quad \text{(6b)}
\]

where \(\Delta a(i-1)\) is determined recursively:

\[
\Delta a(1) = k(2)\eta \quad \text{(7a)}
\]

\[
\Delta a(i-1) = - \sum_{m=2}^{i-1} \frac{\Delta a(i-m)}{E(i-m)} b_m(i-1) r(i-1) \quad \text{(7b)}
\]
and \( b_{(i-1)} \) and \( r(i-1) \) are \((i-1)x1\) column vectors given by

\[
\begin{bmatrix}
  r(1) \\
  r(2) \\
  \vdots \\
  \vdots \\
  r(i-2) \\
  r(i-1)
\end{bmatrix}
\]

(7c)

and

\[
\begin{bmatrix}
  0 \\
  \vdots \\
  \vdots \\
  \vdots \\
  -a_{i-1}(i-m) \\
  -a_{i-m}(i-m)
\end{bmatrix}
\]

(7d)

From (6) and (7), it is seen that \( \Delta k(i) \) can now be computed recursively. However, the computation of the reflection coefficient errors for higher orders is quite tedious. However, using the normal equation of (2) simplifies the computation substantially (see Appendix B). The results are

\[
\Delta k(1) = 0 \quad (8a)
\]

\[
|\Delta k(2)| = \frac{1}{E(1)}|k(2)\eta| \quad (8b)
\]

\[
|\Delta k(i)| = \frac{1}{E(i-1)}|a_{i-2}(i-1)k(2)\eta| \quad i \geq 3 \quad (8c)
\]

where

\[
|\eta| \leq \frac{1}{2}^{-b} \quad (8d)
\]
4. Simulation and Results

The recursions of (8a) - (8d) provide an analytical method of computing the degradation in reflection coefficient due to the FP implementation. In its current general form it requires the IP calculation of prediction error \( E(i-l) \) and IP reflection coefficient, \( k(i) \). These may be found from the IP Durbin's algorithm (5), which requires the true autocorrelation coefficients be computed from the AR model.

4.1. Calculation of Autocorrelation Coefficients

The calculation of the autocorrelation coefficients is possible from impulse response of the AR model. To obtain the impulse response, consider an AR process modeled by the difference equation [2]

\[
\begin{align*}
s(n) &= Gx(n) + \sum_{k=1}^{p} a_k s(n-k) \\
\end{align*}
\]

(9)

If \( x(n) \) is the unit sample and \( G=1 \), then \( s(n) \) corresponds to the impulse response \( h(n) \). Hence \( h(n) \) can be iteratively determined as:

\[
\begin{align*}
h(0) &= 1 \\
h(1) &= a_1 h(0) = a_1 \\
h(2) &= a_1 h(1) + a_2 h(0) = a_1^2 + a_2 \\
&\quad \ldots \\
h(p) &= a_1 h(p-1) + a_2 h(p-2) + \ldots + a_p h(0) \\
h(p+1) &= a_1 h(p) + a_2 h(p-1) + \ldots + a_p h(1) \\
&\quad \ldots \\
\end{align*}
\]

(10)

Of course, \( h(n) \) can be obtained from another way, i.e., by long division:
\[ H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \sum_{n=0}^{\infty} h_n z^{-n} \quad (11) \]

where

\[ A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \quad (12) \]

\[ H(z) \] is given by

\[ H(z) = 1 + a_1 z^{-1} + (a_1 + a_2) z^{-2} + \ldots \quad (13) \]

Here the coefficient of each term in \( H(z) \) corresponds to \( h(n) \). From the impulse response, the autocorrelation coefficients are given by

\[ R(i) = \sum_{n=0}^{\infty} h(n) h(n+i) \quad 0 \leq i \leq p \quad (14) \]

In calculation of \( R(i) \), we cannot obtain the exact values unless the entire infinite impulse response is used in the analysis. However, in practice, very good approximation can be obtained by truncating the impulse response at a point where most of the decay of the response has already occurred [6]. In the simulations of this paper \( h(n) \) is truncated beyond \( n=300 \), which contributes negligible error. Additionally, to minimize the effect of rounding in autocorrelation coefficient, the computation is done using double precision variables.

4.2. Situation of Maximum Reflection Coefficient Error

In general, the theoretical analysis of the general \( p \)th order equations is quite difficult and must be done numerically for large order \( p \) (an example will be shown directly). However, a great deal of insight into situations which potentially cause severe FP degradation may be seen by considering the \( p=2 \) order case.
4.2.1. Theoretical Analysis - Second Order Case

To examine the situation of maximum reflection coefficient error, consider the simple case, i.e., the 2nd order system. From (8), the absolute value of reflection coefficient error at the 2nd stage, $|\Delta k(2)|$, has been shown to be

$$|\Delta k(2)| = \left| \frac{k(2)\eta}{E(1)} \right| = \left| \frac{k(2)\eta}{(1-r(1))E(0)} \right|$$

(15)

Note that $E(0) = r(0) = 1$, since the autocorrelation coefficients are normalized. As shown in (15), when the value of $r(1)$ gets closer to unity, it is expected that $\Delta k(2)$ becomes larger and therefore $k(2)$ can be greater than unity for the quantized implementation. Therefore, we need to find out the condition that $r(1) < 1$.

The 2nd order filter with poles shown in Fig. 1 can be represented by

$$H(z) = \frac{1}{(1-p_1z^{-1})(1-p_2z^{-1})}$$

(16)

where

$$p_1 = \text{re}^{j\varphi}$$
$$p_2 = p_1^* = \text{re}^{-j\varphi}$$

Using partial expansion,

$$H(z) = \frac{1}{2j\sin\varphi} \left( \frac{\text{e}^{j\varphi}}{1 - \text{re}^{j\varphi} z^{-1}} - \frac{\text{e}^{-j\varphi}}{1 - \text{re}^{-j\varphi} z^{-1}} \right)$$

(17)

Taking the inverse z transform of $H(z)$, we have

$$h(n) = \frac{r^n}{\sin\varphi} \sin(n+1)\varphi \ u(n)$$

(18)

From (14) with $i=0$, 

From (23), for any given \( r \), when the poles are close to the real axis in the \( z \) plane the absolute value of \( r(1) \) will have maximum value. Additionally when the poles approach the imaginary axis in \( z \) plane the value of \( r(1) \) is almost zero. For a given \( \psi \), the absolute value of \( r(1) \) has the maximum value when \( r=1 \), since the function \( \frac{r}{1+r^2} \) has the maximum value \( \frac{1}{2} \) at \( r=1 \). Therefore, if \( r - 1 \) and \( \psi + 0 \), then \( r(1) - 1 \).

Therefore, from this investigation of the 2nd order system, when the poles are near the unit circle and are near the real axis in \( z \) plane, it is expected that the reflection coefficient error will be large due to FP quantization effects. Therefore, FP effects are more important for the correlated data and for the spectral poles which are near the unit circle and/or the real axis in the \( z \) plane. The qualitative results of this second order system can be readily extended to the higher order signal processes which have poles near the unit circle or the real axis.

4.2.2. Numerical Examples

In order to display the applicability of the preceding analysis, numerical evaluations for the 2nd order system are shown in Table I. The values of \( \Delta k(2) \) in Table I are calculated from (8). The results in (a) are obtained from \( \psi = 5 \) deg. and 85 deg. with \( r=0.95 \). For the case of 5 deg., the absolute value of \( r(1) = k(1) \) is about 12 times larger than for 85 deg. case. \( E(1) \) for 85 deg. case is almost 100 times larger than for 5 deg. case. Therefore, the system for 5 deg. case has more FP degradation at the 2nd stage due to FP quantization.

In (b), the radius is changed to 0.8. As expected, the value of \( r(1) \) is smaller than the case (a). Furthermore, \( k(2) \) is away from unit circle, com-
pared with case (a). Therefore, the FP effects are less important in this case.

Comparing (a) with (b), we see that FP effects are much more important in case that spectral poles are close to unit circle.

Table I. Second-order system comparison according to radius and angle

<table>
<thead>
<tr>
<th>j</th>
<th>k(1)</th>
<th>k(2)</th>
<th>E(1)</th>
<th>Δk(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.9949</td>
<td>-0.9025</td>
<td>0.0102</td>
<td>88.48</td>
</tr>
<tr>
<td>85</td>
<td>0.0870</td>
<td>-0.9025</td>
<td>0.9924</td>
<td>0.91</td>
</tr>
</tbody>
</table>

(a) r=0.95

<table>
<thead>
<tr>
<th>j</th>
<th>k(1)</th>
<th>k(2)</th>
<th>E(1)</th>
<th>Δk(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.9719</td>
<td>-0.64</td>
<td>0.0554</td>
<td>11.55</td>
</tr>
<tr>
<td>85</td>
<td>0.0850</td>
<td>-0.64</td>
<td>0.9928</td>
<td>0.64</td>
</tr>
</tbody>
</table>

(b) r=0.8

4.3. Divergence due to Finite Word Length

To examine the FP effects using the analysis results in Section 3, consider the 8th order system with pole locations shown in Table II. The results of the previous section suggest that the autocorrelation method for this process is severely degraded by FP effects. All spectral poles are near the unit circle and are within ±40 deg. around the real axis. The selected pole locations are the following:
Table II. Pole locations for 8th order example

<table>
<thead>
<tr>
<th>Radius</th>
<th>Angle (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.8</td>
<td>9.02</td>
</tr>
<tr>
<td>.95</td>
<td>18.04</td>
</tr>
<tr>
<td>.65</td>
<td>27.06</td>
</tr>
<tr>
<td>.88</td>
<td>38.34</td>
</tr>
</tbody>
</table>

Only the half of the poles are listed since the others have the complex conjugate values of the corresponding poles. Since these poles are the roots of \( A(z) \) in (12), we can calculate AR coefficients:

\[
A(z) = 1 - \sum_{k=1}^{8} a_k z^{-k} = \prod_{k=1}^{4} (1 - p_k z^{-1})(1 - p_k^* z^{-1})
\]  

(24)

Also the reflection coefficients \( k(i) \) are obtained from AR coefficients \( a_k \) [2]. The AR coefficient values and corresponding reflection coefficient values are computed and shown in Table III. Since the selected poles are inside the unit circle in z plane, the reflection coefficient value should meet the stability condition, \(-1 < k < 1\), which is evident in Table III. For this example, note that the absolute value of reflection coefficient is larger than 0.9 until the 5th stage.

Table III. AR parameter and k coefficient of selected example (8 poles)

<table>
<thead>
<tr>
<th>j</th>
<th>( a_j(8) )</th>
<th>( k_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.924945</td>
<td>0.964728</td>
</tr>
<tr>
<td>2</td>
<td>-15.78852</td>
<td>-0.980748</td>
</tr>
<tr>
<td>3</td>
<td>24.68931</td>
<td>0.946935</td>
</tr>
<tr>
<td>4</td>
<td>-24.76141</td>
<td>-0.916640</td>
</tr>
<tr>
<td>5</td>
<td>16.29885</td>
<td>0.904978</td>
</tr>
<tr>
<td>6</td>
<td>-6.874399</td>
<td>-0.845861</td>
</tr>
<tr>
<td>7</td>
<td>1.699614</td>
<td>0.601383</td>
</tr>
<tr>
<td>8</td>
<td>-0.1889815</td>
<td>-0.188982</td>
</tr>
</tbody>
</table>
As shown in (8), the exact values of mean square prediction error and AR coefficient at each iteration of algorithm are needed to predict the FP performance. To obtain these values, a subroutine for Durbin's algorithm which consists of the double precision variables is executed.

From (8), the reflection coefficient error for \( i \geq 2 \) depends upon the value of \( \eta \), the size of the quantization error. Since \( \eta \) is a random variable which for b-bits has an uniform distribution over \([-\frac{1}{2^b}, \frac{1}{2^b}]\), the value of \( |\Delta k(i)| \) can not be computed exactly. However, using the worst case of \( \eta \), i.e., \( \frac{1}{2^b} \), we can estimate the worst case system performance. For this case we can compute \( |\Delta k(i)| \) from (8) using the exact infinite precision values of AR coefficients and mean square prediction error at each stage.

After computation of \( |\Delta k(i)| \) from (8), \( \hat{k}(i) \) can be computed from the true \( k(i) \) value shown in Table III, since \( \hat{k}(i) = k(i) + \Delta k(i) \). Here, from \( |\Delta k(i)| \), the sign of \( \Delta k(i) \) is selected to make the absolute value larger, which assumes the worst case. For a given word length, the computation is made recursively, stage-by-stage, beginning with the first stage. If the computed \( |\hat{k}(i)| \) is larger than unity at some stage, we regard this stage as the stage at which the algorithm diverges. For the selected example, the computed results are shown in Table III with the simulated results obtained from the b bit uniform quantizer.

Table III. Comparison of divergence prediction according to word length (including sign bit)

<table>
<thead>
<tr>
<th>Word length</th>
<th>Stage that starts to diverge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation</td>
<td>Simulation</td>
</tr>
</tbody>
</table>
Except for the case of 16 bits, the computation gives the very close results. The prediction is based upon the condition that Δk(i) will make the absolute value of k(i) increase. For 16 bit word length simulation Δk(4) decreases the absolute value of k(4). Hence actual divergence starts at the higher stage than at the predicted one. Note that our computations are based upon the limit value of η.

5. Summary

The instability of the finite precision implementation of the autocorrelation method of linear prediction has been examined from an analytical standpoint. A derivation of the reflection coefficient error due to the finite precision arithmetic has been made and shown in (8).

From (8), the reflection coefficient error has been shown to be inversely proportional to the mean square prediction error at the previous stage and is proportional to the AR coefficient at the previous stage and the roundoff error.

Since the mean square prediction error is recursively determined by (4e), the decay of its value depends highly upon E(1), which is related to r(1). Therefore, when the value of r(1) gets closer to unity, the system often shows the instability due to quantization. From the analysis of the 2nd order system, it has been shown that when the spectral poles are near the unit circle and near the real axis, the value of r(1) gets closer to unity. Therefore, FP effects are more important for correlated data and for poles near the unit
circle and the real axis.
Appendix A

This appendix derives equations (7) and (8) for the degradation due to FP effects in the computation of the reflection coefficients.

(1) \( i=1 \)

Using the FP Durbin's algorithm of (5a)-(5f) for the first iteration provides the following results.

\[ \hat{E}(0) = 1 \]
\[ \hat{k}(1) = r(1) = k(1) \]  
\[ \hat{a}_1(1) = \hat{k}(1) = k(1) \]
\[ \hat{E}(1) = 1 - k^2(1) - \eta_{E}(1) = E(1) + \Delta E(1) \]

Therefore at the end of iteration 1,

\[ \Delta k(1) = 0 \]  

(2) \( i=2 \)

From (5c),

\[ \hat{k}(2) = k(2) + \Delta k(2) = \frac{r(2) - r(1)a(1) - \eta(2)}{\hat{E}(1)} + \eta_k(2) \]  

Multiply both sides of (A3) by \( \hat{E}(1) \):

\[ [k(2) + \Delta k(2)]\hat{E}(1) = r(2) - r(1)a(1) - \eta_y(2) + \eta_k(2)\hat{E}(1) \]  

Using \( \hat{E}(1) \) in (A1):

\[ k(2)E(1) + \Delta k(2)E(1) + k(2)\Delta E(1) = r(2) - r(1)a(1) - \eta_y(2) + \eta_k(2)E(1) \]  

But from the IP Durbin's algorithm,
\[ k(2)E(1) = r(2) - r(1)a_1(1) \]  

Therefore, (A3b) simplifies to the expression for \( \Delta k(2) \):

\[
\Delta k(2) = \eta_k(2) + \frac{k(2)\eta_E(1)}{E(1)} - \frac{\eta_y(2)}{E(1)}
\]  

(A4)

The second term in the right hand side is assumed to dominate the first term. This is often valid assumption since for spectral poles are near the unit circle \( |k(2)| \approx 1 \) and \( E(1) \ll 1 \). With this assumption,

\[
\Delta k(2) = \frac{k(2)\eta_E(1)}{E(1)} - \frac{\eta_y(2)}{E(1)}
\]  

(A5)

From (5d),

\[
\hat{a}_2(2) = \hat{k}(2) = k(2) + \Delta k(2)
\]  

(A6a)

and from (5e) and (A3),

\[
\hat{a}_1(2) = a_1(1) - k(2)a_1(1) - \eta_\alpha(2)
\]

(A6b)

\[
= a_1(1) - k(2)a_1(1) - \Delta k(2)a_1(1) - \eta_\alpha(2)
\]

The \( \eta \) term in (A6b) can be neglected since \(|\Delta k(2)a_1(1)| >> |\eta_\alpha(2)|\).

Then from (5f),

\[
\hat{E}(2) = (1-k(2)-\eta_k)E(1) + \eta_E(2)
\]

\[
E(2) + \Delta E(2) = E(1)(1-k^2(2)) - 2k(2)\Delta k(2)E(1) + \eta_E(2)
\]

(A7)

\[ + \Delta E(1)(1-k^2(2)) - E(1)\eta_k \]

But \( E(2) \) is given by the first term on the right side of (A7). Therefore

\[
\Delta E(2) = -2k(2)\Delta k(2)E(1) + \eta_E(2) + \Delta E(1)(1-k^2(2)) - E(1)\eta_k
\]

(A7a)

However, the third term in \( \Delta E(2) \) is negligible since \( \Delta E(1) \) is the order of \( \eta \) and \( k(2) \) is near 1. Also the fourth term is negligible since \( E(1) \ll 1 \). Hence, a
simplification for $\Delta E(2)$ becomes

$$
\Delta E(2) = -2k(2)\Delta k(2)E(1) + \eta_E(2)
$$

(A7b)

(3) $i = 3$

$$
k(3) = \frac{r(3) - \hat{a}_1(2)r(2) - \hat{a}_2(2)r(1) - \eta_y(3)}{E(2) + \Delta E(2)} + \eta_k(3)
$$

(A8)

Doing the similar manipulation as for the case of $i = 2$ produces

$$
\Delta k(3) = -\frac{\Delta k(2)(r(1) - r(2)a_1(1))}{E(2)} + \frac{k(3)\Delta E(2)}{E(2)} - \frac{\eta_y(3)}{E(2)}
$$

(A9)

Let $\Delta k(2)[r(1) - r(2)a_1(1)]$ be $-\Delta a(2)$, then $\Delta k(3)$ can be represented by

$$
\Delta k(3) = \frac{\Delta a(2)}{E(2)} - \frac{k(3)\Delta E(2)}{E(2)} - \frac{\eta_y(3)}{E(2)}
$$

(A10)

Using the matrix notation, we have

$$
\Delta a(2) = -\Delta k(2)\begin{bmatrix}1 & -a_1(1)\end{bmatrix}\begin{bmatrix}r(1) \\ r(2) \end{bmatrix} = -\Delta k(2)\begin{bmatrix}1 & -a_1(1) \\ r(1) & 1 \end{bmatrix}\begin{bmatrix}a_1(2) \\ a_2(2) \end{bmatrix} = -\Delta k(2)\begin{bmatrix}E(1) & 0 \\ E(1) & 1 \end{bmatrix}\begin{bmatrix}a_1(2) \\ a_2(2) \end{bmatrix} = -\Delta k(2)E(1)a_1(2)
$$

(A11)

From (5e), (A6a), and (A6b):

$$
\hat{a}_1(3) = \hat{a}_1(2) - k(3)\hat{a}_2(2) - \eta_{a_1}(3)
$$

$$
\approx \hat{a}_1(2) - k(3)a_2(2) - \Delta k(2)a_1(1) - \Delta k(3)a_2(2) - \Delta k(2)k(3)
$$

$$
\hat{a}_2(3) = \hat{a}_2(2) - k(3)\hat{a}_1(2) - \eta_{a_2}(3)
$$

$$
\approx \hat{a}_2(2) - k(3)a_1(2) + \Delta k(2) + k(3)\Delta k(2)a_1(1) - \Delta k(3)a_1(2)
$$

$$
\hat{a}_3(3) = k(3) = k(3) + \Delta k(3)
$$

(A12)
In the resulting expression for $\tilde{a}_1(3)$ and $\tilde{a}_2(3)$, the $\eta_a$ terms are neglected since $|\Delta k(2)|$ or $|\Delta k(3)| \gg \eta_a$.

(4) $i=4$

$$\Delta k(4) = \frac{\Delta a(3)}{E(3)} - \frac{k(4)\Delta E(3)}{E(3)} - \frac{\eta_y(3)}{E(3)}$$

(A13)

where

$$\Delta a(3) = -\Delta k(3) \begin{bmatrix} 1 & -a_1(2) & -a_2(2) \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix}$$

$$-\Delta k(2) \begin{bmatrix} 0 & 1 & -a_1(1) \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix}$$

$$-k(3)\Delta k(2) \begin{bmatrix} 0 & a_1(1) & -1 \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix}$$

(A14)

From the observation of (A10) and (A13), general formula for $\Delta k(i)$ can be expressed by

$$\Delta k(i) = \frac{\Delta a(i-1)}{E(i-1)} - \frac{k(i)\Delta E(i-1)}{E(i-1)} - \frac{\eta_y(i-1)}{E(i-1)}$$

$i \geq 3$

(A15)

As the iteration $i$ increases, the IP result is that $k(i)$ approaches zero and $\Delta a$ term is greater than $\eta$ term. Therefore, the first term on the right hand side of (A15) dominates the expression for $\Delta k(i)$. Hence $\Delta k(i)$ can be approximated as

$$\Delta k(i) = \frac{\Delta a(i-1)}{E(i-1)}$$

(A16)

Here, to simplify the calculation, we will use the numerator of the first term in (A5) as the initial value. Hence

$$\Delta a(1) = k(2)\eta$$

(A17)
Therefore the final step needed to obtain an expression for the $\Delta k(i)$ is to derive an iterative expression for $\Delta a(i-1)$. To produce this expression, first we examine (A14). In expression (A14), we will neglect the term related to $k(3)$ to simplify the expression. Therefore, we regard the first two terms in (A14) as the expression of $\Delta a(3)$:

$$\Delta a(3) = -\Delta k(3)\begin{bmatrix} 1 & -a_1(2) & -a_2(2) \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix}$$

$$- \Delta k(2)\begin{bmatrix} 0 & 1 & -a_1(1) \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix} \tag{A18}$$

With same procedure, we obtain:

$$\Delta a(4) = -\Delta k(4)\begin{bmatrix} 1 & -a_1(3) & -a_2(3) & -a_3(3) \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ r(4) \end{bmatrix}$$

$$- \Delta k(3)\begin{bmatrix} 0 & 1 & -a_1(2) & -a_2(2) \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ r(4) \end{bmatrix}$$

$$- \Delta k(2)\begin{bmatrix} 0 & 0 & 1 & -a_1(1) \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ r(4) \end{bmatrix} \tag{A19}$$

From the observation of $\Delta a(2), \Delta a(3)$ and $\Delta a(4)$ in (A11), (A18), and (A19), we can induce the general form of $\Delta a(i-1)$:

For $i \geq 3$,

$$\Delta a(i-1) = -\sum_{m=2}^{i-1} \Delta k(i+1-m) b_m^T (i-1) r(i-1) \tag{A20}$$

where $b_m(i-1)$ and $r(i-1)$ are $(i-1) \times 1$ column vectors given by
\[ r(i-1) = \begin{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(i-2) \\ r(i-1) \end{bmatrix} \end{bmatrix} \]

and

\[ b_m(i-1) = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 1 \\ -a_1(i-m) \\ \vdots \\ -a_m(i-m) \\ i-m \end{bmatrix} \]

Making use of (A16), we can obtain another form of \[ \Delta a(i-1) \]:

\[
\Delta a(i-1) = - \sum_{m=2}^{i-1} \frac{\Delta a(i-m) T_b_m(i-1) r(i-1)}{E(i-m)}
\]  

(A21)

which is equation (7b) in the text. As an initial value,

\[
\Delta a(1) = k(2) \eta
\]

(A17)

From (A17) and (A21), \[ \Delta a(i-1) \] can be obtained in a recursive fashion. Once the \[ \Delta a(i-1) \] have been computed, then \[ \Delta k(i) \] can be easily obtained by:

\[
\Delta k(i) = \frac{1}{E(i-1)} \Delta a(i-1)
\]

(A16)

In summary, to obtain the value of \[ \Delta k(i) \], first find the \[ a(i-1) \] using (A17) and (A21) and then divide it by \[ E(i-1) \].
The form in (A21) may be simplified substantially. An example using the simple case of $i=3$ will illustrate the procedure. From the normal equation in (2) and using $r(i)$ instead of $R(i)$:

$$
\begin{bmatrix}
  r(1) \\
  r(2) \\
  r(3)
\end{bmatrix} =
\begin{bmatrix}
  1 & r(1) & r(2) \\
  r(1) & 1 & r(1) \\
  r(2) & r(1) & 1
\end{bmatrix}
\begin{bmatrix}
  a_1(3) \\
  a_2(3) \\
  a_3(3)
\end{bmatrix} =
\begin{bmatrix}
  R(3) \\
  a(3)
\end{bmatrix}
$$

\tag{B1}

where

$$R(3) =
\begin{bmatrix}
  1 & r(1) & r(2) \\
  r(1) & 1 & r(1) \\
  r(2) & r(1) & 1
\end{bmatrix}
$$

and

$$a(3) =
\begin{bmatrix}
  a_1(3) \\
  a_2(3) \\
  a_3(3)
\end{bmatrix}
$$

Using (B1), the first term in (A21) with $i=4$ can be represented as:

$$\frac{\Delta a(2)}{E(2)}\begin{bmatrix}
  1 & -a_1(2) & -a_2(2)
\end{bmatrix}R(3)a(3)
$$

\tag{B2}

Hence (B2) may be simplified by noting that:

$$\begin{bmatrix}
  1 & -a_1(2) & -a_2(2)
\end{bmatrix}R(3) =
\begin{bmatrix}
  E(2) & 0 & 0
\end{bmatrix}
$$

\tag{B3}

Substituting (B3) into (B2), we get:

$$\frac{\Delta a(2)}{E(2)}a_1(3)
$$

\tag{B4}

Similarly, for the second term, we have
\[
\begin{align*}
- \frac{\Delta a(2)}{E(1)} & \begin{bmatrix} 0 & 1 & -a_1(1) \end{bmatrix} R(3) \alpha(3) \\
= & - \frac{\Delta a(1)}{E(1)} \begin{bmatrix} a_1(2)E(1) \end{bmatrix} E(1) \begin{bmatrix} 0 \end{bmatrix} \alpha(3) \\
= & - \frac{\Delta a(1)}{E(1)} \{ a_1(2) a_1(3) + a_2(3) \}
\end{align*}
\]

From (B4) and (B5),

\[
\Delta a(3) = - \Delta a(2) a_1(3) - \Delta a(1) \{ a_1(2) a_1(3) + a_2(3) \}
\]

Since

\[
\Delta a(1) = k(2) \eta
\]
\[
\Delta a(2) = - a_1(2) k(2) \eta,
\]

(B6) can be rearranged as:

\[
\Delta a(3) = - a_2(3) k(2) \eta
\]

Doing the same procedure for the \( \Delta a(4) \), we have

\[
\begin{align*}
\Delta a(4) & = - \Delta a(3) a_1(4) - \Delta a(2) \{ a_1(3) a_1(4) + a_2(4) \} \\
& \quad - \Delta a(1) \{ \{ a_1(2) a_1(3) + a_2(3) \} a_1(4) + a_1(2) a_2(4) + a_3(4) \} \\
& \quad = - a_3(4) k(2) \eta
\end{align*}
\]

By induction we have

\[
\Delta a(i-1) = - a_{i-2}(i-1) k(2) \eta
\]

Therefore, making use of (A2), (A16), and (A17), we have
\[ \Delta k(1) = 0 \]
\[ |\Delta k(2)| = \frac{1}{E(1)} |k(2)\eta| \]
\[ |\Delta k(i)| = \frac{1}{E(i-1)} |a_{i-2}(i-1)k(2)\eta| \quad i \geq 3 \]

where

\[ |\eta| \leq \frac{1}{2^b} \]

The equations (B11) are equations (8) in the text.
References


Fig. 1. Pole-Zero pattern of a second-order system

\[ R(0) = \sum_{n=0}^{\infty} h^2(n) \]  

After some manipulation, we have

\[ R(0) = \frac{4(1+r^2)(\sin^2 \varphi)}{D(r, \varphi)} \]  

where

\[ D(r, \varphi) = 4\sin \varphi \left( 1 - 2r \cos 2\varphi + r^2 \right) \left( 1 - r^2 \right) \]  

From (14) with \( i=1 \),

\[ R(1) = \sum_{n=0}^{\infty} h(n)h(n+1) \]

\[ = \frac{8r \cos \varphi \sin \varphi}{D(r, \varphi)} \]  

Here \( D(r, \varphi) \) has been defined in (21).

Therefore, we obtain \( r(1) \)

\[ r(1) = \frac{R(1)}{R(0)} \]

\[ = 2 \frac{r}{1+r^2} \cos \varphi \]