so that (27) holds. Equations (27) and (28) can be put together as

\[
\begin{bmatrix}
R_{K+1} & W_{K+1}^	ext{opt}
\end{bmatrix} = R_

\begin{bmatrix}
K \\
K+1
\end{bmatrix} \begin{bmatrix}
0 & 0
\end{bmatrix}
\begin{bmatrix}
R_{K+1} & W_{K+1}^	ext{opt}
\end{bmatrix}.
\]

(29)

Notice that we can write (29) also as

\[
P_{K+1} = R_{K+1} \begin{bmatrix}
W_{K+1}^	ext{opt} \\
0
\end{bmatrix} = R_{K+1} \begin{bmatrix}
1 & 0
\end{bmatrix} W_{K+1}^	ext{opt}.
\]

(30)

This is the normal equation (4) for the $K+1$st approximation with the constraint that $w_{K+1}$, the last weight component of the weight vector being zero, while the rest of the weights being the same as that of an order $K$ approximation, i.e., $W_{K+1}^	ext{opt} = [W_1, W_2, \ldots, W_K, 0]$. This implies that at the local extrema performances of order $K$ and order $K+1$ approximations are identical, explaining the fact that $J_{\text{ach}}$ curves are touching each other as shown in Fig. 1.

**REFERENCES**


**Comments on “Discrete Convolution by Means of Forward and Backward Modeling”**

Ayşin Ertüzün and Kadri Çıtmacı

**Abstract**—A recent article by Porsani and Ulyrch presented a method for calculating discrete convolution by means of a tap-delay line structure in conjunction with the Levinson-Durbin recursions as a new alternative to the conventional discrete convolution method. In this correspondence paper, we show that care must be taken in implementing this algorithm.

**I. INTRODUCTION**

In convolving two discrete-time signals, it is implicitly assumed that the signals are zero outside the observation interval. However, this assumption may create some undesirable effects, as in the periodogram method of spectrum estimation. In the Porsani-Ulyrch algorithm [1], the zero assumption about data outside the observation interval is no longer necessary since in their algorithm, the convolution is viewed as least-squares modeling of one function in terms of another function.

Porsani and Ulyrch, in their paper [1], have demonstrated the relationship between the cross-correlation vector associated with normal equations in forward–backward prediction and the convolution of two signals and then, by the recursive solution of the normal equations, have related the convolution terms to the filter coefficients.

This correspondence paper gives some comments on the implementation of the aforementioned algorithm. The organization of this correspondence paper is as follows: in Section II, the algorithm by Porsani and Ulyrch [1] is described; in Section III, the comments and examples to illustrate these comments are put forward; and finally in Section IV, conclusions are given.

**II. DISCRETE CONVOLUTION BY FORWARD AND BACKWARD MODELING**

In a least-squares forward and backward prediction problem, an $m$-point signal $x(n)$ is transformed into a desired signal $d(n)$ by forward and backward shaping filters of order $j+1$, respectively.

The normal equations for forward and backward modeling of order $j+1$ are given as follows, respectively:

\[
\mathbf{R}_{j+2} \begin{bmatrix}
1 \\
j+1
\end{bmatrix} = \begin{bmatrix}
\mathbf{r}_0 \\
\mathbf{r}_{x,d,j+1} \\
\mathbf{r}_{x,z,j+1}
\end{bmatrix} \begin{bmatrix}
1 \\
j+1
\end{bmatrix} = \begin{bmatrix}
\mathbf{E}_{h,j+1} \\
\mathbf{E}_{f,j+1}
\end{bmatrix}
\]

(2.1)

\[
\mathbf{R}_{j+2} \begin{bmatrix}
1 \\
j+1
\end{bmatrix} = \begin{bmatrix}
\mathbf{r}_0 \\
\mathbf{r}_{x,d,j+1} \\
\mathbf{r}_{x,z,j+1}
\end{bmatrix} \begin{bmatrix}
1 \\
j+1
\end{bmatrix} = \begin{bmatrix}
\mathbf{0}_{j+1} \\
\mathbf{E}_{f,j+1}
\end{bmatrix}
\]

(2.2)

where $\mathbf{h}_{j+1}$ and $\mathbf{f}_{j+1}$ are the forward and the backward modeling operators of order $j+1$, respectively. $\mathbf{E}_{h,j+1}$ and $\mathbf{E}_{f,j+1}$ are the modeling energies associated with forward and backward modeling operators of order $j+1$, respectively. $\mathbf{r}_{x,d,j+1}$ and $\mathbf{r}_{x,z,j+1}$ are the cross-correlation vectors of the desired response and the input signal for $j$ lags to the left and to the right of $x(n)$, respectively. $\mathbf{R}_{x,z,j+1}$ is the Toeplitz autocorrelation matrix.

Manuscript received February 3, 1993; revised July 14, 1994. The associate editor coordinating the review of this paper and approving it for publication was Prof. Stanley J. Reeves.

The authors are with the Department of Electrical and Electronic Engineering, Boğaziçi University, Istanbul, Turkey.

IEEE Log Number 9408240.
In the Porsani-Ulrych algorithm [1], the desired response \( d(n) \) presents the time reverse of particular filter weights, namely, 
\( (n) = y(k - n) \). With this correspondence, \( r_{d, d+1} \) represents \( j \) convolution terms between the filter and the signal situated to the left of the initial term \( r_{d, 0} \). Similarly, by backward modeling, we will obtain \( j \) convolution terms to the right of the term \( r_{d, 0} \). Using the notation in [1], the convolution terms associated with \( r_{d, d+1} \) and \( r_{d, d+1} \) are designated by \( \hat{c}_{0, j+1} \) and \( \hat{c}_{0, j+1} \) with elements \( \hat{c}_{k-1}, \hat{c}_{k-2}, \ldots, \hat{c}_{k-j}, \) and \( \hat{c}_{k+1}, \ldots, \hat{c}_{k+j} \), respectively.

When the zero data assumption is left out, the normal equations are modified as follows:

\[
C_{h, j+2} \begin{bmatrix} 1 \\ h_j+1 \end{bmatrix} = \begin{bmatrix} c_{0, j+1} - \hat{c}_{0, j+1} \\ \hat{c}_{0, j+1} \\ c_{0, j+1} \end{bmatrix} \begin{bmatrix} 1 \\ h_j+1 \end{bmatrix} = \begin{bmatrix} E_{h, j+1} \\ 0_j \end{bmatrix} \tag{2.3}
\]

\[
C_{f, j+2} \begin{bmatrix} f_j+1 \\ 1 \end{bmatrix} = \begin{bmatrix} c_{0, j+1} - \hat{c}_{0, j+1} \\ \hat{c}_{0, j+1} \\ c_{0, j+1} \end{bmatrix} \begin{bmatrix} f_j+1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0_j \\ E_{f, j+1} \end{bmatrix} \tag{2.4}
\]

where the correlation matrices \( C_{h, j+2} \) and \( C_{f, j+2} \) are no longer Toeplitz. \( \hat{c}_{0, j+1} \) and \( \hat{c}_{0, j+1} \) are the cross-correlation vectors with elements \( \hat{c}_{k-1}, \hat{c}_{k-2}, \ldots, \hat{c}_{k-j} \) and \( \hat{c}_{k+1}, \ldots, \hat{c}_{k+j} \), respectively. Now, the elements of the cross-correlation vector do not give the exact convolution terms but rather approximations to them. Thus, a correction term has to be introduced in order to obtain each desired convolution term. These correction terms are zero if zero data assumption is made.

The following equations give the \( j \) convolution terms to the left and to the right of \( c_k \) in terms of forward and backward predictor operators \( a_j \) and \( b_i \), respectively:

\[
\hat{c}_{k-j} = (c_{0, j} - \hat{c}_{0, j})^T b_j + c_{k-j} \tag{2.5}
\]

\[
\hat{c}_{k+j} = (c_{0, j} - \hat{c}_{0, j})^T a_j + c_{k+j} \tag{2.6}
\]

The first terms of (2.5) and (2.6) are the correction terms that must be added to the approximate correlation values \( c_{k-j} \) and \( c_{k+j} \), respectively. To obtain the correction terms, the difference terms are obtained between the forward and backward predictor operators. These difference terms show how much the \( j \) convolution terms previously calculated deviate from the \( j \) terms of the cross-correlation vector associated with the normal equations for forward and backward modeling of order \( j+1 \).

The first \( j \) terms of \( c_{0, j+1} \) and the last \( j \) terms of \( c_{0, j+1} \) can be obtained from the \( j \) terms determined at the previous stage by the following update equations:

\[
c_{0, j+1} = c_{0, j} - y_{k-j+1} (x_{j-1}, x_{j-2}, \ldots, x_0)^T \tag{2.7}
\]

\[
c_{0, j+1} = c_{0, j} - y_{m+k+j} (x_{m-1}, x_{m-2}, \ldots, x_{m-j})^T \tag{2.8}
\]

Using (2.5) and (2.6) together with (2.7) and (2.8), \( 2j \) convolution terms adjacent to \( c_k \) can be calculated recursively without invoking zero data assumption outside the observation interval.

### III. Comments on the Algorithm

We performed the computer simulation of the algorithm for an AR process of length 101-points, and we think that there are many important points to be explored in detail.

Care should be taken in choosing the following parameters during the execution of the program.

1. The **order** \( J \) of the prediction filter.
2. The **preassigned length** \( m \) of the incoming series \( x \) (\( m \) is the predetermined length for \( x \), which is smaller than or equal to the actual length in the algorithm).

3. The **central value** \( k \) for the convolution terms.
4. The understated but vital **coupled relation** between the prediction filter order \( J \), the central value \( k \), and the preassigned length of the incoming series \( m \).

Now, we will examine each of these items in some detail.

### A. Order of the Prediction Error Filter

The choice of the filter length \( J \) is not independent of the lengths of the series to be convolved. Due to the recursive calculation of the convolution terms starting with order \( j = 1 \), the error propagates through the steps and builds up as the order of the prediction filter increases. A better way of implementing this algorithm is to increase the length of the convolving signals if a long filter is used. As already known very well, increasing the length of the incoming signal increases the accuracy of the time averaging operation that replaces the expectations. To justify this point, two series of lengths 101 points and 21 points are convolved. The preassigned length \( m \) and the central value \( k \) are fixed to 90 and 88, respectively. When the filters of different lengths are used to find the convolution values, it is observed that the error in calculating these values has increased as longer filter lengths are used. Table I illustrates this observation.

### B. Preassigned Length \( m \) of \( x(n) \)

The autocorrelation values for different lags of the input sequence are given as follows:

\[
r(l) = \frac{1}{N-l} \sum_{n=0}^{N-l} x(n)x^{\ast}(n-l) \quad l = 0, \ldots, J \tag{3.1}
\]

where \( N \) is the actual length of the input signal and \( J \) is the final order of the prediction filter. It is clear that the autocorrelation coefficients strictly depend on the duration of the signal. Naturally, for sufficiently large \( N \), autocorrelation values will be unbiased. The Porsani-Ulrych algorithm initially assumes that the incoming signal is of duration \( m \) points, calculates the side convolution terms (convolution terms to the right and to the left of the central value \( c_k \)), and then, leaving this assumption aside, adds correction terms to the convolution coefficients previously calculated. If the preassigned length \( m \) is assumed to be small, the autocorrelation coefficients will be averaged by a small number since \( N \), in (3.1), will be replaced by the preassigned length of the signal, namely by \( m \). This will create biased values for the autocorrelation values, which will deviate the convolution terms obtained by the algorithm from those of the true values. Table II shows how error in calculating the convolution terms decreases as \( m \) increases for fixed signal sizes and for fixed \( k \) and \( J \) values.

### Justification of Comments 1 and 2

Equation (3.1) gives the autocorrelation values for different lags of the input sequence \( x(n) \). The autocorrelation values form the matrix \( C_{x, x+1} \) used in (2.3) and (2.4). Close examination of (3.1) shows that the averaging operation is done over \( N - l \) samples to calculate the autocorrelation values for lags \( l \) where the upper limit of \( l \) goes up to the filter order \( J \). For lags equal to filter length \( J \), or for lags in the vicinity of this number, the averaging will be done over a small number of terms, leading to biased autocorrelation values. Obviously, not all convolution terms will experience the same bias. However, one obvious point is that the autocorrelation values will encounter higher biases if the order of the filter is high. Let \( \Delta c(l) \) designate the bias associated with the autocorrelation value of lag \( l \) where \( l = 0, \ldots, J \). Each term of \( C_{x, x+1} \) matrix is thus modified by \( \Delta c(l) \). Thus through matrix \( C_{x, x+1} \), the biased autocorrelation terms will affect the calculation of vectors \( a_j \) and \( b_i \), which in turn will affect the correction terms that must be added to the approximate correlation values \( c_{k-j} \) and \( c_{k+j} \).
TABLE I

<table>
<thead>
<tr>
<th>c</th>
<th>83</th>
<th>84</th>
<th>85</th>
<th>86</th>
<th>87</th>
<th>88</th>
<th>89</th>
<th>90</th>
<th>91</th>
<th>92</th>
<th>93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>True</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Err</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

J=1

<table>
<thead>
<tr>
<th>c</th>
<th>83</th>
<th>84</th>
<th>85</th>
<th>86</th>
<th>87</th>
<th>88</th>
<th>89</th>
<th>90</th>
<th>91</th>
<th>92</th>
<th>93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>-9.0</td>
<td>-9.0</td>
<td>-11.0</td>
<td>9.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>True</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>-9.0</td>
<td>-9.0</td>
<td>-11.0</td>
<td>10.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Err</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

J=2

<table>
<thead>
<tr>
<th>c</th>
<th>83</th>
<th>84</th>
<th>85</th>
<th>86</th>
<th>87</th>
<th>88</th>
<th>89</th>
<th>90</th>
<th>91</th>
<th>92</th>
<th>93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg</td>
<td>-</td>
<td>21.0</td>
<td>0.0</td>
<td>-9.0</td>
<td>-9.0</td>
<td>-17.0</td>
<td>10.2</td>
<td>8.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>True</td>
<td>-</td>
<td>21.0</td>
<td>0.0</td>
<td>-9.0</td>
<td>-9.0</td>
<td>-11.0</td>
<td>10.0</td>
<td>7.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Err</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.0</td>
<td>-0.2</td>
<td>-1.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

J=3

<table>
<thead>
<tr>
<th>c</th>
<th>83</th>
<th>84</th>
<th>85</th>
<th>86</th>
<th>87</th>
<th>88</th>
<th>89</th>
<th>90</th>
<th>91</th>
<th>92</th>
<th>93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg</td>
<td>11.0</td>
<td>21.0</td>
<td>0.0</td>
<td>-9.0</td>
<td>-9.0</td>
<td>-23.0</td>
<td>3.3</td>
<td>15.4</td>
<td>11.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>True</td>
<td>11.0</td>
<td>21.0</td>
<td>0.0</td>
<td>-9.0</td>
<td>-9.0</td>
<td>-11.0</td>
<td>10.0</td>
<td>7.0</td>
<td>-2.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Err</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>12.0</td>
<td>6.7</td>
<td>-8.4</td>
<td>-13.7</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

J=4

<table>
<thead>
<tr>
<th>c</th>
<th>83</th>
<th>84</th>
<th>85</th>
<th>86</th>
<th>87</th>
<th>88</th>
<th>89</th>
<th>90</th>
<th>91</th>
<th>92</th>
<th>93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg</td>
<td>28.0</td>
<td>11.0</td>
<td>21.0</td>
<td>0.0</td>
<td>-9.0</td>
<td>-9.0</td>
<td>-23.0</td>
<td>2.0</td>
<td>10.2</td>
<td>4.5</td>
<td>-0.8</td>
</tr>
<tr>
<td>True</td>
<td>28.0</td>
<td>11.0</td>
<td>21.0</td>
<td>0.0</td>
<td>-9.0</td>
<td>-9.0</td>
<td>-11.0</td>
<td>10.0</td>
<td>7.0</td>
<td>-2.0</td>
<td>-19.0</td>
</tr>
<tr>
<td>Err</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>12.0</td>
<td>8.0</td>
<td>-3.2</td>
<td>-6.5</td>
<td>-18.2</td>
<td></td>
</tr>
</tbody>
</table>

J=5

\[ c_{k+j} \text{ in (2.5) and (2.6), respectively. In (3.1), } N \text{ is the actual length of the sequence } x(n), \text{ and it is fixed. The preassigned length } m, \text{ which is also fixed, is used instead of } N, \text{ by the same argument } m - l \text{ should be a large number in order to have unbiased estimates. Thus, in order to have } N - l \text{ or } m - l \text{ be large, when } N \text{ or } m \text{ is fixed, filter order or namely, } J, \text{ should be small. The larger the filter order, the higher will be the error in calculating the convolution values by (3.1) and by (2.3) (2.6) in that order. After the erroneous calculation of the convolution terms, this error will build up through the recursive nature of the algorithm [1]. It is shown that the performance of the algorithm is improved by unbiased averaging of the autocorrelation values, and this can be achieved either by decreasing the order of the prediction filter (comment 1) or by increasing the preassigned length } m \text{ (comment 2).} \]

C. Central Value \( k \)

The algorithm proposed by Porsani and Ulrych [1] sets a central value \( k \) and determines \( j \) side terms to the left and to the right of this central value, respectively. The central value \( k \) can be interpreted as the foldover position or the position for the reverse of the filter operator. It is logical to choose the \( k \) value somewhere at the vicinity of the midpoint of the preassigned length \( m \). It is very well known that convolving two series of lengths \( m \) and \( j \) creates an output series of length \( m + j - 1 \). The first and the last \( j \) elements of the output series are the postcursors and precursors, respectively. In the Porsani-Ulrych algorithm [1], since no assumption is made about the data outside the observation interval, the postcursors and precursors are not calculated. This can be easily verified by examining the limits of the summations in the expressions giving the side convolution terms [1], namely \( c_{\phi, j+1} \) and \( c_{\phi, j+1} \) which are given here for convenience

\[ c_{\phi, j+1} = \sum_{i=j}^{m-1} y_{k-i} x_{j+1} \]  \quad (3.2)

\[ c_{\phi, j+1} = \sum_{i=j}^{m-1} y_{k-i} x_{j+1} \]  \quad (3.3)

Thus, if \( k \) is chosen near the endpoints, we try to calculate the convolution terms either in the postcursor or precursors part, depending whether \( k \) is at the beginning or at the end, which will create erroneous terms since the zero data assumption is not valid any more. Choosing \( k \) somewhere in the middle of the input series preserves the symmetry in calculating the convolution values. This also explains the errors created in calculating the convolution values to the right of \( c_k \) in Table I and as the erroneous terms on both sides of \( c_k \) for \( k = 97 \) and \( k = 107 \) in Table III. Table III illustrates the results of the convolution of 101-point and 21-point signals for different \( k \) values. In this example, \( m \) is fixed to 90 and \( J \) is chosen to be 5. Note that 97 and 107 are greater than the preassigned length \( m = 90 \) in this example. For the convolution of the same two signals, when \( k \) is chosen as 45, which is the midpoint of the preassigned length, no errors occur on either side of \( c_k \). Thus, it is more reliable to keep the \( k \) value away from the tails of the final series. As a rule of thumb, the value of \( k \) should be chosen in the vicinity of the midpoint of the preassigned length \( m \). This important issue was ignored in the paper of Porsani and Ulrych [1].

D. Coupled Relation Between \( J \), \( m \), and \( k \)

In almost all of the above cases, we somehow assume that there is an implicit relationship between the order of the filter, the preassigned
### TABLE II

Effect of Increasing the Preassigned Length of the Incoming Series on the Convolution of 101-Point and 21-Point Signals; \( k = 80 \) and \( J = 2 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>78</th>
<th>79</th>
<th>80</th>
<th>81</th>
<th>82</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg.</td>
<td>-23.8</td>
<td>11.4</td>
<td>19.0</td>
<td>-9.8</td>
<td>-11.1</td>
</tr>
<tr>
<td>True</td>
<td>-21.0</td>
<td>-5.0</td>
<td>14.0</td>
<td>-15.0</td>
<td>-32.0</td>
</tr>
<tr>
<td>Error</td>
<td>2.8</td>
<td>-16.4</td>
<td>-5.0</td>
<td>-5.2</td>
<td>-20.9</td>
</tr>
</tbody>
</table>

\( m=70 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>78</th>
<th>79</th>
<th>80</th>
<th>81</th>
<th>82</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg.</td>
<td>-10.9</td>
<td>-2.8</td>
<td>15.0</td>
<td>-6.9</td>
<td>-12.5</td>
</tr>
<tr>
<td>True</td>
<td>-21.0</td>
<td>-5.0</td>
<td>14.0</td>
<td>-15.0</td>
<td>-32.0</td>
</tr>
<tr>
<td>Error</td>
<td>6.0</td>
<td>-3.2</td>
<td>0.0</td>
<td>-8.1</td>
<td>6.7</td>
</tr>
</tbody>
</table>

\( m=75 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>78</th>
<th>79</th>
<th>80</th>
<th>81</th>
<th>82</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg.</td>
<td>-27.0</td>
<td>-1.8</td>
<td>14.0</td>
<td>-6.9</td>
<td>-38.7</td>
</tr>
<tr>
<td>True</td>
<td>-21.0</td>
<td>-5.0</td>
<td>14.0</td>
<td>-15.0</td>
<td>-32.0</td>
</tr>
<tr>
<td>Error</td>
<td>6.0</td>
<td>-3.2</td>
<td>0.0</td>
<td>-8.1</td>
<td>6.7</td>
</tr>
</tbody>
</table>

\( m=80 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>78</th>
<th>79</th>
<th>80</th>
<th>81</th>
<th>82</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg.</td>
<td>-21.0</td>
<td>-5.0</td>
<td>14.0</td>
<td>-15.0</td>
<td>-32.0</td>
</tr>
<tr>
<td>True</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Error</td>
<td>6.0</td>
<td>-3.2</td>
<td>0.0</td>
<td>-8.1</td>
<td>6.7</td>
</tr>
</tbody>
</table>

\( m=85 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>78</th>
<th>79</th>
<th>80</th>
<th>81</th>
<th>82</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg.</td>
<td>-21.0</td>
<td>-5.0</td>
<td>14.0</td>
<td>-15.0</td>
<td>-32.0</td>
</tr>
<tr>
<td>True</td>
<td>-21.0</td>
<td>-5.0</td>
<td>14.0</td>
<td>-15.0</td>
<td>-32.0</td>
</tr>
<tr>
<td>Error</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

\( m=90 \)

### TABLE III

Effect of Different \( k \) Values on the Convolution of 101-Point and 21-Point Signals; \( m = 90 \) and \( J = 5 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>40</th>
<th>41</th>
<th>42</th>
<th>43</th>
<th>44</th>
<th>45</th>
<th>46</th>
<th>47</th>
<th>48</th>
<th>49</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg.</td>
<td>104.0</td>
<td>60.0</td>
<td>35.0</td>
<td>52.0</td>
<td>23.0</td>
<td>21.0</td>
<td>51.0</td>
<td>41.0</td>
<td>41.0</td>
<td>17.0</td>
<td>20.0</td>
</tr>
<tr>
<td>True</td>
<td>104.0</td>
<td>60.0</td>
<td>35.0</td>
<td>52.0</td>
<td>23.0</td>
<td>21.0</td>
<td>51.0</td>
<td>41.0</td>
<td>41.0</td>
<td>17.0</td>
<td>20.0</td>
</tr>
<tr>
<td>Error</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

\( k=45 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>92</th>
<th>93</th>
<th>94</th>
<th>95</th>
<th>96</th>
<th>97</th>
<th>98</th>
<th>99</th>
<th>100</th>
<th>101</th>
<th>102</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg.</td>
<td>-17.8</td>
<td>-20.5</td>
<td>-18.6</td>
<td>-24.6</td>
<td>-4.1</td>
<td>-16.0</td>
<td>-11.6</td>
<td>-2.3</td>
<td>-15.1</td>
<td>-26.1</td>
<td>-21.1</td>
</tr>
<tr>
<td>True</td>
<td>-2.0</td>
<td>-19.0</td>
<td>-22.0</td>
<td>-45.0</td>
<td>-19.0</td>
<td>-21.0</td>
<td>-17.0</td>
<td>-19.0</td>
<td>-5.0</td>
<td>17.0</td>
<td>12.0</td>
</tr>
<tr>
<td>Error</td>
<td>15.8</td>
<td>1.5</td>
<td>-3.4</td>
<td>-20.4</td>
<td>-14.9</td>
<td>-5.0</td>
<td>-5.4</td>
<td>-16.7</td>
<td>10.1</td>
<td>43.1</td>
<td>33.3</td>
</tr>
</tbody>
</table>

\( k=97 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>102</th>
<th>103</th>
<th>104</th>
<th>105</th>
<th>106</th>
<th>107</th>
<th>108</th>
<th>109</th>
<th>110</th>
<th>111</th>
<th>112</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg.</td>
<td>-0.3</td>
<td>2.8</td>
<td>6.0</td>
<td>21.2</td>
<td>24.1</td>
<td>-35.0</td>
<td>-15.2</td>
<td>-11.0</td>
<td>-7.8</td>
<td>-8.3</td>
<td>-7.8</td>
</tr>
<tr>
<td>True</td>
<td>12.0</td>
<td>4.0</td>
<td>4.0</td>
<td>15.0</td>
<td>20.0</td>
<td>-24.0</td>
<td>-25.0</td>
<td>-18.0</td>
<td>-6.0</td>
<td>-4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Error</td>
<td>12.3</td>
<td>1.2</td>
<td>-2.0</td>
<td>-6.2</td>
<td>-4.1</td>
<td>11.0</td>
<td>-9.8</td>
<td>-7.0</td>
<td>1.8</td>
<td>4.3</td>
<td>7.8</td>
</tr>
</tbody>
</table>

\( k=107 \)

length, and the central value around which the side convolution terms will be placed. To get satisfactory results, the following conditions should be satisfied:

- If \( k \) is near the endpoints, increase \( m \) and decrease \( J \), thus preventing \( k \) from falling in the tails.
- If \( k \) is free and \( J \) is large (more than 4 or 5) choose \( m \) close to the actual length of the input signal \( x(n) \) to reduce bias in averaging.

- If \( k \) is fixed, as \( J \) increases, increase \( m \) to reduce bias in averaging.
- If \( J \) is fixed, as \( k \) increases, increase \( m \) in order to prevent the foldover position from falling in the tails.
- If \( m \) is fixed, as \( k \) increases, decrease \( J \).

These conditions can easily be verified by the arguments given above.
Another important result is about the choice of the preassigned length \( m \). As a rule of thumb, choose the value of \( m \) at least \( J + 1 \) more than the given \( k \) value. This will give satisfactory results in most cases.

IV. CONCLUSION

It is clear that this method of discrete convolution can be employed more effectively and efficiently if the values of \( m, k, \) and \( J \) are chosen with utmost care. The best position to place the center point \( k \) is the middle point of the preassigned length of the incoming series. This choice of \( k \) will much improve the performance of the algorithm. This very important point is not emphasized in the paper by Porsani and Ulrych [1].

REFERENCES


Selection of Observations in Signal Reconstruction

Stanley J. Reeves and Larry P. Heck

Abstract—In some signal reconstruction problems, the observation equations can be used as a priori information for selecting the best combination of observations before acquiring them. In this correspondence, we define a selection criterion and propose efficient methods for optimizing the criterion with respect to the combination of observations. Our examples illustrate the value of optimized sampling using the proposed methods.

I. INTRODUCTION

Signal reconstruction uses measurements in one domain to estimate parameters or distributions in another domain. A common approach to the signal reconstruction problem is to model the observed signal \( y \) as a linear transformation of \( x \) observed in the presence of additive noise; that is

\[
y = Ax + u
\]

where \( u \) is additive noise and \( A \in \mathbb{C}^{m \times n} \) (\( m \geq n \)) is full rank. For this problem, the goal is to reconstruct a good estimate of \( x \) given the observed signal \( y \).

In many applications, the relationship between the observation \( y \) and the original signal \( x \)—the \( i \)th row of \( A \)—is known a priori. However, the number of observations of the elements of \( y \) that can be made is often limited since collecting the observations may be expensive, time-consuming, or even dangerous for some applications. This limitation exists in computed tomography, magnetic resonance imaging (MRI), and magnetic resonance spectroscopic imaging [1].

Manuscript received December 23, 1993; revised August 1994. The associate editor coordinating the review of this paper and approving it for publication was Dr. Barry Sullivan. S. J. Reeves is with the Department of Electrical Engineering, Auburn University, Auburn, AL 36849 USA. L. P. Heck is with the Acoustics and Radar Technology Laboratory, SRI International, Menlo Park, CA 94025 USA. IEEE Log Number 9408245.

[2]. We would like to select the limited set of observations that will yield the best possible reconstruction of \( x \), using the known mapping from the original signal to the observations being considered to make the selection. This problem is equivalent to choosing rows for \( A \) that correspond to the best set of observations \( y \). Once the best combination of observations is determined, the available resources can be devoted to acquiring only those observations.

Although aspects of this problem have been addressed for specific applications [3]–[7], we have not found the problem as a whole addressed for the general category of signal reconstruction. A related problem has been addressed in the statistics literature on optimal experiment design [8]. However, the goals and constraints of signal reconstruction differ somewhat from that of experiment design. In experiment design, one is typically concerned with obtaining good predictions rather than good regression parameters. Furthermore, optimal design usually assumes either total freedom in the choice of \( A \) or freedom within the confines of a specific regression model, such as a polynomial model.

We pose observation selection as a problem of candidate selection. We consider a candidate set of observations \( \{ y_{ik} : k = 1, \ldots, M \} \), where \( M > m \). The problem then is to find a combination of \( m \) out of \( M \) observations by minimizing an appropriate criterion based on the rows of \( A \) corresponding to the candidate observations. This general formulation of the problem allows us to address many applications of observation selection. To solve this problem, we must first define an appropriate criterion in terms of the candidate matrix \( A \). Then we must identify an efficient means for optimizing the criterion with respect to the choice of observations.

II. OPTIMALITY CRITERION

If we let \( A^+ = (A^H A)^{-1} A^H \), then the least-squares estimate of \( x \) is given by

\[
x_{LS} = A^+ y = A^+ A x + A^+ u = x + A^+ u
\]

Thus, we would like to choose \( A \) so that \( x_{LS} \) is in some sense a reliable estimate of \( x \). That is, we want to reduce the uncertainty in the solution introduced by \( A^+ u \). The sense in which the solution uncertainty is reduced is controlled by the criterion we adopt for the minimization problem. If we adopt a minimum 2-norm criterion on the reconstructed signal, we must minimize

\[
\| x - x_{LS} \|^2 = \| x - (x + A^+ u) \|^2 = \| A^+ u \|^2
\]

for an appropriate choice of \( A \). However, since we do not know \( u \), we must settle for some other measure upon which to base our choice. Therefore, we assume that \( u \) is zero-mean unit variance white noise and take the expected value of (3). (Note that correlated noise with correlation matrix \( R_{uu} \) can be handled by premultiplying \( A \) by the whitening operator \( R_{uu}^{-\frac{1}{2}} \) and then proceeding with \( R_{uu}^{-\frac{1}{2}} A \) in place of \( A \).) The minimization criterion then becomes

\[
E\{\| A^+ u \|^2\} = E\{u^H A (A^H A)^{-1} (A^H A)^{-1} A^H u\} = \text{trace} A (A^H A)^{-1} (A^H A)^{-1} A^H = \text{trace} (A^H A)^{-1} A^H (A^H A)^{-1} = \text{trace} (A^H A)^{-1} = \|A^+\|^2
\]