

**F-X Adaptive Seismic Trace Interpolation**

*Mostafa Naghizadeh* and *Mauricio D. Sacchi*

---

**ABSTRACT**

Exponentially Weighted Recursive Least Squares (EWRLS) is adopted to estimate adaptive prediction filters for f-x seismic interpolation. Adaptive prediction filters are able to model signals where the dominant wavenumbers are varying in space. This concept leads to a f-x interpolation method that does not require windowing strategies for optimal results. In other words, adaptive prediction filters can be used to interpolate waveforms with spatially variant dips. The performance of the interpolation method depends on two parameters: the filter length and a forgetting factor. Particular attention is paid to the selection of the forgetting factor, a parameter that controls the adaptability of the algorithm to changes in local dip. Finally, synthetic and real data examples are used to illustrate the performance of the proposed adaptive f-x interpolation method.

**INTRODUCTION**

Spitz (1991) introduced a seismic trace interpolation method that utilizes prediction filters in the frequency-space (f-x) domain. Spitz’s algorithm considers data as a superposition of events with linear moveout in the time-space (t-x) domain. A superposition of linear events transform into a superposition of complex sinusoids in the f-x domain. Complex sinusoids can be reconstructed via prediction filters (autoregressive operators); this property is used to establish a signal model not only for f-x interpolation (Spitz, 1991) but also for f-x random noise attenuation methods (Canales, 1984; Soubaras, 1994; Sacchi and Kuehl, 2000).

Spitz (1991) showed that the prediction filter estimated from the temporal frequency $f$ can be used to interpolate data at frequency $2f$. Prediction filters estimated from the low-frequency (alias-free) portion of the data are used to interpolate the high-frequency aliased data components. The latter makes f-x interpolation with prediction filters a strong contender for interpolating aliased data (Abma and Kabir, 2005).

Several modifications to Spitz’s prediction filtering interpolation have been proposed. For instance, Porsani (1999) proposed a half-step prediction filter scheme that makes the interpolation process more efficient. Gulunay (2003) introduced an algorithm with similarities to f-x prediction filtering with a very elegant representation in the frequency-wavenumber f-k domain. Recently, Naghizadeh and Sacchi (2007) proposed a modification of f-x interpolation that allows it to reconstruct data with gaps.

Seismic interpolation algorithms depend on a signal model. f-x interpolation methods are not an exception to the preceding statement; they assume data composed of a finite number of waveforms with constant dip. This assumption can be validated via windowing. Interpolation methods driven by, for instance, local Radon transforms (Sacchi et al., 2004) and Curvelet frames (Herrmann and Hennenfent, 2008) assume a signal model that consists
of events with constant local dip. In addition, they implicitly define operators that are local without the necessity of windowing. This is an attractive property, in particular, when compared to non-local interpolation methods (operators defined on a large spatial aperture) where optimal results are only achievable when the seismic event matches the kinematic signature of the operator. Examples of the latter are interpolation methods based on the hyperbolic/parabolic Radon transforms (Darche, 1990; Trad et al., 2002) and migration operators (Trad, 2003).

As we have already pointed out, f-x methods require windowing strategies to cope with continuous changes in dominant wavenumbers (or dips in t-x). In this article we propose a method that avoids the necessity of spatial windows; the prediction filters are automatically updated as lateral variations of dip are encountered. This concept can be implemented in a somehow cumbersome process that requires classical f-x interpolation in a rolling window. In this paper we have preferred to use the framework of recursive least squares (Honig and Messerschmidt, 1984; Marple, 1987) to update prediction filters in a recursive fashion. Following Spitz (1991), prediction filters estimated at temporal frequency \( f \) are used to reconstruct data at frequency \( 2f \). We made a fundamental modification to Spitz's method; the interpolation stage of our algorithm uses local filters obtained via adaptive estimation with Exponentially Weighted Recursive Least Squares (EWRLS).

It is important to mention that interpolation with nonstationary t-x prediction filters was proposed by Crawley et al. (1999). Their algorithm evaluates local t-x prediction filters and missing data; the estimation process is stabilized by imposing smoothness to adjacent local prediction filters. Our implementation, on the other hand, is in the f-x domain and the estimation process is implemented via EWRLS.

The paper is organized as follows: First, we provide the theory for adaptive f-x interpolation and pay particular attention to a recursive implementation of adaptive interpolation via EWRLS. The formal proof is provided in the Appendix. We then evaluate the algorithm with synthetic and real data examples and discuss the parameter selection problem associated to our algorithm.

**THEORY**

**Problem definition**

We consider spatial data in the f-x domain. The data at one monochromatic temporal frequency \( f \) are indicated by the length-N discrete signal \( \mathbf{x} = [x_1, x_2, x_3, \ldots, x_N]^T \). We assume local prediction filters of length \( M \). Forward and backward prediction equations are written as follows:

\[
x_{M+n} = p_1(n)x_{M+n-1} + p_2(n)x_{M+n-2} + \ldots + p_M(n)x_n + f_n
\]

\[
x_n^* = p_1(n)x_{n+1}^* + p_2(n)x_{n+2}^* + \ldots + p_M(n)x_{n+M}^* + b_n
\]

where \( p(n) = [p_1(n), p_2(n), \ldots, p_M(n)]^T \) denotes the adaptive prediction filter at spatial sample \( n \). The quantities denoted \( f_n \) and \( b_n \) indicate the innovation terms of the forward and backward processes. Adaptive prediction filtering can be viewed as a non-stationary autoregressive model. In other words, it can be considered as an autoregressive model with time(space)-variant coefficients. It is important to point out that such a model can also
be used to estimate evolutionary spectra for time-frequency (space-wavenumber) analysis (Priestly, 1988).

Adaptive prediction filters are estimated by minimizing the following weighted forward and backward error functions:

\[ J_f(n) = \sum_{i=1}^{n} \lambda^{n-i} |x_{i+M} - \sum_{k=1}^{M} p_k(n)x_{i+M-k}|^2, \]

where 0 < \( \lambda < 1 \) is the forgetting factor. This parameter is used to deemphasize the contribution of data samples far away from the estimation point \( n \).

Expressions 3 and 4 can be independently solved to estimate adaptive prediction filters. In our numerical implementation we have adopted forward modeling (equation 1) to estimate the prediction filters. Backward modeling equations (equation 2) were used to initialize our algorithm. The latter is explained in a forthcoming section. It is important to stress, however, that one could have defined a numerical implementation where backward prediction is utilized to estimate adaptive prediction filters with forward prediction to initialize the algorithm.

Defining the following auxiliary vector \( \mathbf{u}(i) = [x_{i+M-1}, x_{i+M-2}, \ldots, x_i]^T \) and scalar \( d(i) = x_{i+M} \), the solution that minimize the error function in equation 3 is given by

\[ p(n) = (\sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)\mathbf{u}(i)^H)^{-1} \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)d(i) \]

where

\[ \Phi(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)\mathbf{u}(i)^H \]

\[ \psi(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)d(i). \]

Adaptive estimation via EWRLS

One possible solution of the adaptive prediction problem given by equation 3 involves solving 5 for each spatial position \( n \). The latter will require the inversion of the matrix \( \Phi(n) \) at every spatial position \( n \). We will circumvent the inversion of \( \Phi(n) \) by adopting a recursive scheme where \( p(n) \) is obtained from \( p(n-1) \) and the data point \( x(n) \).

The development of the recursive scheme (EWRLS) can be found in the Appendix section as well as in Honig and Messerschmidt (1984). The method is summarized as
follows
\[ R(n - 1) = \Phi^{-1}(n - 1) \]

Update \( p \) and \( R \)
\[
\omega(n) = \frac{\lambda^{-1}R(n - 1)u(n)}{1 + \lambda^{-1}u(n)^H R(n - 1)u(n)}
\]
\[
\alpha(n) = d(n) - u(n)^H p(n - 1)
\]
\[
p(n) = p(n - 1) + \omega(n)\alpha(n)
\]
\[
R(n) = \lambda^{-1}R(n - 1) - \lambda^{-1}\omega(n)u(n)^H R(n - 1).
\]

Initialization

It is evident from equation 8 that in order to initiate the recursive algorithm \( p(1) \) and \( R(1) \) are required. Reliable estimates of these parameters are obtained using backward prediction modeling.

Denoting \( c(i) = x_i^* \) and \( v(i) = [x_{i+1}^*, x_{i+2}^*, \ldots, x_{i+M}^*]^T \), the solution for equation 4 at estimation point 1 is given by:
\[
\Phi(1) = \sum_{i=1}^{n} \lambda^{i-1}v(i)v(i)^H,
\]
\[
\psi(1) = \sum_{i=1}^{n} \lambda^{i-1}v(i)c(i).
\]

The required initial parameters are given by
\[
R(1) = \Phi^{-1}(1)
\]
\[
p(1) = R(1)\psi(1).
\]

Equation 11 is the only inversion needed by the algorithm. After initializing the algorithm with \( R(1) \) and \( p(1) \), the recursive solution (equation 8) is used to estimate adaptive prediction filters for all spatial positions.

Interpolation via adaptive prediction filters

In order to interpolate the data we consider spatial samples at frequency \( f \) with their associated prediction filters estimated from frequency \( f/2 \) (Spitz, 1991). Consider, for instance, a prediction filter of length \( M = 3 \), the equations for adaptive forward and backward prediction associated to the \( i \)-th filter are given by
\[
\begin{pmatrix}
  x_i \\
  x_{2i+1} \\
  x_{i+1} \\
  x_{2(i+1)+1} \\
  x_{i+2} \\
  x_{2(i+2)+1} \\
  x_{i+3}
\end{pmatrix}
\begin{pmatrix}
  x_i \\
  x_{2i+1} \\
  x_{i+1} \\
  x_{2(i+1)+1} \\
  x_{i+2} \\
  x_{2(i+2)+1} \\
  x_{i+3}
\end{pmatrix}
\approx
\begin{pmatrix}
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\]
\[
\begin{pmatrix}
  p_3(i) \\
  p_2(i) \\
  p_1(i) \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\begin{pmatrix}
  p_3(i) \\
  p_2(i) \\
  p_1(i) \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\approx
\begin{pmatrix}
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\]
\[
\begin{pmatrix}
  p_3(i) \\
  p_2(i) \\
  p_1(i) \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\begin{pmatrix}
  p_3(i) \\
  p_2(i) \\
  p_1(i) \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\approx
\begin{pmatrix}
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\]
\[
\begin{pmatrix}
  p_3(i) \\
  p_2(i) \\
  p_1(i) \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\begin{pmatrix}
  p_3(i) \\
  p_2(i) \\
  p_1(i) \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\approx
\begin{pmatrix}
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0
\end{pmatrix}
\]
The rational indexes indicate the desired samples to be estimated by the interpolation scheme. Notice that in the system 13 we have not included the innovation/error terms seen in expressions 1-2. We have, however, indicated the presence of the small innovation term by including $\approx 0$.

Equations similar to 13, for all possible samples $i$, lead to an over-determined system from where one can estimate the unknown data. We let $G^i$ be the above $[2(M+1) \times 2M+1]$ matrix. Matrices for all possible spatial samples $i$ are combined in one augmented system of equations containing all the known and unknown data

\[
\begin{pmatrix}
G^1 & 0_{[2(M+1)\times[2(N-M-1)]} \\
\vdots & \ddots & \ddots \\
0_{[2(M+1)\times[2(N-M-1)]} & \cdots & G^i & 0_{[2(M+1)\times[2(N-M-1)]} \\
0_{[2(M+1)\times[2(N-M-1)]} & \cdots & \cdots & \cdots \\
\end{pmatrix} \approx 0_{[2(M+1)(N-M)]\times[1]}.
\]

The size of the augmented matrix in equation 14 is $[2(M+1)(N-M)] \times [2N-1]$. By dividing data samples into vectors containing known samples

\[
\bar{x}_k = [x_1, x_2, x_3, \ldots, x_N]^T
\]

and unknown samples

\[
\bar{x}_u = [x_{3/2}, x_{5/2}, x_{7/2}, \ldots, x_{(2N-1)/2}]^T,
\]

equation 14 reduces to the following over-determined system

\[
A \bar{x}_u \approx B \bar{x}_k.
\]

The matrices $A$ and $B$ depend on adaptive prediction filters estimated with EWRLS. The last system of equations is solved via the method of least squares

\[
\bar{x}_u = (A^H A)^{-1} A^H B \bar{x}_k.
\]

In our numerical examples we have used the method of conjugate gradients to solve for the unknown samples $\bar{x}_u$. An important advantage of using the method of conjugate gradients is that the number of iterations required for convergence to $\bar{x}_u$ can be truncated to gain computational efficiency. In our examples we have used a maximum number of iterations $\approx N/4$.

**SYNTHETIC AND FIELD DATA EXAMPLES**

**Synthetic data**

We first examine the performance of $f-x$ adaptive interpolation with synthetic examples. Amplitude spectra are portrayed in terms of normalized frequency and normalized wavenumber. Normalized frequency and wavenumber axes are obtained by considering $\Delta t = 1$ and
F-X Adaptive Interpolation

Figure 1: a) Original data. b) Interpolated data using f-x adaptive prediction filtering with forgetting factors $\lambda = 1$ and c) $\lambda = 0.25$. Prediction filters of length $M = 3$ were used for both figures.

$\Delta x = 1$, respectively. This means that in order to obtain frequency axes in Hz and Cycles/m one must divide the normalized ones by $\Delta t$ and $\Delta x$, respectively.

The first synthetic data set is composed of three linear events. Figure 1a shows the synthetic data. The data are interpolated using adaptive prediction filters of length $M = 3$ and forgetting factors $\lambda = 1$ (Figure 1b) and $\lambda = 0.25$ (Figure 1c). Figures 2a, 2b and 2c show the f-k domain representation of Figures 1a, 1b and 1c, respectively. When $\lambda = 1$ the estimation of the prediction filter considers all the data with equal weight regardless of their proximity to the estimation point. In other words, there is no need of adaptability to local dip and the EWRLS algorithm identifies without effort the correct prediction filters required to interpolate the data. In this case the algorithm works for both $\lambda = 1$ and $\lambda = 0.25$. A small improvement in performance is obtained for $\lambda = 1$. The latter corresponds to the stationary filter case where adaptability to dip is not required as expected for this particular example.

To continue with our analysis, we now propose to examine an example where the dip of the waveforms vary with space. A synthetic section composed of hyperbolic events is provided in Figure 3a. The f-k spectrum is shown in Figure 3c. We decimate the original gather to create the decimated data portrayed in Figure 3b. The f-k spectrum of the decimated data is provided in Figure 3d. We interpolate the decimated section of Figure 3b with forgetting factors $\lambda = 1$ (Figure 4a) and $\lambda = 0.15$. In contrast to the previous example, a forgetting factor $\lambda = 1$ results in an extremely low quality interpolation. By choosing $\lambda = 0.15$ the prediction filters are allowed to adapt to the local dip of the data and the algorithm successfully interpolates the data. Figures 4c and 4d show the f-k spectra of Figures 4a and 4b, respectively.

It is insightful to compare the performance of f-x adaptive interpolation to classical f-x interpolation (Spitz, 1991). The decimated synthetic data shown in Figure 3b is interpolated.
using classical f-x interpolation. We first applied the interpolation to the full aperture and then to small overlapping spatial windows. f-x interpolation with Spitz’s method using a single predictor error obtained from the full data aperture is portrayed in Figure 5a. This example is quite unfair to Spitz’s f-x interpolation because the dip of the reflection is rapidly varying with offset. Figure 5b portrays Spitz’s interpolation when the method is applied on small overlapping spatial windows of 7 traces. Figures 5c and 5d show the f-k spectra of Figures 5a and 5b, respectively.

The interpolated data in Figures 4b and 5b as well as the original data (Figure 3a) contain a significant amount of alias. In order to dealias the data one can successively apply adaptive f-x interpolation. Figure 6a shows the result of adaptive f-x interpolation applied to the original data (Figure 3a) using $\lambda = 0.3$. The data in Figure 6a is interpolated again using $\lambda = 0.6$ and the result is depicted in Figure 6b. Notice that we have increased $\lambda$ since the local dip variability has decreased after the first interpolation. Figures 6c and 6d show the de-aliased f-k spectra of Figures 6a and 6b.

Our last synthetic example consists of parabolic events with conflicting dips (Figure 7a). The original data were decimated to obtain the decimated data in Figure 7b. The decimated data are interpolated using adaptive f-x interpolation with $\lambda = 0.3$. The result is shown in Figure 7c. Figures 8a, 8b and 8c provide the f-k spectra of the original, decimated and interpolated data, respectively. This example shows that adaptive f-x interpolation can also resolve conflicting space-variant dips. It is important to stress that an important amount of aliased energy is visible in the original data. The adaptive f-x interpolation has properly resolved the alias as indicated by the f-k panels.
Figure 3: a) Original data, b) decimated data. c) and d) are the f-k spectra of a) and b), respectively.
Figure 4: a) Interpolated data via adaptive f-x interpolation with $\lambda = 1$. b) Interpolated data via adaptive f-x interpolation with $\lambda = 0.15$. The length of the prediction filter for all frequencies and panels is $M = 4$. The f-k spectra of a) and b) are provided in c) and d), respectively.
Figure 5: a) Interpolation with Spitz’s f-x interpolation. Prediction filters were estimated from the full aperture. The latter violates the constant dip assumption. b) Spitz’s f-x applied on small overlapping spatial windows of 7 traces. The method can now successfully interpolate the data. c) and d) provide the f-k spectra of a) and b), respectively.
Figure 6: a) Adaptive f-x interpolation of the original data in Figure 3a with $\lambda = 0.3$. b) Adaptive f-x interpolation of a) with $\lambda = 0.6$. c) and d) are the f-k spectra of a) and b), respectively.
Figure 7: Synthetic example with conflicting dips. a) Original data. b) Decimated data. c) Interpolated data using adaptive $f$-$x$ interpolation with forgetting factor $\lambda = 0.3$ and prediction filter length $M = 4$.

Figure 8: a), b) and c) provide the $f$-$k$ spectra of the data portrayed in Figures 7a, 7b and 7c, respectively.
Marine data example

We first test f-x adaptive interpolation on a marine shot gather from the Gulf of Mexico. The entire gather was interpolated but only a small data window from 1.8s to 3.0s is illustrated. The original data are provided in Figure 9a. The decimated data used to test the interpolation are displayed in Figure 9b. The interpolated data and the interpolation error are provided in Figures 9c and 9d, respectively. A forgetting factor $\lambda = 0.2$ and prediction filters of length $M = 4$ were used for this example. For completeness, we also provide f-k spectra of the original data (Figure 10a), the decimated data (Figure 10b) and the interpolated data (Figure 10c).

Figure 11a shows a near offset section from the Gulf of Mexico. The section was interpolated using adaptive f-x interpolation with $\lambda = 0.2$ and prediction filters of length $M = 4$. The final interpolation is shown in Figure 11b. It is evident that curved diffracted events were properly interpolated. Similar results were obtained using classical f-x interpolation with small overlapping windows of 7 traces.

At this point a few comments about computational efficiency are in order. It is difficult to provide a comparative measure of computational cost for our examples. We have adopted the method of conjugate gradients for the data reconstruction stage for both the adaptive and non-adaptive algorithms. Therefore, the computational cost is heavily controlled by the number of iterations required by the conjugate gradients method to achieve convergence at a given frequency. Our simulations indicate that there is no significant difference in the computational cost of the adaptive algorithm using EWRLS and the non-adaptive Spitz method using overlapping windows.

PARAMETER SELECTION

For optimal results we require an automatic process for the selection of the forgetting factor $\lambda$ and filter length $M$. We have adopted the following heuristic strategy. We first decimate the data, then 3 temporal frequencies are used to compute the average reconstruction error for different values of $M$ and $\lambda$. The minimum reconstruction error provides optimal values $M_{\text{dec}}$ and $\lambda_{\text{dec}}$ for the decimated data. When the algorithm is used to interpolate the original data we use $M = M_{\text{dec}}$ and $\lambda = \lambda_{\text{dec}}^{1/2}$. Using $\lambda = \lambda_{\text{dec}}^{1/2}$ stresses that the forgetting factor should be larger than the one for the decimated data. The above-described strategy was adopted for parameter selection in the synthetic and real data examples shown in this article.

It is also important to address the problem of noisy data. As the SNR decreases one should increase the forgetting factor to allow for more data to participate in the estimation of the adaptive filter. One needs to realize, however, that by increasing $\lambda$ we loose adaptability to changes in local dip. In other words, robustness to the presence of noise and adaptability to local dip are not attainable at the same time. Nevertheless, our real data examples show that f-x adaptive interpolation can handle less than ideal data quite well.
Figure 9: a) Marine common source gather from a data set from the Gulf of Mexico. b) Decimated data. c) $f$-$x$ interpolation with forgetting factor $\lambda = 0.2$ and prediction filter length $M = 4$. d) The interpolation error, original minus interpolated data.
CONCLUSIONS

In this paper we introduced an efficient and easy-to-implement method to interpolate seismic records. We consider the problem of interpolating waveforms with variable dip by re-writing f-x interpolation as an adaptive process. The method eliminates the need of selecting window parameters (window size and amount of overlapping between adjacent windows).

The proposed adaptive f-x interpolation algorithm is robust under strong changes of curvature. In addition, the method performs quite well in the presence of conflicting dips with alias as illustrated by our examples. Adaptive f-x interpolation depends on two parameters: operator length (as in the classical f-x interpolation scheme) and an extra parameter, the forgetting factor, that controls adaptability to changes in local dip. We have also proposed a heuristic method to determine the operator length and forgetting factor.

ACKNOWLEDGMENTS

This research has been supported by the sponsors of the Signal Analysis and Imaging Group, at the University of Alberta and the National Sciences and Engineering Research Council of Canada via a Discovery Grant to MDS. We thank Sam Kaplan for suggesting changes to improve our manuscript. We also acknowledge the Associate Editor Dr C. Kostov and two anonymous reviewers for their valuable input.

APPENDIX

EXPONENTIALY WEIGHED RECURSIVE LEAST SQUARES (EWRLS)

We derive the EWRLS algorithm following (Honig and Messerschmidt, 1984) and modify it for complex algebra. The prediction filter at estimation point n is given by
Figure 11: Real data example portraying the interpolation of a near offset section from the Gulf of Mexico. a) Original section. b) Interpolated section using adaptive $f$-$x$ interpolation with forgetting factor $\lambda = 0.2$ and prediction filter length $M = 4$. 
\[ p(n) = [\Phi(n)]^{-1}\psi(n). \]  
Similarly, the prediction filter at spatial position \( n-1 \) is given by
\[ p(n-1) = [\Phi(n-1)]^{-1}\psi(n-1). \]  
We now require an update rule to go from A-2 to A-1 where we bypass the inversion of the matrices \( \Phi(n-1) \) and \( \Phi(n) \). We first write the matrix \( \Phi(n) \) and the vector \( \psi(n) \) in terms of \( \Phi(n-1) \) and \( \psi(n-1) \)

\[
\Phi(n) = \sum_{i=1}^{n} \lambda^{n-i}u(i)u(i)^H
\]
\[ = \lambda \sum_{i=1}^{n-1} \lambda^{n-1-i}u(i)u(i)^H + u(n)u(n)^H \]
\[ = \lambda\Phi(n-1) + u(n)u(n)^H, \quad (A-3) \]

\[
\psi(n) = \sum_{i=1}^{n} \lambda^{n-i}u(i)d(i)
\]
\[ = \lambda \sum_{i=1}^{n-1} \lambda^{n-1-i}u(i)d(i) + u(n)d(n) \]
\[ = \lambda\psi(n-1) + u(n)d(n). \quad (A-4) \]

We now use the matrix inversion lemma (also called the Sherman-Morrison-Woodbury formula (Hager, 1989)). If \( A \) and \( B \) are \( K \times K \) positive definite matrices, \( D \) is a \( L \times L \) matrix, and \( C \) is a \( K \times L \) such that
\[ A = B^{-1} + CD^{-1}C^H \]  
then the inverse of the matrix \( A \) is given by the following expression
\[ A^{-1} = B + BC(D + C^HBC)^{-1}C^HB. \]  
(A-5)

The following substitutions \( \Phi_{M \times M} = \Phi(n), B_{M \times M} = \lambda^{-1}\Phi^{-1}(n-1), C_{M \times 1} = u(n), \) and \( D_{1 \times 1} = 1 \) are used to apply the inversion lemma to (A-3)
\[ \Phi^{-1}(n) = \lambda^{-1}\Phi^{-1}(n-1) - \frac{\lambda^{-2}\Phi^{-1}(n-1)u(n)u(n)^H\Phi^{-1}(n-1)}{1 + \lambda^{-1}u(n)^H\Phi^{-1}(n-1)u(n)}. \]  
(A-7)

We now define \( R(n) = \Phi^{-1}(n) \), and introduce a new vector \( \omega(n) \)
\[ \omega(n) = \frac{\lambda^{-1}R(n-1)u(n)}{1 + \lambda^{-1}u(n)^H\Phi^{-1}(n-1)u(n)}. \]  
(A-8)

We now multiply both sides of equation A-7 by \( u(n) \) and, after some mathematical manipulations, expression A-8 simplifies to \( \omega(n) = R(n)u(n) \). Now equation A-7 can be written as
\[ R(n) = \lambda^{-1}R(n-1) - \lambda^{-1}\omega(n)u(n)^H\Phi^{-1}(n-1). \]  
(A-9)
We now derive the update equation for $\tilde{p}(n)$. Starting from equation A-1 we have:

$$
\tilde{p}(n) = [\Phi(n)]^{-1}\psi(n) = R(n)\psi(n) = R(n)[\lambda\psi(n-1) + u(n)d(n)] = R(n)[\lambda\Phi(n-1)p(n-1) + u(n)d(n)] = R(n)[\Phi(n) - u(n)u(n)^H \tilde{p}(n-1) + u(n)d(n)] = p(n-1) - R(n)u(n)u(n)^H \tilde{p}(n-1) + R(n)u(n)d(n) = p(n-1) + R(n)u(n)[d(n) - u(n)^H \tilde{p}(n-1)] = p(n-1) + \omega(n)\alpha(n)
$$

(A-10)

where we have defined the scalar variable

$$
\alpha(n) = d(n) - u(n)^H \tilde{p}(n-1).
$$

(A-11)

We have derived expressions A-8, A-9, A-10 and A-11, which are the basic expressions for the recursive algorithm provided in 8.

REFERENCES


