FINAL REPORT

NASA CONTRACT #NAG5-3997

UNIVERSITY OF COLORADO
BOULDER

WAVELET-BASED INTERPOLATION AND REPRESENTATION OF NON-UNIFORMLY SAMPLED SPACECRAFT MISSION DATA

Tamal Bose

Department of Electrical and Computer Engineering

September 5, 2000

1 Objectives

A well-documented problem in the analysis of data collected by spacecraft instruments is the need for an accurate, efficient representation of the data set. The data may suffer from several problems, including additive noise, data dropouts, an irregularly-spaced sampling grid, and time-delayed sampling. These data irregularities render most traditional signal processing techniques unusable, and thus the data must be interpolated onto an even grid before scientific analysis techniques can be applied. In addition, the extremely large volume of data collected by scientific instrumentation presents many challenging problems in the area of compression, visualization, and analysis. Therefore, a representation of the data is needed which provides a structure which is conducive to these applications. Wavelet representations of data have already been shown to possess excellent characteristics for compression, data analysis and imaging.

The main goal of this project is to develop a new adaptive filtering algorithm for image restoration and compression. The algorithm should have low computational complexity and a fast convergence rate. This will make the algorithm suitable for real-time applications. The algorithm should be able to remove additive noise and reconstruct lost data samples from images.

2 Introduction & Accomplishments

The problems of image restoration and compression have received widespread attention in the past two decades. There are several linear and nonlinear techniques that have been proposed. One of the methods used is linear prediction using adaptive filtering. The goal of the adaptive filter is to minimize a cost function in real time with low numerical complexity and simplicity of implementation. Most of the adaptive algorithms have to trade-off between computational complexity and performance. For instance, the least mean squared algorithm (LMS) is widely used in many applications because of its low computationally cost even though it has a somewhat slow convergence rate. In contrast, the recursive least square algorithm (RLS) has fast convergence and low mean squared error, but it has a high computational cost.

Recently, there has been some interest in the implementation of adaptive filters in the frequency domain. Advantages in performance have been reported in several applications. In this project we develop new adaptive algorithms and investigate their implementation in the wavelet domain. These algorithms are then used in conjunction with algorithms for lost sample reconstruction for use in image restoration and compression. In particular, we have accomplished the following in this project.

2.1 Accomplishments

1. Developed a new adaptive filtering algorithm called the Fast Euclidean Direction Search (FEDS) method. This algorithm has a low computational complexity of $O(N)$ and a convergence rate which is comparable to the RLS.
2. Developed 2-D versions of the FEDS algorithm and their variations.
3. Investigated the use of the 1-D and 2-D FEDS algorithms for many applications. The results are quite impressive and outperform those of existing methods.
4. Implemented FEDS algorithms in the wavelet domain.
5. Implemented an algorithm for lost-sample-recovery in conjunction with the FEDS method. We used this for image restoration and compression with encouraging results.

Extra Work We have started working on the design and implementation of 2-D multiplierless filters. Multiplierless filters are those that have coefficients which are only powers-of-two. Therefore, multipliers are simply replaced by shifters. These filters have high computational speed and low cost. Multiplierless 2-D filters have tremendous potential for use in image restoration, compression, and video processing. We have developed some design methods for multiplierless filters. Further design and implementation work on this topic will be conducted in the future.

2.2 Publications

The following publications resulted from this project.

1. M. Radenkovic and Tamal Bose, “Adaptive IIR filtering of nonstationary signals,” Signal Processing, 2000 (in press).

2. M. Radenkovic, Tamal Bose, and T. Mathurasai, “Optimality and almost sure convergence of adaptive IIR filters with output error recursion,” J. Digital Signal Processing, Vol. 9, No. 4, pp. 315-328, 1999.

3. G.F. Xu, Tamal Bose, W. Kober, and J. Thomas, “A fast adaptive algorithm for image restoration,” IEEE Transactions on Circuits and Systems –I, vol. CAS-46, pp. 216-220, Jan. 1999.

4. R. Thamvichai, Tamal Bose, and D.M. Etter, “Design of multiplierless 2-D filters,” Proc. IEEE Intl. Symposium on Intelligent Signal Processing and Communication Systems, Nov. 2000 (accepted).

5. T. Mathurasai, Tamal Bose, M. Radenkovic, and D.M. Etter, “Wavelet domain adaptive filtering based on the Euclidean direction search method,” Proc. IEEE Intl. Symposium on Intelligent Signal Processing and Communication Systems, Nov. 2000 (accepted).

6. R. Thamvichai, Tamal Bose, and D.M. Etter, “2-D Periodically shift variant filters without multipliers,” Proc. World Multiconference on Systemics, Cybernetics, and Informatics, vol. 1, July 2000.

7. T. Mathurasai, Tamal Bose, and D.M. Etter, ”Decision feedback equalization using an euclidean direction based adaptive algorithm,” Proc. of the Asilomar Conference on Signals, Systems, and Computers, vol. 1, pp. 519-523, Oct. 1999.

8. M. Radenkovic and Tamal Bose, ”Global stability of adaptive IIR filters based on the output error method,” Proc. of the Asilomar Conference on Signals, Systems, and Computers, vol. 1, pp. 663-667, Oct. 1999.

9. G.F. Xu, *Fast Algorithms for Adaptive Filtering: Theory and Applications*, Ph.D. Dissertation, University of Colorado at Boulder, 1999.

The rest of the report is organized as follows. In Section 3, the derivation of the FEDS algorithm is presented. In Section 4, the wavelet domain implementation of the FEDS is described. Section 5 reports some results of the application of these algorithms for image restoration. In Section 6, some new results are reported on the design of multiplierless digital filters. Section 7 is the conclusion.

3 Euclidean Direction Search Adaptive Algorithm

The Euclidean Direction Search (EDS) algorithm has been reported by the PI and his research team in [6], [7], [8], [9], [10]. In this section, we present the basic derivation and some features of the algorithm.

The EDS algorithm is essentially a derivative of the Powell and Zangwill DS algorithm. In adaptive applications, we consider the following quadratic optimization problem,

$$J_n(\mathbf{w}) = \min_{\mathbf{w}\in R^N} \left\{ \mathbf{w}^T \mathbf{Q}(n)\,\mathbf{w} - 2\mathbf{w}^T \mathbf{r}(n) + \sigma^2(n) \right\} \qquad (1)$$

where $\mathbf{Q}(n)$ is an $N \times N$ symmetric positive definite matrix, $\mathbf{r}(n)$ is a column vector of order N, $\mathbf{w}$ is the unknown weight vector, and σ^2 is a variance of the desired signal, d. The minimum of $J_n(\mathbf{w})$ can be calculated by setting the gradient to zero,$\nabla \triangleq \frac{\partial J_n(\mathbf{w})}{\partial \mathbf{w}} = 0$. Obviously, the optimal solution is $\mathbf{w}_{opt}= \mathbf{Q}^{-1}\mathbf{r}$.

For the least squares optimization problem, the cost function can be expressed as,

$$\begin{aligned}
F_n(\mathbf{w}) &\triangleq \sum_{i=1}^{n} \lambda^{n-i} e^2(n) && (2)\\
&= \sum_{i=1}^{n} \lambda^{n-i} \left(d(i) - \mathbf{w}^T \mathbf{x}(i)\right)^2 && (3)\\
&= \sigma_d^2(n) - 2\mathbf{w}^T(n)\,\mathbf{r}(n) + \\
&\quad\ \mathbf{w}^T(n)\,\mathbf{Q}(n)\,\mathbf{w}(n) && (4)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{Q}(n) &= \sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}(i)\,\mathbf{x}^T(i) && (5)\\
&= \lambda \mathbf{Q}(n-1) + \mathbf{x}(n)\,\mathbf{x}^T(n) && (6)\\
\mathbf{r}(n) &= \sum_{i=1}^{n} \lambda^{n-i} d(i)\,\mathbf{x}(i) && (7)\\
&= \lambda \mathbf{r}(n-1) + d(n)\,\mathbf{x}(n). && (8)
\end{aligned}$$

$\mathbf{x}(n) = [x(n)\ x(n-1)\ \ x(n-2)\ \ ...\ x(n-N+1)]^T$is an input vector to the adaptive filter of length N at time n, $d(n)$ is a desired signal, and λ is a forgetting factor. Consider the update weight iterative function

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha(n)\,\mathbf{g}(n) \qquad (9)$$

where $\alpha(n)$ and $\mathbf{g}(n)$ are step size and search direction at iteration n, respectively. One can obtain the best step size to minimize the next step cost function by setting $\nabla_\alpha \triangleq \frac{\partial J_n(\mathbf{w}+\alpha\mathbf{g})}{\partial \alpha} = 0$ as follows,

$$\frac{\partial J_n(\mathbf{w}+\alpha\mathbf{g})}{\partial \alpha} = 0 = 2\mathbf{g}^T\mathbf{Q}(n)(\mathbf{w}+\alpha\mathbf{g}) - 2\mathbf{g}^T\mathbf{r}(n).$$

As a result,

$$\alpha = -\frac{\mathbf{g}^T(\mathbf{Q}(n)\,\mathbf{w} - \mathbf{r}(n))}{\mathbf{g}^T\mathbf{Q}(n)\,\mathbf{g}}. \qquad (10)$$

The proposed algorithm utilizes the Euclidean direction, that is, $\mathbf{g}^{(i)} = [0\ ...\ 0\ 1\ 0\ ...\ 0]^T$, where 1 is located in the *i-th* position. This set of directional search allows us to update the weight vector in a simple manner. Considering equation (10), when the $i-th$ position of the direction is selected, we pick up the $i-th$ element of vector $(\mathbf{Q}(n)\,\mathbf{w} - \mathbf{r}(n))$, and simply divide it by $\mathbf{Q}(i,i)$. The EDS algorithm needs totally N rounds to completely update all weight elements in a filter. The algorithm is described explicitly in Table 1. We denote $(\mathbf{w})_i$ and $(\mathbf{r})_i$ as the $i-th$ element of weight vector, $\mathbf{w}$, and, cross correlation vector, $\mathbf{r}$, respectively. $\mathbf{q}^{(i)}$ is $i-th$ row of matrix **Q**.

```
For $n = 1, 2, ...$
For $i = 1, \cdots, N$
(1) $\varepsilon = \left(\mathbf{q}^{(i)}\right)^T \mathbf{w} - (\mathbf{r})_i$;
(2) $a = \left(\mathbf{q}^{(i)}\right)_i$;
(3) if $a > 0 \quad \alpha = -\frac{\varepsilon}{a}; \quad (\mathbf{w})_i = (\mathbf{w})_i + \alpha$;
(4) $\mathbf{q}^{(i)} = \lambda \mathbf{q}^{(i)} + (\mathbf{x}(k+1))_i \mathbf{x}^T(k+1)$
(5) $(\mathbf{r})_i = \lambda (\mathbf{r})_i + (\mathbf{x}(k+1))_i d(k+1)$
end $i$
```

Table 1. EDS Algorithm

The EDS algorithm needs $N(3N+3)$ multiplications for each incoming data sample. In other words, it has $O(N^2)$ computational complexity [7]. This is relatively high compared to other algorithms, although EDS has comparable performance. There is however, an alternative approach which can reduce the computational cost to $O(N)$. The new algorithm is the so called EDS1 or Fast-EDS (FEDS) [6], [7], [8], [9], [10]. In order to decrease the computations to $O(N)$, we need to modify the objective function into a block exponentially weight least square form as follows:

$$\begin{aligned}\min_{\mathbf{w} \in R^N} J_n(\mathbf{w}) &= \sum_{i=0}^{k-1} \lambda^{k-i} \{ \sum_{j=1}^{N} (diN + j - \\ &\quad \mathbf{w}(n)^T \mathbf{x}(iN + j))^2 \} + \\ &\sum_{j=1}^{l} \left(d(kN + j) - \mathbf{w}(n)^T \mathbf{x}(kN + j) \right)^2 \end{aligned} \tag{11}$$

with $n = kN + l$, where k is the number of full blocks within N and l is the number of samples remaining. The new cost function leads to modifications of $\mathbf{Q}(n)$, $\mathbf{r}(n)$, and $\sigma^2(n)$ as

$$\begin{aligned}
\mathbf{Q}(n) &= \mathbf{Q}(kN+l) \\
&= \sum_{i=0}^{k-1} \lambda^{k-i} \left(\sum_{j=1}^{N} \mathbf{x}(iN+j)\,\mathbf{x}(iN+j)^T \right) + \\
&\quad \sum_{j=1}^{l} \mathbf{x}(kN+j)\,\mathbf{x}(kN+j)^T ; \qquad (12) \\
\mathbf{r}(n) &= \mathbf{r}(kN+l) \\
&= \sum_{i=0}^{k-1} \lambda^{k-i} \left(\sum_{j=1}^{N} d(iN+j)\,\mathbf{x}(iN+j) \right) + \\
&\quad \sum_{j=1}^{l} d(kN+j)\,\mathbf{x}(kN+j) ; \qquad (13) \\
\sigma^2(n) &= \sigma^2(kN+l) \\
&= \sum_{i=0}^{k-1} \lambda^{k-i} \left(\sum_{j=1}^{N} d^2(iN+j) \right) + \\
&\quad \sum_{j=1}^{l} d^2(kN+j) . \qquad (14)
\end{aligned}$$

To compute these parameters recursively, equation (12) and (13) can be written as,

$$\begin{aligned}
\mathbf{Q}(kN+l) &= \mathbf{Q}(n) \\
&= \lambda \mathbf{Q}(kN) + \widetilde{\mathbf{Q}}(kN+l) , \qquad (15) \\
\mathbf{r}(kN+l) &= \mathbf{r}(n) \\
&= \lambda \mathbf{r}(kN) + \widetilde{\mathbf{r}}(kN+l) , \qquad (16)
\end{aligned}$$

where

$$\widetilde{\mathbf{Q}}(kN+l) = \sum_{j=1}^{l} \mathbf{x}(kN+j)\,\mathbf{x}(kN+j)^T \qquad (17)$$

$$\widetilde{\mathbf{r}}(kN+l) = \sum_{j=1}^{l} d(kN+j)\,\mathbf{x}(kN+j) . \qquad (18)$$

The FEDS algorithm differs from the EDS algorithm in the following ways. First, it updates the correlation matrix, **Q**, and cross correlation vector, **r**, using the above equations. Secondly, only one Euclidean direction search is conducted per incoming sample of data. The FEDS algorithm is illustrated in Table 2.

```
For $k = 1, 2, ...$
Step (I): For $i = 1, \cdots, N$
    (1) $\mathbf{q}^{(i)} = \lambda \mathbf{q}_i^{(i)}$;
    (2) $(\mathbf{r})_i = \lambda (\mathbf{r})_i$;
    (3) $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{Q}} + \mathbf{x}(kN+i)\mathbf{x}^T(kN+i)$;
    (4) $\widetilde{\mathbf{r}} = \widetilde{\mathbf{r}} + d(kN+i)\mathbf{x}(kN+i)$;
    (5) $\varepsilon = \left(\mathbf{q}^{(i)} + \widetilde{\mathbf{q}}^{(i)}\right)^T \mathbf{w} - (\mathbf{r} + \widetilde{\mathbf{r}})_i$;
    (6) $a = \left(\mathbf{q}^{(i)}\right)_i + \left(\widetilde{\mathbf{q}}^{(i)}\right)_i$;
    (7) If $a > 0$   $\alpha = -\frac{\varepsilon}{a}$;   $(\mathbf{w})_i = (\mathbf{w})_i + \alpha$;
end $i$
Step (II)
    (1) $\mathbf{Q} = \mathbf{Q} + \widetilde{\mathbf{Q}}$;   $\widetilde{\mathbf{Q}} = 0$;
    (2) $\mathbf{r} = \mathbf{r} + \widetilde{\mathbf{r}}$;   $\widetilde{\mathbf{r}} = 0$;
```

Table 2. FEDS Algorithm

It is shown in [7] that the computational count of the FEDS algorithm is $4N + 2$ multiplications, that is $O(N)$.

3.1 Stability of The EDS Algorithm

Consider a time-invariant quadratic optimization problem,

$$J_n(\mathbf{w}) = \min_{\mathbf{w} \in R^N} \{\mathbf{w}^T \mathbf{Q} \mathbf{w} - 2\mathbf{w}^T \mathbf{r} + \sigma^2\} \tag{19}$$

where **Q** is an $N \times N$ real symmetric positive definite matrix, **r** and variable **w** are vectors of order N, and σ is a scaler. The notation $(.)^T$ means the transpose of $(.)$.

By introducing a constant μ to regulate the convergence rate, the Newton's method can be expressed as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\mu}{2}\mathbf{Q}^{-1}\nabla(k),$$

where $\mathbf{w}(k)$ denotes the variable vector at step k, $\nabla(k) \triangleq \nabla \mid_{\mathbf{w}=\mathbf{w}(k)} = 2\mathbf{Q}\mathbf{w}(k) - 2\mathbf{r}$, and $0 < \mu < 2$.

The steepest descent method avoids dealing with matrix $\mathbf{Q}^{-1}$ and updates the variable vector $\mathbf{w}$ on the negative gradient direction as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\mu}{2}\boldsymbol{\nabla}(k),$$

where $0 < \mu < \frac{2}{\lambda_{\max}}$ and $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{Q}$.

The Euclidean direction set (EDS) method in a constant environment is equivalent to Gauss-Seidel method, which can be written as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{1}{2}\mathbf{A}^{-1}\boldsymbol{\nabla}(k) \ , \tag{20}$$

where $\mathbf{A}$ is an $N \times N$ triangular matrix or block triangular matrix of $\mathbf{Q}$ as shown in [17].

The stability result of the EDS algorithm can be obtained by defining an error vector as $\mathbf{C}(k) \triangleq \mathbf{w}(k) - \mathbf{w}_*$, where $\mathbf{w}_* = \mathbf{Q}^{-1}\mathbf{r}$ is the optimal solution of (19). Subtracting $\mathbf{w}_*$ from both sides of (20), and using $\boldsymbol{\nabla}(k) = 2(\mathbf{Q}\mathbf{w}(k) - \mathbf{Q}\mathbf{w}_*)$, the error recursive equation becomes

$$\mathbf{C}(k+1) = (I - \mathbf{A}^{-1}\mathbf{Q})\mathbf{C}(k), \tag{21}$$

where $\mathbf{I}$ is the identity matrix. With an initial value of $\mathbf{C}(0)$, for the error vector, the solution of (21) is

$$\mathbf{C}(k) = (I - \mathbf{A}^{-1}\mathbf{Q})^k\mathbf{C}(0). \tag{22}$$

It is well known that $\mathbf{C}(k) \to 0$ as $k \to \infty$, if and only if $\rho(I - \mathbf{A}^{-1}\mathbf{Q}) < 1$, where $\rho(.)$ denotes the maximum absolute eigenvalue of $(.)$.

Based on the fact that matrix $\mathbf{A}$ is a lower triangular submatrix of $\mathbf{Q}$, the following theorem provides a sufficient condition for the convergence of the EDS algorithm. It is worth noting that this sufficient condition is independent of the largest eigenvalue of matrix $\mathbf{Q}$.

Theorem 3.1 *Let $\mathbf{Q}$ be an $N \times N$ symmetric positive definite matrix, and matrix $\mathbf{A}$ be the lower triangular submatrix or a block triangular submatrix of $\mathbf{Q}$. The euclidean direction search (EDS) algorithm $\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{1}{2}\mathbf{A}^{-1}\boldsymbol{\nabla}(k)$,converges to the optimal solution.*

Proof. Let $\mathbf{B} \triangleq \mathbf{A} + \mathbf{A}^T - \mathbf{Q}$. It is easy to verify that matrix $\mathbf{B}$ is a block diagonal submatrix of $\mathbf{Q}$, with each block size being 1×1 or 2×2. As proven in the next subsection, matrix $\mathbf{B}$ is symmetric and positive definite. Now, assume that λ and $\mathbf{x}$ are the eigenvalue and associated eigenvector of matrix $\mathbf{A}^{-1}\mathbf{Q}$ where

$$(\mathbf{I} - \mathbf{A}^{-1}\mathbf{Q})\mathbf{x} = (1 - \lambda)\mathbf{x}.$$

Multiplying by $\mathbf{x}^*\mathbf{A}$, taking the absolute value of both sides, and rearranging gives

$$|1 - \lambda| = \frac{|\mathbf{x}^*\mathbf{A}\mathbf{x} - \mathbf{x}^*\mathbf{Q}\mathbf{x}|}{|\mathbf{x}^*\mathbf{A}\mathbf{x}|},$$

where $(.)^*$ denotes the conjugate transpose of $(.)$.

Recall that for $\mathbf{x} \in \mathbf{C}^N; \mathbf{x} \neq \mathbf{0}$, $\mathbf{x}^*\mathbf{Q}\mathbf{x} > 0$, $\mathbf{x}^*\mathbf{B}\mathbf{x} > 0$, and $\mathrm{Re}\{\mathbf{x}^*\mathbf{A}\mathbf{x}\} = \mathbf{x}^*(\frac{A+A^T}{2})\mathbf{x} = \mathbf{x}^*(\frac{\mathbf{Q}+\mathbf{B}}{2})\mathbf{x} > 0$,

$$\begin{aligned} \mathrm{Re}\{\mathbf{x}^*\mathbf{A}\mathbf{x} - \mathbf{x}^*\mathbf{Q}\mathbf{x}\} &< \mathrm{Re}\{\mathbf{x}^*\mathbf{A}\mathbf{x}\}; \\ \mathrm{Im}\{\mathbf{x}^*\mathbf{A}\mathbf{x} - \mathbf{x}^*\mathbf{Q}\mathbf{x}\} &= \mathrm{Im}\{\mathbf{x}^*\mathbf{A}\mathbf{x}\}. \end{aligned}$$

That is, $|1 - \lambda| < 1$,which implies that $\rho(\mathbf{I} - \mathbf{A}^{-1}\mathbf{Q}) < 1$. That is, $\mathbf{C}(k) \to 0$ as $k \to \infty$ in (22), and the conclusion follows. $\sharp$

3.2 Convergence Rate of The EDS Method

In the previous section, we have shown that $|1 - \lambda| < 1$, which implies that $\lambda \neq 0$. Therefore, using similarity transformation, we may express (22) as

$$\mathbf{C}(k) = \mathbf{S}(\mathbf{I} - \mathbf{\Lambda})^k\mathbf{S}^{-1}\mathbf{C}(0),$$

where matrix $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of matrix $(\mathbf{A}^{-1}\mathbf{Q})$, and $\mathbf{A}^{-1}\mathbf{Q} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$. For convenience, define a new vector $\mathbf{V}(k) = \mathbf{S}^{-1}\mathbf{C}(k)$, so that

$$\mathbf{V}(k) = (\mathbf{I} - \mathbf{\Lambda})^k\mathbf{V}(0).$$

Clearly, the convergence rate of each element $v_i(k)$ in vector $\mathbf{V}(k)$ is dependent on the associated eigenvalue λ_i of matrix $(\mathbf{A}^{-1}\mathbf{Q})$.

The quantity $r_i = 1 - \lambda_i$ is known as the "geometric ratio". Note that when the absolute value of r_i is less than 1, the rate of convergence increases as r_i decreases.

As is well known, the overall convergence rate r cannot be expressed in a simple closed form. But fortunately, the absolute value of geometric ratio r is lower bounded. So, we indicate that the convergence performance of the EDS method is superior to the steepest descent method by showing that the lower bound $|r|_{bound}$ in the EDS method is lower than that of the steepest descent method.

Throughout the rest of the chapter, $\lambda_i(.)$, $\lambda_{\max}(.)$ and $\lambda_{\min}(.)$ will denote the *ith*, the largest and the smallest eigenvalues of matrix (.), respectively.

The *ith* geometric ratio, for the steepest descent method, is $r_i = 1 - \mu\lambda_i(\mathbf{Q})$, where $0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{Q})}$. The overall convergence rate is lower bounded as

$$|r| \geq \max\{|1 - \mu\lambda_{\max}(\mathbf{Q})|\,, |1 - \mu\lambda_{\min}(\mathbf{Q})|\}. \tag{23}$$

The best step size μ for the convergence occurs at the cross-point: $\mu\lambda_{\max}(\mathbf{Q}) - 1 = 1 - \mu\lambda_{\min}(\mathbf{Q})$,which gives $\mu = \frac{2}{\lambda_{\max}(\mathbf{Q})+\lambda_{\min}(\mathbf{Q})}$. Substituting it into (23), yields $|r| \geq \frac{\lambda_{\max}(\mathbf{Q})-\lambda_{\min}(\mathbf{Q})}{\lambda_{\max}(\mathbf{Q})+\lambda_{\min}(\mathbf{Q})}$. So, in the steepest descent method,

$$\begin{aligned} |r|_{bound} &= \frac{\lambda_{\max}(\mathbf{Q}) - \lambda_{\min}(\mathbf{Q})}{\lambda_{\max}(\mathbf{Q}) + \lambda_{\min}(\mathbf{Q})} \\ &= 1 - \frac{2}{\frac{\lambda_{\max}(\mathbf{Q})}{\lambda_{\min}(\mathbf{Q})} + 1} \end{aligned} \tag{24}$$

where $\frac{\lambda_{\max}(\mathbf{Q})}{\lambda_{\min}(\mathbf{Q})}$ equals the condition number of $\mathbf{Q}$.

Note that $|r|_{bound}$ decreases as the ratio $\frac{\lambda_{\max}(\mathbf{Q})}{\lambda_{\min}(\mathbf{Q})}$ decreases. Hence, the convergence rate increases as the ratio $\frac{\lambda_{\max}(\mathbf{Q})}{\lambda_{\min}(\mathbf{Q})}$ decreases.

In order to compare the bounds between two methods, let's prove the following lemma first.

Lemma 3.1 *Assume* $\mathbf{Q}$ *is an* $N \times N$ *symmetric positive definite matrix, and matrix* $\mathbf{B}$ *is a block diagonal submatrix of* $\mathbf{Q}$, *such as*

$$\mathbf{B} = \left[\begin{array}{ccccccc} q_{11} & q_{12} & 0 & & \cdots & 0 & \\ q_{21} & q_{22} & & & & & \\ 0 & & q_{33} & q_{34} & \ddots & \vdots & \\ & & q_{43} & q_{44} & & & \\ \vdots & & \ddots & & \ddots & 0 & \\ 0 & & \cdots & & 0 & q_{N-1,N-1} & q_{N,N-1} \\ & & & & & q_{N,N-1} & q_{N,N} \end{array}\right],$$

with each block being square. Then, matrix $\mathbf{B}$ *is symmetric, positive definite, and*

$$\lambda_{\min}(\mathbf{B}) \geq \lambda_{\min}(\mathbf{Q}); \ \lambda_{\max}(\mathbf{B}) \leq \lambda_{\max}(\mathbf{Q}).$$

Proof. For $i = 1, 2, \cdots, N$, the *ith* principle submatrix $\mathbf{Q}_i$ of $\mathbf{Q}$ is the $i \times i$ submatrix consisting of the intersection of the first i rows and columns of $\mathbf{Q}$. Let $\mathbf{B}_j$ denote the *jth* block submatrix in $\mathbf{B}$, i.e. $\mathbf{B} = \{\mathbf{B}_1 \oplus \mathbf{B}_2 \cdots \oplus \mathbf{B}_L\}$. Since any principle submatrix of a symmetric positive definite matrix is symmetric positive definite and $\lambda_k(\mathbf{Q}) \leq \lambda_k(\mathbf{Q}_i) \leq \lambda_{k+N-i}(\mathbf{Q})$ for each integer k such that $1 \leq k \leq i$ [14], it therefore, follows that the first block in $\mathbf{B}$ is symmetric, positive definite, and $\lambda_{\min}(\mathbf{B}_1) \geq \lambda_{\min}(\mathbf{Q})$; $\lambda_{\max}(\mathbf{B}_1) \leq \lambda_{\max}(\mathbf{Q})$.

Recall that there exists the permutation matrices which can bring the other blocks being the first block without effect of the symmetric and positive definite properties [16], and

$$\begin{aligned} \lambda_{\min}(\mathbf{B}) &= \min_{j=1,\cdots L} \{\lambda_{\min}(\mathbf{B}_j)\}\,; \\ \lambda_{\max}(\mathbf{B}) &= \max_{j=1,\cdots L} \{\lambda_{\max}(\mathbf{B}_j)\}\,. \end{aligned}$$

We thus proved that matrix B is symmetric, positive definite, and

$$\lambda_{\min}(\mathbf{B}) \geq \lambda_{\min}(\mathbf{Q});\ \lambda_{\max}(\mathbf{B}) \leq \lambda_{\max}(\mathbf{Q}). \qquad \sharp \tag{25}$$

In the EDS method, the overall convergence rate $|r| \geq \max_{i=1,2,\cdots,N} \{|1 - \lambda_i(\mathbf{A}^{-1}\mathbf{Q})|\}$, and

$$|r|_{bound} = \sup_{\mathbf{A}^{-1}\mathbf{Q}\mathbf{x}=\lambda\mathbf{x};\mathbf{x}\neq\mathbf{0}} \left\{\left|1 - \frac{\mathbf{x}^*\mathbf{Q}\mathbf{x}}{\mathbf{x}^*\mathbf{A}\mathbf{x}}\right|\right\}. \tag{26}$$

Since $\mathbf{x}^*\mathbf{Q}\mathbf{x}$ is real, the supremum occurs when $\mathbf{x}^*\mathbf{A}\mathbf{x}$ is real. Note that, if $\mathbf{x}^*\mathbf{A}\mathbf{x}$ is real, then $\mathbf{x}^*\mathbf{A}\mathbf{x} = \mathbf{x}^*\frac{\mathbf{Q}+\mathbf{B}}{2}\mathbf{x}$ and $\mathbf{x}^*\mathbf{A}\mathbf{x} - \mathbf{x}^*\mathbf{Q}\mathbf{x} = \mathbf{x}^*\frac{\mathbf{B}-\mathbf{Q}}{2}\mathbf{x}$. Therefore,

$$\begin{aligned} |r|_{bound} &\leq \sup_{\mathbf{x}^*\mathbf{A}\mathbf{x}\in\mathbf{R}^N;\mathbf{x}\neq\mathbf{0}} \left\{\frac{|\mathbf{x}^*\mathbf{Q}\mathbf{x} - \mathbf{x}^*\mathbf{B}\mathbf{x}|}{\mathbf{x}^*\mathbf{Q}\mathbf{x} + \mathbf{x}^*\mathbf{B}\mathbf{x}}\right\} \\ &= \left\{\begin{array}{ll} 1 - \frac{2}{\sup_{\mathbf{x}\in\mathbf{R}^N;\mathbf{x}\neq\mathbf{0}} \left\{\frac{\mathbf{x}^*\mathbf{Q}\mathbf{x}}{\mathbf{x}^*\mathbf{B}\mathbf{x}}\right\}+1} & \text{if } \mathbf{x}^*\mathbf{Q}\mathbf{x} \geq \mathbf{x}^*\mathbf{B}\mathbf{x}; \\ 1 - \frac{2}{\sup_{\mathbf{x}\in\mathbf{R}^N;\mathbf{x}\neq\mathbf{0}} \left\{\frac{\mathbf{x}^*\mathbf{B}\mathbf{x}}{\mathbf{x}^*\mathbf{Q}\mathbf{x}}\right\}+1} & \text{if } \mathbf{x}^*\mathbf{Q}\mathbf{x} < \mathbf{x}^*\mathbf{B}\mathbf{x}; \end{array}\right\} \\ &\leq \left\{\begin{array}{ll} 1 - \frac{2}{\frac{\lambda_{\max}(\mathbf{Q})}{\lambda_{\min}(\mathbf{B})}+1} & \text{if } \mathbf{x}^*\mathbf{Q}\mathbf{x} \geq \mathbf{x}^*\mathbf{B}\mathbf{x}; \\ 1 - \frac{2}{\frac{\lambda_{\max}(\mathbf{B})}{\lambda_{\min}(\mathbf{Q})}+1} & \text{if } \mathbf{x}^*\mathbf{Q}\mathbf{x} < \mathbf{x}^*\mathbf{B}\mathbf{x}; \end{array}\right\}. \end{aligned} \tag{27}$$

That is, $|r|_{bound} \leq \max\left\{1 - \frac{2}{\frac{\lambda_{\max}(\mathbf{Q})}{\lambda_{\min}(\mathbf{B})}+1}, 1 - \frac{2}{\frac{\lambda_{\max}(\mathbf{B})}{\lambda_{\min}(\mathbf{Q})}+1}\right\}$.

Recall Lemma 3.1, and note that when the equality in (25) occurs, $|r|_{bound} = 0$. Therefore, $\frac{\lambda_{\max(\mathbf{Q})}}{\lambda_{\min(\mathbf{Q})}} > \frac{\lambda_{\max(\mathbf{Q})}}{\lambda_{\min(\mathbf{B})}}$ and $\frac{\lambda_{\max(\mathbf{Q})}}{\lambda_{\min(\mathbf{Q})}} > \frac{\lambda_{\max(\mathbf{B})}}{\lambda_{\min(\mathbf{Q})}}$ are true in general, which asserts that $|r|_{bound}$ in the EDS is lower than that in steepest descent method.

If $\frac{\lambda_{\max(\mathbf{Q})}}{\lambda_{\min(\mathbf{Q})}}$ is close to 1, with proper step size μ, both EDS and steepest descent method converge fast. However, when $\frac{\lambda_{\max(\mathbf{Q})}}{\lambda_{\min(\mathbf{Q})}}$ is large, the last " $\leq$ " in (27) is a very conservative step. Since matrix $\mathbf{B}$ is a block diagonal submatrix from $\mathbf{Q}$ and matrix $\mathbf{Q}$ is central majorized, $\sup\limits_{\mathbf{x}\in\mathbf{R}^N;\mathbf{x}\neq\mathbf{0}} \frac{|\mathbf{x}^*\mathbf{Q}\mathbf{x}-\mathbf{x}^*\mathbf{B}\mathbf{x}|}{\mathbf{x}^*\mathbf{Q}\mathbf{x}+\mathbf{x}^*\mathbf{B}\mathbf{x}}$ is most likely close to the ratio $\frac{\lambda_{\min}(\mathbf{B})-\lambda_{\min}(\mathbf{Q})}{\lambda_{\min}(\mathbf{B})+\lambda_{\min}(\mathbf{Q})}$ instead of $\frac{\lambda_{\max}(\mathbf{Q})-\lambda_{\min}(\mathbf{B})}{\lambda_{\max}(\mathbf{Q})+\lambda_{\min}(\mathbf{B})}$ or $\frac{\lambda_{\max}(\mathbf{B})-\lambda_{\min}(\mathbf{Q})}{\lambda_{\max}(\mathbf{B})+\lambda_{\min}(\mathbf{Q})}$, i.e. when $\frac{\lambda_{\max(\mathbf{Q})}}{\lambda_{\min(\mathbf{Q})}}$ is large,

$$
\begin{aligned}
|r|_{bound} &\leq \sup_{\mathbf{x}\in\mathbf{C}^N;\mathbf{x}\neq\mathbf{0}} \left\{ \frac{|\mathbf{x}^*\mathbf{A}\mathbf{x} - \mathbf{x}^*\mathbf{Q}\mathbf{x}|}{|\mathbf{x}^*\mathbf{A}\mathbf{x}|} \right\} \\
&\approx \frac{\lambda_{\min}(\mathbf{B}) - \lambda_{\min}(\mathbf{Q})}{\lambda_{\min}(\mathbf{B}) + \lambda_{\min}(\mathbf{Q})}. \qquad (28)
\end{aligned}
$$

This implies that the convergence rate of EDS is limited by the condition number of matrix $\mathbf{Q}$ somewhat too. In addition, (28) also exploits the fact that the double direction search seems to converge more rapidly than the one direction search because $\lambda_{\min}(\mathbf{B})$ is smaller and more close to $\lambda_{\min}(\mathbf{Q})$ in EDS2.

Example 3.1: Consider a simple example of a single-input adaptive linear combiner with two weights. The input and desired signals are sampled sinusoids at the same frequency, with $M = 6$ samples per cycle. The input correlation matrix $\mathbf{Q}$ and the correlation vector $\mathbf{r}$ are calculated as follows:

$$
\begin{aligned}
\mathbf{Q} &= E\left\{ \begin{bmatrix} x_k^2 & x_k x_{k-1} \\ x_{k-1}x_k & x_{k-1}^2 \end{bmatrix} \right\} = \begin{bmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{bmatrix}; \\
\mathbf{r} &= E\left\{ \begin{bmatrix} d_k x_k & d_k x_{k-1} \end{bmatrix} \right\}^T = \begin{bmatrix} 0 & -\frac{\sqrt{3}}{2} \end{bmatrix}^T.
\end{aligned}
$$

Solving $\det(\lambda\mathbf{I} - \mathbf{Q}) = 0$, gives that $\lambda_{\min}(\mathbf{Q}) = 0.25$, $\lambda_{\max}(\mathbf{Q}) = 0.75$. From (24), $|r|_{bound} = \frac{1}{2}$.

In EDS, $\mathbf{A} = \begin{bmatrix} 0.5 & 0 \\ 0.25 & 0.5 \end{bmatrix}$; $\mathbf{B} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$.

$\lambda_{\min}(\mathbf{A}^{-1}\mathbf{Q}) = 0.75$, $\lambda_{\max}(\mathbf{A}^{-1}\mathbf{Q}) = 1$, and $|r|_{boun} = \frac{1}{7}$ based on (26). From (28) and $\lambda_{\min}(\mathbf{B}) = \lambda_{\max}(\mathbf{B}) = 0.5$, we conclude that $|r| \geq \frac{1}{3}$ minimally.

4 Wavelet Transform

In this section, wavelet transform is briefly overviewed. The simplest approach to start is to consider one-dimensional wavelet transform. Wavelet transform is a signal decomposition method using orthogonal basis. Starting with a signal $x(t)$, which can be expressed as a linear combination as,

$$x(t) = \sum_{l} d_l \psi_l(t) \tag{29}$$

where l is a integer index, d_l are decomposition coefficients, and $\psi_l(t)$ denotes the expansion set. For wavelet analysis, the expansion set is constructed from a mother wavelet, ψ. The scaling and translating version of the mother wavelet, $\psi_{j,k}(t)$, produces more orthonormal basis, which are,

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi\left(2^j t - k\right) \tag{30}$$

where j, k are integers. Hence equation (29) becomes

$$\begin{aligned} x(t) &= \sum_{j} \sum_{k} d_{j,k} \psi_{j,k}(t) && (31) \\ &= \sum_{j} \sum_{k} d_{j,k} 2^{\frac{j}{2}} \psi\left(2^j t - k\right). && (32) \end{aligned}$$

In multiresolution analysis, there is another basic function called scaling function, $\varphi(t)$, in addition to wavelet $\psi(t)$. These two functions are highly related and used to form subspaces in $L^2(R)$. The relationship of subspaces can be written as,

$$V_0 \subset V_1 \subset V_2 \subset \ldots \subset L^2$$

where V_j denotes level-j subspace spanned by scaling function $\varphi_{j,k}(t)$, $\varphi_{j,k}(t) = 2^{\frac{j}{2}} \varphi(2^j t - k)$. The difference between basis in subspace V_j and V_{j+1} is the wavelet basis $\psi_{j,k}(t)$. In other words, wavelet function $\psi_{j,k}(t)$ spans subspace $W_j = V_{j+1} - V_j$. It is required that $\langle \varphi_{j,k}(t), \psi_{j,k}(t) \rangle = 0$, for all appropriate j, k, and l. Consequently, wavelet expansion yields

$$x(t) = \sum_{k} c_{0,k}\ \varphi_{0,k}(t) + \sum_{k} \sum_{j=0}^{\infty} d_{j,k}\ \psi_{j,k}(t) \tag{33}$$

where

$$\begin{aligned} c_{j,k} &= \left\langle x(t), \varphi_{j,k}(t) \right\rangle = \int x(t)\, \varphi_{j,k}(t)\ dt \\ d_{j,k} &= \left\langle x(t), \psi_{j,k}(t) \right\rangle = \int x(t)\, \psi_{j,k}(t)\ dt \end{aligned}$$

In discrete time signal applications, the digital wavelet transform (DWT) operation can be done by using filter banks. The configuration of DWT filter bank is composed of a number of high pass filters, low pass filters, and decimators. Conversely, to perform inverse digital wavelet transform (IDWT), some corresponding high pass filters, low pass filters, and interpolators are needed.

In order to transform a signal from time domain into wavelet domain a transformation matrix, **T** is required. Basically, the elements in the transformation matrix are obtained from the coefficients of high pass filters and low pass filters. The details of constructing the transformation matrix, **T** can be found in [2] and [13]. This matrix is given by

$$\mathbf{X}(n) = \mathbf{T}\mathbf{x}(n) \tag{34}$$

where $\mathbf{X}(n)$ is the transformed signal in the wavelet domain and $\mathbf{x}(n)$ is a signal in the time domain.

Now consider 2-band two-dimensional wavelet transform, the same manner as one-dimension is used to decompose data into subbands except that two scaling functions, $\varphi_{j,k}(t)$ and $\phi_{j,k}(t)$, perform wavelet transform in the horizon and vertical directions. The transformation matrix, **T**, for two-dimensional wavelet transform can be calculated by using those scaling functions. The transformation equation can also be written as,

$$\mathbf{X}_{LL} = \mathbf{T}_{LL}\mathbf{x} \tag{35}$$

$$\mathbf{X}_{LH} = \mathbf{T}_{LH}\mathbf{x} \tag{36}$$

$$\mathbf{X}_{HL} = \mathbf{T}_{HL}\mathbf{x} \tag{37}$$

$$\mathbf{X}_{HH} = \mathbf{T}_{HH}\mathbf{x} \tag{38}$$

where X_{PQ} are images in wavelet domain resulted from performing P operator in horizontal and Q operator in vertical direction. H and L denote high pass and low pass filtering respectively. And x is an image in time domain.

4.1 Wavelet Transform Based Adaptive Filtering

Wavelet transform is a unitary transform that decomposes a signal by using wavelet basis functions. Some work on wavelet based adaptive filtering [1], [2], [3], [4], [5], indicated advantages in the wavelet domain over time domain. The impact of these results to the EDS algorithm comes from subband decomposition properties of the input images. Since each decomposed image contains different frequency components, they are orthogonal to each other. This orthogonality of the transformation results in faster convergence of the EDS.

From the cost function in equation (1),

$$J_n(\mathbf{w}) = \min_{\mathbf{w} \in R^N} \left\{ \mathbf{w}^T \mathbf{Q}(n) \mathbf{w} - 2\mathbf{w}^T \mathbf{r}(n) + \sigma^2(n) \right\}$$

we have,

$$\begin{aligned} \mathbf{Q}(n) &= E\left[\mathbf{x}(n)\mathbf{x}(n)^T\right] \\ \mathbf{r}(n) &= E\left[d(n)\mathbf{x}(n)\right] \end{aligned}$$

In the wavelet domain, we transform the input signal x, using equation (34) for one-dimensional and equations (35)-(38) for two-dimensional. The corresponding correlation matrix $\mathbf{Q}_w$ and cross correlation vector $\mathbf{r}_w$ are,

$$\begin{aligned} \mathbf{Q}_w(n) &= E\left[\mathbf{X}(n)\mathbf{X}(n)^T\right] \\ &= E\left[\mathbf{T}\mathbf{x}(n)\mathbf{x}(n)\mathbf{T}^T\right] \\ &= \mathbf{T}E\left[\mathbf{x}(n)\mathbf{x}(n)^T\right]\mathbf{V}^T \\ &= \mathbf{T}\mathbf{Q}\mathbf{T}^T, \end{aligned} \tag{39}$$

$$\begin{aligned} \mathbf{r}_w(n) &= E\left[d(n)\mathbf{X}(n)\right] \\ &= E\left[d(n)\mathbf{T}\mathbf{x}(n)\right] \\ &= \mathbf{T}E\left[d(n)\mathbf{x}(n)\right] \\ &= \mathbf{T}\mathbf{r}(n) \end{aligned} \tag{40}$$

The Wiener optimal solution in the wavelet domain is

$$\mathbf{w}_{w_{opt}} = \mathbf{Q}_w^{-1}\mathbf{r}_w \tag{41}$$

$$= \left[\mathbf{T}\mathbf{Q}\mathbf{T}^T\right]^{-1}\mathbf{T}\mathbf{r}. \tag{42}$$

4.2 Image restoration in Wavelet Domain

In our previous study [8], we have shown the outstanding SNR improvement in image restoration application using FEDS algorithm in time domain. The corrupted noise in testing images is removed by linear predictor using $(K-1) \times (L-1)$ FIR filter. The output of the filter can be obtained as

$$y(n_1, n_2) = \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} w_{ij} x(n_{1-i}, n_{2-j}) + w_{00}$$

where $(i, j) \neq (0, 0)$, and $x(n_1, n_2)$ is the noisy image. Because of the fact that images have non zero mean, coefficient w_{00} is added into the above equation. From previous section, we know that wavelet transformation can be done by using equations (35)-(38). For M-band wavelet transform, where $M = 2$, we transform two dimensional data in each level. As a result, there are 4 transformed images which contain different information. Each image is restored in the wavelet domain separately as shown in Figure 1. As a result, we have 4 restored images, Y's, in wavelet domain as follows:

$$\begin{aligned} Y_{LL}(n_1, n_2) &= \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} w_{ij} X_{LL}(n_{1-i}, n_{2-j}) + w_{00} && (43) \\ Y_{LH}(n_1, n_2) &= \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} w_{ij} X_{LH}(n_{1-i}, n_{2-j}) + w_{00} && (44) \\ Y_{HL}(n_1, n_2) &= \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} w_{ij} X_{HL}(n_{1-i}, n_{2-j}) + w_{00} && (45) \\ Y_{HH}(n_1, n_2) &= \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} w_{ij} X_{HH}(n_{1-i}, n_{2-j}) + w_{00} && (46) \end{aligned}$$

The final image in time domain can be obtained by inverse wavelet transforming of these 4 restored images, i.e. Y_{LL}, Y_{LH}, Y_{HL}, Y_{HH}.

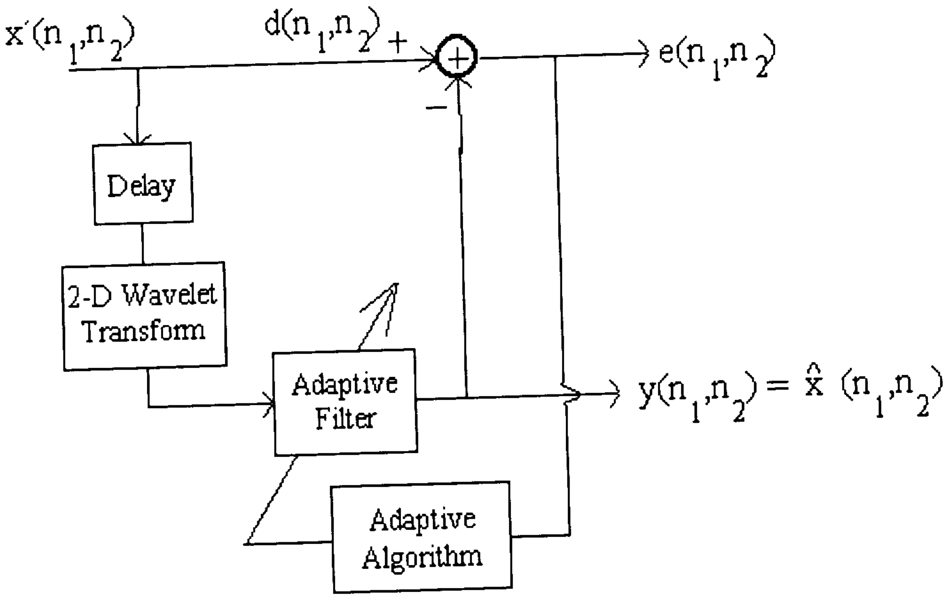


Figure 1: Adaptive predictor configuration

5 Simulation Results

5.1 Image Restoration

To demonstrate the performance of the EDS algorithm in the wavelet domain, we conducted simulations for image restoration in both time and wavelet domain. In this simulation, 1-level wavelet transform, with Daubechie-8 basis, is applied to lena and clown images. The restoration process using FEDS algorithm is performed in each subband image and then the restored image is obtained by inverse wavelet transform of those subband restored images. The simulation results is shown as the improvement of SNR (SNRI) in Table 3.

	SNR = -1.8dB		SNR = -7dB	
	Time Domain	Wavelet Domain	Time Domain	Wavelet Domain
Lena	9.4	9.9	11.9	13.1
Saturn	7.7	8.7	11.0	12.4

Table 3. Comparison of SNRI using FEDS algorithm between in time and wavelet domain.

It is obvious that EDS algorithm and FEDS algorithm in wavelet domain exhibit improvement over their time domain counterparts.

5.2 Image Restoration with Lost Samples

In real world applications in communication systems, received images at the receiving end are not only corrupted by noise but also lose some data or samples. Therefore, it is necessary to recover those lost data points prior to restoring the images. The configuration of the system is shown in Fig 2. x_s is an image with lost sample. x' is an image with reconstruction of lost samples, and $\hat{x}$ is a final restored image.

To reconstruct the lost samples or data in an image, method described in [11] is used. The reconstruction subsystem is illustrated in Fig 3. x_s is a lost sample image which is an input to the reconstruction system. $x_{s_{lp}}$ is obtained by low pass filtering x_s. The input image is also, parallely, fed to hard limiter and rectifier to produce x_p. Therefore, it is obviously seen that x_p represents sequence of impulses where the lost samples are located. $x_{p_{lp}}$ is a low pass filtered version of x_p. Division of $x_{s_{lp}}$and $x_{p_{lp}}$ yields an image without lost of samples, x'. Detailed derivation of the reconstruction of lost samples in one dimension data can be found in [11]. Now, we can perform noise cancellation of the reconstructed image by passing x' to the adaptive predictor which is operating in wavelet domain to finally obtain an estimate of the original image, $\hat{x}$.

	SNR = -1.8dB	SNR = -7dB
Lena	12.1	12.5
Saturn	8.6	11.0

Table 4. SNRI using FEDS algorithm in wavelet domain with 20% lost of samples of noisy images.

Table 4 and Figures 4 and 5 are the simulation results when 20% of samples in the noisy images are lost. From the images and the SNRI numbers, it is clear that the developed system achieves significant improvement in image restoration.

6 Multiplierless Filters

Multiplierless digital filters are important due to their high computational speed. In addition, when these filters are implemented in hardware, the elimination of the multipliers results in a reduction in cost. In two dimensional

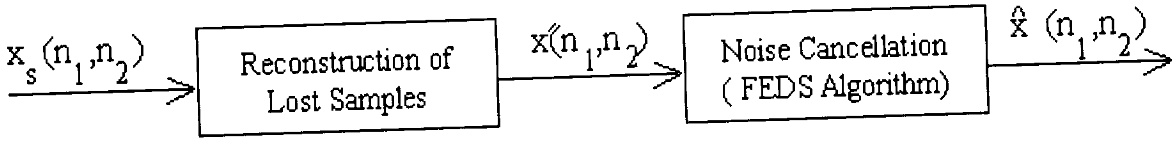

$x_s(n_1,n_2)$
Reconstruction of Lost Samples
$x(n_1,n_2)$
Noise Cancellation (FEDS Algorithm)
$\hat{x}(n_1,n_2)$

Figure 2: Image restoration block diagram

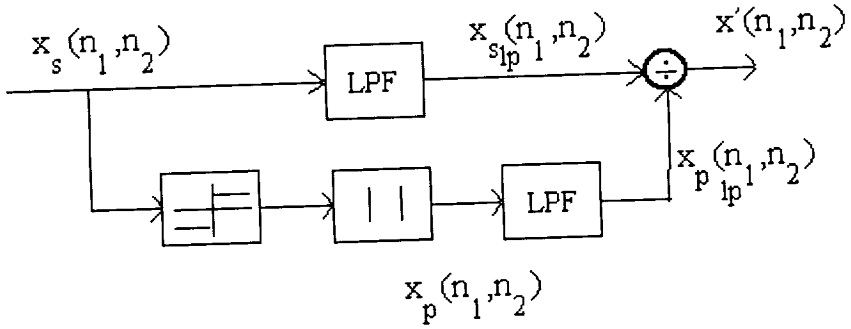
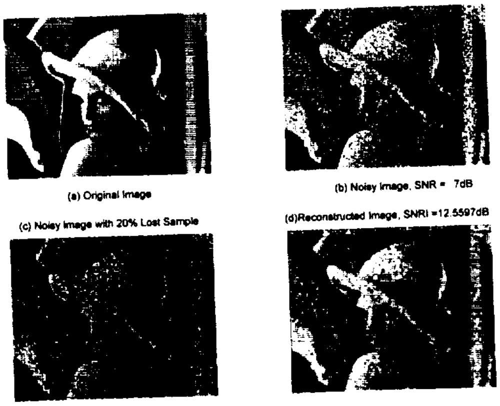


Figure 3: Reconstruction of lost sample subsystem

(2-D) signal processing, these are very significant advantages because of the large amounts of filtering required in such applications as image and video processing. Several techniques have been developed for designing 2-D multiplierless FIR and IIR filters during the past decade [18]-[21]. Design methods for 2-D state space multiplierless filters based on a generic algorithm have been given in [18],[19]. Design of multiplierless FIR filters using a McClellan transformation were presented in [20],[21]. For the one-dimensional case, there are also various techniques proposed for designing multiplierless filters such as a design which considers simplified parallel implementations of FIR filters for signed powers-of-two implementation [22],[23] and a design of multiplierless IIR elliptic filters based on sensitivity analysis[24]. Also, multiplierless filters have been designed using periodically shift-variant (PSV) filters [25]. The use of periodic filters increases the total number of coefficients, thereby providing the necessary degree of freedom for obtaining power-of-two coefficients. If the coefficients are all power-of-two, then they can be implemented using simple shifting operations only.

In this project, we derive the mathematical tools required to design 2-D PSV filters with power-of-two coefficients. We start with a given filter in



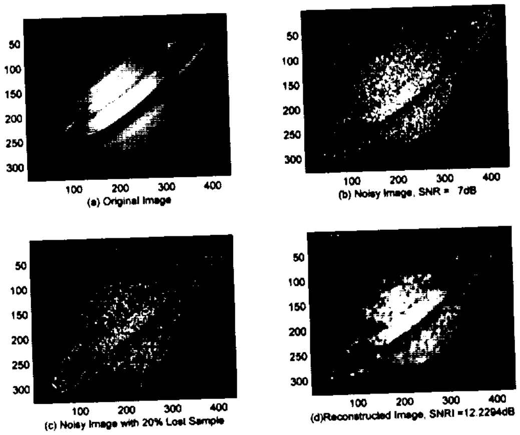
(a) Original Image

(b) Noisy Image, SNR = 7dB

(c) Noisy Image with 20% Lost Sample

(d)Reconstructed Image, SNRI =12.5597dB

Figure 4: Image Restoration with Lena



(a) Original Image

(b) Noisy Image, SNR = 7dB

(c) Noisy Image with 20% Lost Sample

(d)Reconstructed Image, SNRI =12.2294dB

Figure 5: Image Restoration with Saturn

the state-space domain and convert it to an equivalent PSV filter. Then the coefficients of this filter can be designed to be power-of-two. An example is given to illustrate the design methodology.

6.1 2-D State-Space System

The state-space model used in this paper is the second model of Fornasini-Marchesini (FM) [26]. For a linear shift variant (LSV) system, this model can be written as

$$\begin{aligned} x\,(h+1,k+1) &= A_1(h,k)x(h,k+1)+ \\ & \quad A_2(h,k)x(h+1,k)+ \\ & \quad A_0(h,k)x(h,k)+ \\ & \quad B(h,k)u(h,k) \\ y(h,k) &= C(h,k)x(h,k) \end{aligned} \tag{47}$$

where $x(h,k)$ is an $L \times 1$ state vector, $u(h,k)$ is a scalar input, $y(h,k)$ is a scalar output, $A_0(h,k), A_1(h,k), A_2(h,k)$ are $L \times L$ state matrices, $B(h,k)$ is an $L \times 1$ vector, and $C(h,k)$ is a $1 \times L$ vector.

For the special case when the state matrices are related as

$$A_0(h,k) = -A_1(h,k)A_2(h,k)$$

and

$$A_1(h,k)A_2(h,k) = A_2(h,k)A_1(h,k),$$

the system is called the Attasi model. A filter is called a periodically shift varying (PSV) filter with period (P,Q) if all the coefficients of the state matrices are periodic with a period (P,Q) i.e. $A(h+mP,k+nQ) = A(h,k)$ for all integers (m,n).

For simplicity, we assume that $C(h,k)$ is a constant vector C. Then, the impulse response at position (m,n) due to an input at position (i,j) $(h(m,n;i,j))$can be derived in the following way.

Let $u(i,j) = \delta(i,j)$, Find $y(m,n)$ for $m = i, i+1, i+2, ...$ and $n = j, j+1, j+2, ...$ in a recursive way and then use an induction method to get a closed form for an impulse response assuming a zero initial condition (i.e. $x(h,k) = 0 \quad for\ h < 1\ or\ k < 1$).

For m = i,
(m,n)= (i,j); $y(m,n) = Cx(i,j) = 0$;
(m,n)= $(i,j+1)$; $y(m,n) = Cx(i,j+1) = 0$;
(m,n)= $(i,j+2)$; $y(m,n) = Cx(i,j+2) = 0$;
(m,n)= $(i,j+3)$; $y(m,n) = Cx(i,j+3) = 0$;
(m,n)= $(i,j+4)$; $y(m,n) = Cx(i,j+4) = 0$.
For m = i+1,
(m,n)= $(i+1,j)$ gives
$y(m,n) = Cx(i+1,j) = 0$.
(m,n)= $(i+1,j+1)$ gives

$$\begin{aligned} y(m,n) &= Cx(i+1,j+1) \\ &= CB(i,j)u(i,j) \\ &= CB(i,j)\delta(i,j) = CB(i,j). \end{aligned}$$

(m,n)= $(i+1,j+2)$ gives

$$\begin{aligned} y(m,n) &= Cx(i+1,j+2) \\ &= C[A_0(i,j+1)x(i,j+1)+ \\ &\quad A_1(i,j+1)x(i,j+2)+ \\ &\quad A_2(i,j+1)x(i+1,j+1)] \\ &= CA_2(i,j+1)B(i,j). \end{aligned}$$

(m,n)= $(i+1,j+3)$ gives

$$\begin{aligned} y(m,n) &= Cx(i+1,j+3) \\ &= C[A_0(i,j+2)x(i,j+2)+ \\ &\quad A_1(i,j+2)x(i,j+3)+ \\ &\quad A_2(i,j+2)x(i+1,j+2)] \\ &= C[A_2(i,j+2)A_2(i,j+1)]B(i,j). \end{aligned}$$

For m = i+2,
(m,n)= $(i+2,j)$ gives
$y(m,n) = Cx(i+2,j) = 0$.
(m,n)= $(i+2,j+1)$ gives

$$\begin{aligned} y(m,n) &= Cx(i+2,j+1) \\ &= C[A_0(i+1,j)x(i+1,j)+ \\ &\quad A_1(i+1,j)x(i+1,j+1)+ \\ &\quad A_2(i+1,j)x(i+2,j)] \\ &= CA_1(i+1,j)B(i,j). \end{aligned}$$

(m,n)= $(i+2,j+2)$ gives

$$\begin{aligned} y(m,n) &= Cx(i+2,j+2) \\ &= C[A_0(i+1,j+1)x(i+1,j+1)+ \\ &\quad A_1(i+1,j+1)x(i+1,j+2)+ \end{aligned}$$

$$\begin{aligned}
&A_2(i+1,j+1)x(i+2,j+1)] \\
&= C[A_0(i+1,j+1)+ \\
&\quad A_1(i+1,j+1)A_2(i,j+1)+ \\
&\quad A_2(i+1,j+1)A_1(i+1,j)]B(i,j).
\end{aligned}$$

(m,n)= $(i+2,j+3)$ gives

$$\begin{aligned}
y(m,n) &= Cx(i+2,j+3) \\
&= C[A_0(i+1,j+2)x(i+1,j+2)+ \\
&\quad A_1(i+1,j+2)x(i+1,j+3)+ \\
&\quad A_2(i+1,j+2)x(i+2,j+2)] \\
&= C[A_0(i+1,j+2)A_2(i,j+1)+ \\
&\quad A_1(i+1,j+2)A_2(i,j+2)A_2(i,j+1)+ \\
&\quad A_2(i+1,j+2)\{A_0(i+1,j+1)+ \\
&\quad A_1(i+1,j+1)A_2(i,j+1)+ \\
&\quad A_2(i+1,j+1)A_1(i+1,j)\}]B(i,j).
\end{aligned}$$

For m = i+3,

(m,n)= $(i+3,j)$ gives

$$y(m,n) = Cx(i+3,j) = 0.$$

(m,n)= $(i+3,j+1)$ gives

$$\begin{aligned}
y(m,n) &= Cx(i+3,j+1) \\
&= C[A_0(i+2,j)x(i+2,j)+ \\
&\quad A_1(i+2,j)x(i+2,j+1)+ \\
&\quad A_2(i+2,j)x(i+3,j)] \\
&= C[A_1(i+2,j)A_1(i+1,j)]B(i,j).
\end{aligned}$$

For m = i+4,

(m,n)= $(i+4,j)$ gives

$$y(m,n) = Cx(i+4,j) = 0.$$

(m,n)= $(i+4,j+1)$ gives

$$\begin{aligned}
y(m,n) &= Cx(i+4,j+1) \\
&= C[A_0(i+3,j)x(i+3,j)+ \\
&\quad A_1(i+3,j)x(i+3,j+1)+ \\
&\quad A_2(i+3,j)x(i+4,j)] \\
&= C[A_1(i+3,j)A_1(i+2,j) \\
&\quad A_1(i+1,j)]B(i,j).
\end{aligned}$$

Since the input is an impulse $\delta(i,j)$, the output $y(m,n)$ is the impulse response $h(m,n;i,j)$. By induction, we can see that $for\ m \geq i+1, n \geq j+1$,

$$h(m,n;i,j) = CA(m-1,n-1)B(i,j) \tag{48}$$

where $A(i,j) = I_N$ (N*N Identity matrix)
$A(m,n) = 0, \quad m < i \; and \, n < j$
and $for \; m \geq i, n \geq j$

$$\begin{aligned} A(m,n) \;\; = \;\; & A_0(m,n)A(m-1,n-1) + \\ & A_1(m,n)A(m-1,n) + \\ & A_2(m,n)A(m,n-1). \end{aligned} \tag{49}$$

The above impulse response is for a general LSV system without periodicity. Now we specialize it to a linear shift invariant (LSIV) system, the model of which can be written as

$$\begin{aligned} x(h+1,k+1) \;\; = \;\; & \overline{A_1}x(h,k+1)+ \\ & \overline{A_2}\,x(h+1,k)+ \\ & \overline{A_0}x(h,k)+ \\ & \overline{B}u(h,k) \\ y(h,k) \;\; = \;\; & \overline{C}\,x(h,k) \end{aligned} \tag{50}$$

with $\quad \overline{A_0} = -\overline{A_1 A_2} \quad$ and $\quad \overline{A_1 A_2} = \overline{A_2 A_1}.$

The impulse response $h(n_1,n_2)$ of this LSIV system can be derived using (48) and (49) with all constant coefficients. The details are omitted for brevity. The final result is

$$h(n_1,n_2) = \overline{CA_1}^{(n_1-1)}\overline{A_2}^{(n_2-1)}\overline{B}u(n_1-1,n_2-1). \tag{51}$$

6.2 Multiplier-free PSV structure

It will be shown in this section that a 2-D system consisting of a Hold (which increases the signal rate by a factor of (1,Q)), a PSV system, and a decimation (which decreases the signal rate by a factor of (1,Q)) connected in cascade is a shift-invariant system. Thus we can realize a shift-invariant system using a PSV structure as shown in Fig.1. For a multiplier-free

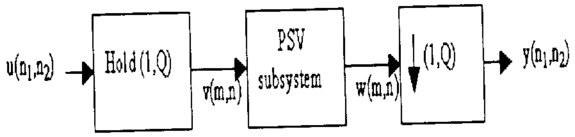


Figure 6: LSIV equivalent system

realization, the elements of matrices are restricted in a power-of-two set $S_n = \{\pm 2^{-n}, \pm 2^{-n+1}, ..., \pm 1, 0\}$.

As a first step towards deriving an impulse response of the system of Fig.1, we find the impulse response of the PSV system.

For simplicity, we make the following assumptions: (a) In (47), let $A_1(h,k)$ be a constant matrix A_1; (b) A_1 is commutative with $A_2(h,k)$; (c) $A_2(h,k)$ are commutative among themselves and periodic with period (1,Q).

Using $h(m,n;i,j)$ in section 2 and replace matrices A_1, A_2 and A_0 with a constant matrix A_1, periodic(1,Q) $A_2(h,k)$ (e.g. $A_2(h+i, k+nQ) = A_2(h,k)$ for all integer (i, n)) and $A_0(h,k) = -A_1A_2(h,k)$ respectively, then the PSV impulse response at any position (m,n) due to an input at (i,j) can be derived as:

$$h(m,n;i,j) = CX(m,n)B(i,j)$$

where $X(m,n)$ is given by the following.
For m = i+1,
$X(i+1,j+1) = I_N$
$X(i+1,j+2) = A_2(i,j+1)$
$X(i+1,j+3) = A_2(i,j+1)A_2(i,j+2)$
$X(i+1,j+4)$
$= A_2(i,j+1)A_2(i,j+2)A_2(i,j+3)$
For m = i+2,
$X(i+2,j+1) = A_1$
$X(i+2,j+2) = A_1A_2(i,j+1)$
$X(i+2,j+3)$
$= A_1A_2(i,j+1)A_2(i,j+2)$
$X(i+2,j+4)$
$= A_1A_2(i,j+1)A_2(i,j+2)A_2(i,j+3)$.
For m =i+3,

$X(i+3,j+1) = A_1^2$
$X(i+3,j+2)$
$= A_1^2 A_2(i,j+1)$
$X(i+3,j+3)$
$= A_1^2 A_2(i,j+1) A_2(i,j+2)$
$X(i+3,j+4)$
$= A_1^2 A_2(i,j+1) A_2(i,j+2) A_2(i,j+3)$.
For m = i+4,
$X(i+4,j+1) = A_1^3$
$X(i+4,j+2)$
$= A_1^3 A_2(i,j+1)$
$X(i+4,j+3)$
$= A_1^3 A_2(i,j+1) A_2(i,j+2)$
$X(i+4,j+4)$
$= A_1^3 A_2(i,j+1) A_2(i,j+2) A_2(i,j+3)$.
From above,we can see that $X(m,n)$ can be written as
$for\ m \geq i+1\ and\ n \geq j+1$,

$$X(m,n) = A_1^{m-(i+1)} \prod_{a=j+1}^{n-1} A_2(i,a)$$

Thus, $for\ m \geq i+1, n \geq j+1$

$$\begin{aligned} h(m,n \ &: \ i,j) = CA_1^{m-(i+1)} \times \\ &\prod_{a=j+1}^{n-1} A_2(i,a) B(i,j) \end{aligned} \tag{52}$$

with $\prod_{a=j+1}^{n-1} A_2(i,a) = I_N$ for $n = j+1$.
Next, we find the overall impulse response of PSV system of Fig.1.
With $u(n_1,n_2) = \delta(n_1,n_2)$, we have
$v(m,n) = \sum_{i=0}^{0} \sum_{j=0}^{Q-1} \delta(m-i, n-j)$.
Since the summation over i is from 0 to 0, then i = 0.
Using the superposition principle, the output can be written as

$$\begin{aligned} y(n_1, n_2) &= w(n_1, n_2 Q) \\ &= \sum_{j=0}^{Q-1} h(n_1, n_2 Q : 0, j) \end{aligned} \tag{53}$$

where $h(n_1, n_2 Q : 0, j), 0 \leq j \leq Q-1$ can be expressed by using (52) in the following $for\ n_1 \geq 1, n_2 \geq 1$,

$$\begin{aligned} h(n_1, n_2 Q &: 0, j) = C\, A_1^{n_1 - 1} \times \\ & \prod_{a=j+1}^{n_2 Q - 1} A_2(0, a) B(0, j) \end{aligned} \tag{54}$$

where $\prod_{a=j+1}^{Q-1} A_2(0,a) = I_N\ when \quad j = Q-1$

Now, (54) can also be written in the following form by using the commutative property of matrices A$_1$ and A$_2$. $For\ n_1 \geq 1, n_2 \geq 1$,

$$\begin{aligned} h(n_1, n_2 Q &: 0, j) = C\, A_1^{n_1 - 1} \times \\ & \left(\prod_{a=j+1}^{(n_2-1)Q+j} A_2(0, a)) \right) \times \\ & \left(\prod_{a=(n_2-1)Q+j+1}^{n_2 Q - 1} A_2(0, a) \right) B(0, j) \end{aligned}$$

$$\begin{aligned} h(n_1, n_2 Q &: 0, j) = C\, A_1^{n_1 - 1} \times \\ & \left(\prod_{a=1}^{(n_2-1)Q} A_2(0, a) \right) \times \\ & \left(\prod_{a=j+1}^{Q-1} A_2(0, a) \right) B(0, j) \end{aligned} \tag{55}$$

Notice that the product over one period remains unchanged regardless of the starting index. Since the input is an impulse, the output $y(n_1, n_2)$ in (53) is the impulse response of the overall system. Combining equations (53) and (55), the impulse response of the PSV system is the following.

For $n_1 \geq 1, n_2 \geq 1$,

$$\begin{aligned} h(n_1, n_2) &= C\, A_1^{n_1 - 1} \left(\prod_{a=1}^{Q} A_2(0, a) \right)^{n_2 - 1} \\ &\quad \times \sum_{j=0}^{Q-1} \prod_{a=j+1}^{Q-1} A_2(0, a) B(0, j) \end{aligned} \tag{56}$$

Thus it is shown that the overall system is linear shift-invariant. Note that the above impulse response resembles that of the LSIV system in (51). Comparing (56) and (51), we get $\overline{C} = C$, $\overline{A_1} = A_1$,

$$\overline{A_2} = \prod_{a=1}^{Q} A_2(0, a), \tag{57}$$

$$\textit{and} \qquad \overline{B} = \sum_{j=0}^{Q-1} \left(\prod_{a=j+1}^{Q-1} A_2(0, a) \right) B(0, j). \tag{58}$$

If we restrict the elements of $\overline{C}$ and $\overline{A_1}$ to be power-of-two, then we have a system whose characterization can be represented by matrices of only powers-of-two coefficients.

There are several ways to find the elements of matrices A_2 and B which give the best approximation of the desired LSIV system such as using the idea from [25] , or using minimum mean square error criteria. The optimization method is under study at this time.

6.3 Design

It is assumed that matrices A_1 and A_2 are circulant matrices [25], [30]. That is A_1 and A_2 are commutative and all A_2 are commutative among themselves. In addition, we can simplify the problem using the properties of circulant matrices as shown below.

Any circulant matrix A of order $N \times N$ can be diagonalized as $A = F^{*}\Lambda F$, where Λ is the diagonal matrix of eigenvalues, F is a DFT matrix whose klth element is $\frac{1}{\sqrt{N}} \exp\left(-j\frac{2\pi(k-1)(l-1)}{N}\right), 1 \leq k, l \leq N$ and F^{*} is the complex conjugate of F which has property that $F^{*} = F^{-1}$. We diagonalize both sides of (57) to get

$$\overline{\Lambda_2} = \prod_{a=1}^{Q} \Lambda_2(0,a) \tag{59}$$

Let $F\overline{B} = \overline{\beta}$ and $FB = \widetilde{B}$. Then from (58) we have

$$\begin{aligned} \overline{\beta} &= \sum_{j=0}^{Q-1} F\left(\prod_{a=j+1}^{Q-1} A_2(0,a)\right) F^{*}FB(0,j) \\ &= \sum_{j=0}^{Q-1} \left(\prod_{a=j+1}^{Q-1} \Lambda_2(0,a)\right) \widetilde{B}(0,j) \end{aligned} \tag{60}$$

Using the criteria based on a square error in time domain,

$$E = \sum_{n_1}\sum_{n_2} \left(h_d(n_1,n_2) - h(n_1,n_2)\right)^2 \tag{61}$$

where $(n_1, n_2) \in R_d$, the region of support of $h_d(n_1, n_2)$,which is a desired impulse response.

Assume that a desired filter is given in a state-space form as in(51).

$for\ n_1 \geq 1, n_2 \geq 1,$

$$h_d(n_1,n_2) = \overline{CA_1}^{(n_1-1)}\overline{A_2}^{(n_2-1)}\overline{B} \tag{62}$$

$otherwise, \quad h_d(n_1,n_2) = 0$

Using (56) for $h(n_1, n_2)$ with $C = \overline{C}, A_1 = \overline{A}_1$, and since all matrices are circulant, we can diagonalize (62) and (56) using a DFT matrix. Let $\overline{C}F^{*} = \widetilde{C}$ and combine with (59) and (60) to get

$$\begin{aligned} h_d(n_1,n_2) &= \overline{C}F^{*}F\overline{A_1}^{(n_1-1)}F^{*}F\overline{A_2}^{(n_2-1)}F^{*}F\overline{B} \\ &= \overline{C}F^{*}(F\overline{A_1}^{(n_1-1)}F^{*})(F\overline{A_2}^{(n_2-1)}F^{*})F\overline{B} \\ &= \widetilde{C}\overline{\Lambda_1}^{(n_1-1)}\overline{\Lambda_2}^{(n_2-1)}\overline{\beta} \end{aligned}$$

$$h(n_1,n_2) = \overline{C}F^{*}F\overline{A_1}^{(n_1-1)}F^{*}F$$

$$
\begin{aligned}
&(\prod_{a=1}^{Q} A_2(0,a))^{n_2-1} F^* F \\
&\sum_{j=0}^{Q-1} (\prod_{a=j+1}^{Q-1} A_2(0,a)) F^* F B(0,j) \\
= &\overline{C} F^* (F \overline{A_1}^{(n_1-1)} F^*) \\
&(\prod_{a=1}^{Q} F A_2(0,a) F^*)^{n_2-1} \\
&\sum_{j=0}^{Q-1} (\prod_{a=j+1}^{Q-1} F A_2(0,a) F^*) F B(0,j) \\
= &\widetilde{C} \overline{\Lambda_1}^{(n_1-1)} (\prod_{a=1}^{Q} \Lambda_2(0,a))^{n_2-1} \\
&\sum_{j=0}^{Q-1} (\prod_{a=j+1}^{Q-1} \Lambda_2(0,a)) \widetilde{B}(0,j)
\end{aligned}
$$

Thus the square error in time domain (61)can be written as

$$
E = \sum_{n_1} \sum_{n_2} \left(\begin{array}{c} \widetilde{C} \overline{\Lambda_1}^{(n_1-1)} (\overline{\Lambda_2}^{(n_2-1)} \overline{\beta} - \\ (\prod_{a=1}^{Q} \Lambda_2(0,a))^{n_2-1} \\ \sum_{j=0}^{Q-1} (\prod_{a=j+1}^{Q-1} \Lambda_2(0,a)) \widetilde{B}(0,j)) \end{array} \right)^2
$$

Note that instead of finding combination of circulant matrices A$_2$(0,a), we find only combinations of diagonal matrices of eigenvalues of $\Lambda_2(0,a)$.

6.4 Example

Consider a LSIV 2-D Attasi's model filter using circulant matrices as following

$$
\overline{A_1} = \begin{bmatrix} 0.5 & -0.5 & 0.125 & -0.125 \\ -0.125 & 0.5 & -0.5 & 0.125 \\ 0.125 & -0.125 & 0.5 & -0.5 \\ -0.5 & 0.125 & -0.125 & 0.5 \end{bmatrix}
$$

$$
\overline{A_2} = \begin{bmatrix} 0.5 & 0 & -0.015 & 0.25 \\ 0.25 & 0.5 & 0 & -0.015 \\ -0.015 & 0.25 & 0.5 & 0 \\ 0 & -0.015 & 0.25 & 0.5 \end{bmatrix}
$$

$$
\overline{B} = \begin{bmatrix} 1 & 0.39 & -1 & 0.45 \end{bmatrix}^T
$$

$$
\overline{C} = \begin{bmatrix} 1 & -1 & -1 & 1 \end{bmatrix}
$$

Using the criteria (61) above with periodicity Q = 2 and elements of matrices belong to set $S_2 = \{-1, \frac{-1}{2}, \frac{-1}{4}, 0, \frac{1}{4}, \frac{1}{2}, 1\}$, the designed matrices are obtained in the following:

$A_1 = \overline{A_1}$ *and* $C = \overline{C}$

$$A_2(0,0) = \begin{bmatrix} 0.5 & 0.5 & 0 & -0.5 \\ -0.5 & 0.5 & 0.5 & 0 \\ 0 & -0.5 & 0.5 & 0.5 \\ 0.5 & 0 & -0.5 & 0.5 \end{bmatrix}$$

$$A_2(0,1) = \begin{bmatrix} 0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0.5 \end{bmatrix}$$

$$B(0,0) = \begin{bmatrix} -1 & -1 & -0.25 & -1 \end{bmatrix}^T$$

$$B(0,1) = \begin{bmatrix} 1 & -0.25 & -1 & -0.5 \end{bmatrix}^T$$

This design yields an error E = 0.0108. The frequency responses of the desired LSIV filter and the designed multiplierless filter are quite well matched as shown in Fig. 7.

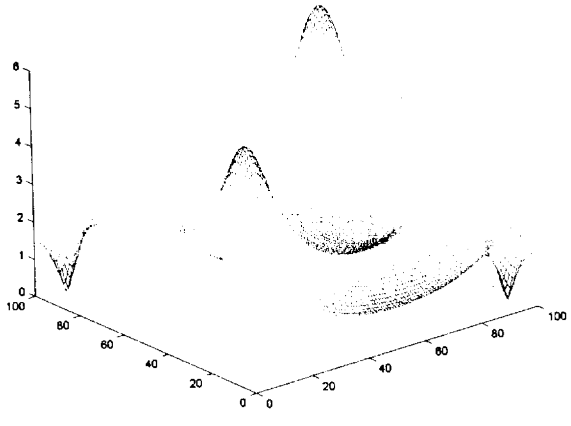


Fig. 7a.) Desired magnitude response.

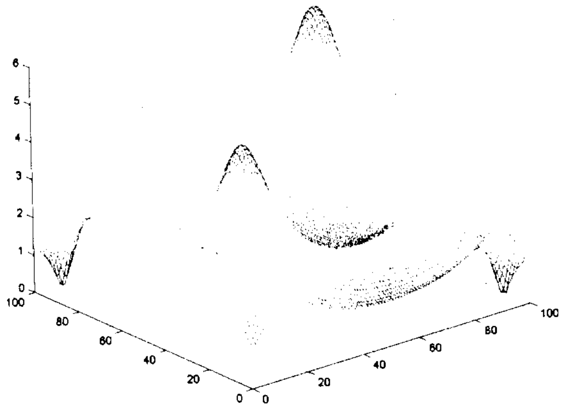


Fig. 7b.) Designed magnitude response.

7 Conclusion

In this project, we have developed a new adaptive filtering algorithm for image restoration and compression. The algorithm is called the Fast Euclidean Direction Search (FEDS) method. We have presented the derivation of this algorithm and all its virtues. This algorithm has then been implemented in the wavelet domain in conjunction with an algorithm for lost sample recovery. The overall system is then shown to perform efficiently for image restoration. We have also presented some new results on the design of 2-D multiplierless filters. These filters will be used in the future for image and video processing applications.

References

[1] S. Hosur and A. H. Tewfik, "Wavelet Transform Domain Adaptive FIR Filtering," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 617-630, Mar. 1997.

[2] A. H. Tewfik and M. Kim, "Fast Positive Definite Linear System Solvers," *IEEE Trans. Signal Processing*, vol. 42, no. 3, pp. 572-585, Mar. 1994.

[3] N. Erdol and F. Basbug, "Wavelet Transform Based Adaptive Filters: Analysis and New results," *IEEE Trans. Signal Processing,* vol. 44, no. 9, pp. 2163-2171, Mar. 1997.

[4] M. E. Zervakis, T. M. Kwon, and J. S. Yang, "Multiresolution Image restoration in the Wavelet Domain ," *IEEE Trans. Circuits Syst.*, vol. 42, no. 9, pp. 578-591, Sept. 1995.

[5] M. Antonini, P. Mathieu, and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE Trans. Image Processing,* vol. 1, no. 2, pp. 205-220, Apr. 1992.

[6] G. F. Xu and T. Bose, "Analysis of the Euclidean Direction Set Adaptive Algorithm," *Proc. IEEE. Int. Conf.. Acoustics, Speech, and Signal Processing,* vol. 3, pp. 1689-1692, 1998.

[7] G. F. Xu, Fast Algorithm for Digital Filtering: Theory and Applications, Ph.D. dissertation, University of Colorado, Boulder,1998.

[8] G. F. Xu, T. Bose, and J. Thomas, "A Fast Adaptive Algorithm for Image restoration," *IEEE Trans. Circuits Syst.*, vol. 46, no. 1, pp. 216-220, Jan. 1999.

[9] G. F. Xu, T. Bose, and J. Schroeder, "Channel Equalization Using Euclidean Direction Search Based Adaptive Algorithm," Feb. 1998.

[10] T. Mathurasai, T. Bose, and D. M. Etter, "Decision Feedback Equalization Using an Euclidean Direction Based Adaptive Algorithm," University of Colorado, Boulder, May 1999.

[11] F. A. Marvasti, P. M. Clarkson, M. V. Dokic, "Reconstruction of Speech Signals with Lost Samples," *IEEE Trans. Signal Processing,* vol. 40, no. 12, pp. 2897-2903, Dec. 1992.

[12] S. Haykin, *Adaptive Filter Theory.* Upper Saddle river, NJ: Prentice-Hall, 1996.

[13] I. Daubechies, *Ten Lectures on Wavelets.* Philadelphia, PA: SIAM, 1992.R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[14] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[15] Guo Fang Xu, and T. Bose, "Analysis of the Euclidean direction set adaptive algorithm," Proc. of the International Conference on Acoustics, Speech and Signal Processing, pp. 1689-1692, April 1998.

[16] R. Bhatia, *Matrix Analysis,* Springer-Verlag New York, 1997.

[17] G.F. Xu, *Fast Algorithms for Adaptive Filtering: Theory and Applications*, Ph.D. Dissertation, University of Colorado at Boulder, 1999.

[18] Y.-H. Lee, M. Kawamata and T. Higuchi, "GA-based design of multiplierless 2-D state-space digital filters with low roundoff noise," *IEEE Proc.-Circuits Devices Syst.*, vol. 145, no. 2, pp. 118-124, Apr. 1998.

[19] ______, " Design of Multiplierless 2-D State-space Digital filters over a Powers-of-Two Coefficient Space," *IEICE Trans. Fundamentals*, vol. E79-A, no.3, pp.374-377, Mar. 1996.

[20] S. Pei and S. Jaw, " Efficient design of 2-D multiplierless FIR filters by transformation," *Proceedings ICASSP 87*, vol. 3, pp. 1669-72, 1987.

[21] H.K. Kwan and C.L. Chan, "Circularly symmetric two-dimensional multiplierless FIR digital filter design using an enhanced McClellan transformation," *IEE Proceedings*, vol. 136, no.3, pp. 129-134, June 1989.

[22] D. Li and Y.C. Lim, "Multiplierless realization of adaptive filters by nonuniform quantization of input signal," *1994 IEEE International Symposium on Circuits and Systems*, vol. 2, pp.457-459, 1994.

[23] M. Yagyn, A. Nishihara and N. Fujii, "Fast FIR digital filter structures using minimal number of adders and its application to filter design," *IEICE Trans. Fundamentals,* vol. E-79 A, no. 8, pp.1120-1128, Aug 1996.

[24] L.D. Milic' and M.D. Lutovac, " Design of multiplierless elliptic IIR filters with a small quantization error," IEEE Trans. on Signal Processing, vol.47, no.2, pp.469-479, Feb 1999.

[25] S. Ghanekar, S. Tantaratana and L.E. Franks, " Multiplier-Free IIR Filter Realization Using Periodically Time-Varying State-Space Structure-Part I: Structure and Design," *IEEE Trans. Signal Processing.*, vol. 42, no.5, pp.1008-1017, May 1994.

[26] S. G. Tzafestas, *Multidimensional Systems*, Marcel decker, Inc. 1985. ch. 2.

[27] E. Fornasini and G. Marchesini, "State-Space Realization Theory of Two-Dimensional Filters," *IEEE Trans. on Automat. Contr.*, vol. Ac-21, no.4, Aug 1976.

[28] W.S. Lu and A. Antoniou, *Two-dimensional Digital Filters*, Marcel decker, Inc. 1992.

[29] J.S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, 1990.

[30] P. J. Davis, *Circulant Matrix*, John Wiley & Sons, Inc., 1979.