

Analysis of Wavelet Transform-Domain LMS-Newton Adaptive Filtering Algorithms with Second-Order Autoregressive (AR) Process

Tanzila Lutfor^{*}, Md. Zahangir Alam, Sohag Sarker

School of Science and Engineering (SSE), UITS, Dhaka, Bangladesh

Abstract This paper analyze the stability, misadjustment, and convergence performance of the Wavelet Transform (WT) domain least mean square (LMS) Newton adaptive filtering algorithm with first order and second order autoregressive (AR) process. The wavelet transform domain signal provides a means of constructing of more orthonormal correlated input signals than other transform. The wiener filter with AR input process are assumed to be stationary, and the Stationary Wavelet Transform (SWT) is used as transform algorithm to provide more correlated input signal than other integral transform. The simulation result of this work shows that Wavelet-domain LMS-Newton algorithm provides better performance result than other transform domain algorithm for both first order and second order AR process. Structure of the SWT based LMS-Newton adaptive algorithm for time-varying process has been also discussed in this paper. Finally, Computer simulations on the SWT based LMS-Newton adaptive algorithms are demonstrated to validate the analysis presented in this paper, and the simulation result shows that the structure of SWT domain LMS-Newton adaptive algorithm provides better denoising of a noisy signal than other transform domain LMS-Newton adaptive algorithm.

Keywords Discrete Fourier Transform, Discrete Cosine Transform, Least Mean Square (LMS) Algorithm, Stationary Wavelet Transform (SWT), Autoregressive (AR) Process

1. Introduction

The computation complexity of the time-domain adaptive filtering algorithms such as LMS algorithms, gradient adaptive lattice, and Least Square algorithms increases linearly with the filter order[1]-[4], and it is difficult to use such filter in high speed real time applications[5]. The frequency domain LMS algorithms using fast Fourier Transform (FFT) reduces the number of mathematical computations and the orthogonal transform provides more computation counts[6]-[8]. The convergence rate depends on the input autocorrelation matrix and decreases radically with higher input correlation. The transform domain algorithms updated the filter coefficients that can be applied for the case of higher input correlation to obtain higher convergence rate[9][10]. The transform domain adaptive filter achieved rapid convergence of the filter coefficients for non-white input signal with a reasonably lower computational complexity than simpler LMS based system. The gradient based LMS algorithm having convergence rate dependency on the input signal statistics but the LMS-Newton (LMSN)

algorithm is another powerful alternatives for larger eigenvalue spread of the input correlation matrix.

The step size (μ) of LMS and the reference signal power play an important role in the stability and convergence of the LMS adaptation process. The fixed value of μ responds to stationary channel, and for the time-varying channel variable step-size LMS algorithm has been proposed to obtain better performance for non-stationary channel[11]-[13]. The Normalized LMS (NLMS) uses variable step-size (μ) algorithm to optimize convergence speed, stability, and performance. The time-variable step-size (μ) is implemented based on the power estimation of the input signals[14]. The larger μ is used for non-stationary and smaller μ is used for stationary without considering the convergence rate[15][16]. Further, modification on NLMS is proposed named as VS-NLMS in which μ is decided by the estimation of input signals, and it achieves a faster convergence rate with maintaining the stability of the NLMS[17]. The proposed VS-LMS with time-varying step-size (μ) improves the output signal to noise ratio (SNR) and the convergence speed. A new type of criteria is proposed using output error with step-size value (μ) from a big value α_2 and smaller one α_1 to provide faster tracing speed and smaller misadjustment[18]-[20]. The stability and convergence properties of transform - domain LMS algorithm has been investigated in[21], the authors also analyzed the effects of the transforms and power

^{*} Corresponding author:

lutfor_tanzila@yahoo.com (Tanzila Lutfor)

Published online at <http://journal.sapub.org/ajsp>

Copyright © 2013 Scientific & Academic Publishing. All Rights Reserved

normalization in various adaptive filters for both first order and second order AR process. In the same work they stated that the power normalization increases the difficulty for analyzing the stability and convergence performance. The LMS-Newton algorithm avoids the slow convergence of the LMS algorithm for highly correlated input signal[15], this property should be very useful in DWT-LMS algorithm and is used in this work.

In[21]-[23], the authors analyze the transform-domain adaptive filters with discrete Fourier transform (DFT), discrete Cosine transform (DCT), discrete Hartely transform (DHT), and discrete Sine transform (DST) for first-order and second-order AR input process with their stabilities and convergence performance for input power normalization. The result of the work[24] shows that the DCT-LMS and DST-LMS provides the better convergence performance for both first and second order AR process than the DFT-LMS and DHT-LMS. In this paper, we analyze the Discrete Wavelet Transform (DWT) in adaptive filtering algorithm in form as DWT-LMS Newton adaptive filters for both first order and second order AR process to enhance Misadjustment to measure the steady-state convergence, and MSE performance. This paper also discuss the design of the DWT-LMS for the time-varying AR process, the time-varying block by block basis data is processed by using ε -decimated DWT algorithm. The simulation study in our work shows that SWT-LMS Newton adaptive filter provides better stability, misadjustment, and convergence performances than that of DFT-LMS and DCT-LMS for both first order and second order AR process[23].

2. Analytical Form of Transform-Domain LMS-Newton Algorithm

The LMS-Newton algorithm estimates the second order statistics of the wide-sense stationary signal. The algorithm avoids the slow convergence of the LMS algorithm for highly correlated input signal. The LMS-Newton algorithm minimized the MSE at instant $(n+1)$ if[15]

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \hat{\mathbf{R}}^{-1}(n) \hat{\mathbf{g}}_w(n) \quad (1)$$

where μ is a convergence factor that is used to protect the algorithm from divergence due to noisy estimate of input auto-correlation matrix \mathbf{R} and the gradient vector $\mathbf{g}_w(\mathbf{R})$.

$\hat{\mathbf{R}}(n)$ is the estimation of the auto-correlation matrix, namely, $\mathbf{R} \approx E[\mathbf{u}(n)\mathbf{u}^H(n)]$, $\mathbf{u}(n)$ is the input signal vector. The unbiased estimate of \mathbf{R} for stationary input signal is as:

$$\hat{\mathbf{R}}(n) = \frac{1}{n+1} \sum_{i=0}^n \mathbf{u}(i)\mathbf{u}^H(i)$$

$$\hat{\mathbf{R}}(n) = \frac{n}{n+1} \hat{\mathbf{R}}(n-1) + \frac{1}{n+1} \mathbf{u}(n)\mathbf{u}^H(n) \quad (2)$$

Here, the Hermitian transpose (H) is used for input complex vector. Again, the estimation on both sides of (2)

gives,

$$E(\hat{\mathbf{R}}(n)) = \frac{1}{n+1} \sum_{i=0}^n E[\mathbf{u}(i)\mathbf{u}^H(i)] = \mathbf{R} \quad (3)$$

The estimation for \mathbf{R} requires infinite memory location for large n . Another form of estimation of the autocorrelation matrix by introducing a small factor α , in the range $0 < \alpha < 0.1$ to provide the good balance of input signal information. The resulting equation to calculate the inverse matrix $\hat{\mathbf{R}}^{-1}(n)$ that is required for LMS-Newton algorithm can be given using the Matrix inverse Lemma as[21]:

$$\hat{\mathbf{R}}^{-1}(n) = \frac{1}{1-\alpha(n)} \left[\hat{\mathbf{R}}^{-1}(n-1) - \frac{\hat{\mathbf{R}}^{-1}(n-1)\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{R}}^{-1}(n-1)}{\frac{1-\alpha(n)}{\alpha(n)} + \mathbf{u}^H(n)\hat{\mathbf{R}}^{-1}(n-1)\mathbf{u}(n)} \right] \quad (4)$$

with $\alpha(n) = \alpha$. The coefficient updating formula (1) can be written using the estimate of gradient vector as[22]:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + 2\mu \mathbf{e}(n) \hat{\mathbf{R}}^{-1}(n) \mathbf{u}(n) \quad (5)$$

where,

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{w}^H(n-1)\mathbf{u}(n) \quad (6)$$

$\mathbf{d}(n)$ =desired signal, $\mathbf{w}(n-1)$ =weighting vector at iteration $n-1$; $\hat{\mathbf{R}}^{-1}(-1) = \delta \mathbf{I}$ (δ is positive constant less than 0, and \mathbf{I} is unity matrix); $\mathbf{w}(0) = [0 \ 0 \ 0 \ \dots \ 0]^T$

The transform domain LMS transformed the input signal vector $\mathbf{u}(n)$ in a more convenient vector $\mathbf{s}(n)$, $\mathbf{s}(n) = \mathbf{T}\mathbf{u}(n)$, where $\mathbf{T}\mathbf{T}^H = \mathbf{I}$, the transform matrix of following form as:

$$\mathbf{T}\mathbf{N}(i,l) \approx \begin{cases} \mathbf{F}_N(i,l) = \frac{1}{\sqrt{N}} e^{j\frac{2\pi il}{N}}, & \text{DFT} \\ \mathbf{H}_N(i,l) = \frac{1}{\sqrt{N}} \left(\cos\left(\frac{2\pi il}{N}\right) + \sin\left(\frac{2\pi il}{N}\right) \right), & \text{DHT} \\ \mathbf{C}_N(i,l) = \sqrt{\frac{2}{\pi}} k_i \cos\left(\frac{\pi i(l+0.5)}{N}\right), & \text{DCT} \\ \mathbf{S}_N(i,l) = \sqrt{\frac{2}{N+1}} \sin\left(\frac{\pi il}{N+1}\right), & \text{DST} \\ \mathbf{W}_N(i,l) = \begin{bmatrix} \tilde{\psi}_{j,k} \\ \tilde{\phi}_{j,k} \end{bmatrix}; j=1, \dots, J; k=0,1, \dots, K-1; & \text{DWT} \end{cases} \quad (7)$$

where, $i, l=0, 1, \dots, N-1$; $k_i=1/\sqrt{2}$ for $i=0$ and 1 for otherwise, $\tilde{\psi}(t)$ and $\tilde{\phi}(t)$ are wavelet function and scaling function respectively. The autocorrelation matrix for the transform-domain input signal vector $\mathbf{s}(n)$ is given by:

$$\mathbf{R}_s = \mathbf{T}\mathbf{R}\mathbf{T}^H \quad (8)$$

The misadjustment of the adaptive filter is the ratio of the excess mean square error (MSE) and minimum MSE in the steady-state which is a measure of the noise in the filter output due to fluctuations in the filter coefficients. The formula for misadjustment based on the expression for weight noise covariance is given as[24][25]:

$$M = \frac{\mu \text{tr} \mathbf{R}}{1 - \mu \text{tr} \mathbf{R}} \quad (9)$$

The result can be used for the transform-domain LMS adaptive filters and is given by:

$$M_s = \frac{\mu \text{tr} \mathbf{R}_s}{1 - \mu \text{tr} \mathbf{R}_s} \quad (10)$$

Stability can be determined by the condition of M of finite value and we must choose μ within the range $0 < \mu < 1/\text{tr}(\mathbf{R})$; for the case of (9), and $0 < \mu < 1/\text{tr} \mathbf{R}_s$; for the case of (10).

3. Wavelet Packet Transform Based LMS Algorithm

The wavelet packet transform of the signal $\mathbf{x}(n)$ can denoted as [25][26]:

$$\mathbf{y}(n) = \mathbf{V}_p \mathbf{x}(n) \quad (11)$$

$\mathbf{y}(n)$ is the $M \times 1$ wavelet packet transform signal vector, \mathbf{V}_p the $2^J \times 2^J$ wavelet packet decomposition for the J series as follows [27]–[29]:

$$\mathbf{W}_p = \begin{bmatrix} \mathbf{V}_J & 0 & \cdot & 0 \\ 0 & \mathbf{V}_J & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \mathbf{V}_J \end{bmatrix} \times \dots \times \begin{bmatrix} \mathbf{V}_2 & 0 \\ 0 & \mathbf{V}_2 \end{bmatrix} \times \mathbf{V}_1 \quad (12)$$

The LMS algorithm is used as: $\mathbf{r}(n) = \mathbf{W}^T(n) \mathbf{y}(n)$

where \mathbf{W} is the weighted vector of LMS algorithm, $\mathbf{r}(n)$ is the output signal of LMS. The error of the filter is as:

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{r}(n) \quad (13)$$

where $\mathbf{d}(n)$ is the desired signal, and we have:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mu \mathbf{e}(n) \mathbf{r}^*(n) \quad (14)$$

where μ is the step-size parameter.

3.1. Algorithm Performance Analysis

The cross-correlation vector \mathbf{P}_y between the desired signal $\mathbf{d}(n)$ and the wavelet packet transform signal $\mathbf{y}(n)$ is as:

$$\mathbf{P}_y = E\{\mathbf{d}(n)\mathbf{y}(n)\} = E\{\mathbf{d}(n)\mathbf{V}_p \mathbf{x}(n)\} = \mathbf{V}_p \mathbf{P}_x \quad (15)$$

Where, \mathbf{P}_x is the cross-correlation between the input signal $\mathbf{x}(n)$ and the desired signal $\mathbf{d}(n)$. The auto-correlation matrix \mathbf{R}_y of the wavelet packet transform signal is as:

$$\mathbf{R}_y = E\{\mathbf{y}(n)\mathbf{y}^T(n)\} = E\{\mathbf{V}_p \mathbf{x}(n)\mathbf{x}^T(n)\mathbf{V}_p\} = \mathbf{V}_p \mathbf{R}_x \mathbf{V}_p \quad (16)$$

where \mathbf{R}_x is the autocorrelation matrix of the signal $\mathbf{x}(n)$. The wiener optimal solution of wavelet packet domain is as:

$$\mathbf{W}_{\text{opt}} = \mathbf{R}_y^{-1} \mathbf{P}_y = \mathbf{R}_y^{-1} \mathbf{V}_p \mathbf{P}_x = \mathbf{R}_y^{-1} \mathbf{V}_p \mathbf{R}_x \mathbf{P}_x = \mathbf{R}_y^{-1} \mathbf{V}_p \mathbf{R}_x \mathbf{W}_0 \quad (17)$$

Here, $\mathbf{W}_0 = \mathbf{R}_x^{-1} \mathbf{P}_x$; the time domain wiener solution. The MSE of wavelet packet domain is derived in appendix A and is given as:

$$J(\mathbf{W}) = \sigma_d^2 - \mathbf{W}^T(n) \mathbf{P}_y \quad (18)$$

The minimum error in the mean square error is as:

$$J_{\min} = \sigma_d^2 - \mathbf{W}_0 \mathbf{P}_y = \sigma_d^2 - \mathbf{P}_x \mathbf{R}_x^{-1} \mathbf{P}_y \quad (19)$$

The minimum MSE in wavelet domain is equal to that of time domain and can be written after further solving (19) and

$$\mathbf{R}_N^W \approx \frac{1}{2} \begin{bmatrix} (1+\rho) & (1-\rho) & \rho^2(1+\rho) & \rho^2(1-\rho) & \cdot & \cdot & \rho^{N-1}(1+\rho) & \rho^{N-1}(1-\rho) \\ (1+\rho) & (\rho-1) & \rho^2(1+\rho) & \rho^2(\rho-1) & \cdot & \cdot & \rho^{N-2}(1+\rho) & \rho^{N-2}(\rho-1) \\ \rho^2(1+\rho) & \rho(\rho-1) & (1+\rho^2) & (\rho^2-1) & \cdot & \cdot & \rho^{N-4}(1+\rho) & \rho^{N-4}(\rho-1) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho^{N-2}(1+\rho^{-1}) & \rho^{N-2}(\rho^{-1}-1) & \cdot & \cdot & \cdot & \cdot & 2 & 0 \end{bmatrix} \quad (25)$$

can be given as [30][31]:

$$J_{\min}^W = \sigma_d^2 - \mathbf{P}_x \mathbf{R}_x^{-1} \mathbf{P}_x \quad (20)$$

3.2. Convergence Analysis of Wavelet Transform Domain LMS Adaptive Filter with AR Process

AR(P) model for univariate time series are Markov process, and P^{th} order AR process to generate the data y_t is as:

$$y_t = \sum_{i=1}^P a_i y_{t-i} + \varepsilon_t; \text{ where } \varepsilon_t \sim N(0, \sigma) \quad (21)$$

where a_k ; $k=1, 2, 3, \dots, P$ are AR coefficients and ε_t a white sequence with $\varepsilon_t \perp \varepsilon_s$ for $t \neq s$. The M order $N \times N$ autocorrelation matrix \mathbf{R}_N can be given according to [23] as:

$$\mathbf{R}_N = \sum_{k=1}^M c_k \mathbf{R}_N^k \quad (22)$$

Here,

$$\mathbf{R}_N^k \approx \begin{bmatrix} 1 & \rho_k & \rho_k^2 & \cdot & \cdot & \rho_k^{N-1} \\ \rho_k & 1 & \rho_k^2 & \cdot & \cdot & \rho_k^{N-2} \\ \rho_k^2 & \rho_k & 1 & \cdot & \cdot & \rho_k^{N-3} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho_k^{N-1} & \rho_k^{N-2} & \cdot & \cdot & \cdot & 1 \end{bmatrix} \quad (23)$$

where, c_1, c_2, \dots, c_M are constant and ρ_k for all k are the M poles of low pass filter with $\rho_k \in [0, 1]$ for all k .

3.2.1. Eigenvalue Spread for First Order AR Process

The first order Markov signals can be obtained by passing a white noise through a one-pole low-pass filter. The N by N autocorrelation matrix of AR (1) or Markov-1 process is obtained from (23) as:

$$\mathbf{R}_N \approx \begin{bmatrix} 1 & \rho & \rho^2 & \cdot & \cdot & \rho^{N-1} \\ \rho & 1 & \rho^2 & \cdot & \cdot & \rho^{N-2} \\ \rho^2 & \rho & 1 & \cdot & \cdot & \rho^{N-3} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho^{N-1} & \rho^{N-2} & \cdot & \cdot & \cdot & 1 \end{bmatrix} \quad (24)$$

The autocorrelation matrix \mathbf{R}_N^W in the wavelet domain for AR (1) process is the autocorrelation matrix of the Markov-1 process of the wavelet transform domain input signal of the same parameter ρ that is: $\mathbf{R}_N^W \approx \mathbf{W} \mathbf{R}_N$; where \mathbf{W} is the $N \times N$ order wavelet transform matrix. Now, putting the value of the wavelet transform matrix from (B. 10) of appendix-B, we have the following form as:

The Toeplitz matrix \mathbf{D}_N^W defined by the first row of (25) as:

$$\mathbf{D}_N^W(0, l) = \mathbf{R}_N^W(0, l) - \frac{l}{N} \mathbf{R}_N^W(0, l) + \frac{l}{N} \mathbf{R}_N^W(0, N-l) \quad (26)$$

Now, substituting the corresponding elements from (25) into (26) we have:

$$\mathbf{D}_N^W(0, l) = (1 + \rho) \left[\frac{\rho^l}{2} - \frac{l\rho^l}{2N} + \frac{l\rho^{N-l}}{2N} \right]; \quad (27)$$

for odd value of l

$$\mathbf{D}_N^W(0, l) = (1 - \rho) \left[\frac{\rho^l}{2} - \frac{l\rho^l}{2N} + \frac{l\rho^{N-l}}{2N} \right]; \quad (28)$$

for even value of l

The power spectrum of the wavelet domain input AR(1) process can be written as[26]:

$$P_x(\omega) = \sum_{l=-\infty}^{\infty} \mathbf{R}_N(l, l) e^{-j\omega l} \quad (29)$$

where, $\mathbf{R}_N(l, l) = \frac{\rho^2}{2} [1 + (-1)^l \rho]$. Now, from (29) we have the power spectrum of the form as:

$$P_x(\omega) = \sum_{l=-(N-1)}^{N-1} \mathbf{R}_N(l) e^{-j\omega l},$$

$$P_x(\omega) = \sum_{l=-(N-1)}^{N-1} \left\{ \frac{\rho^2}{2} (1 + (-1)^l \rho) \right\} e^{-j\omega l} \quad (30)$$

Now, considering $a = 0.5\{1 + (-1)^2 \rho\}$, we have (30) in the form as:

$$P_x(\omega) = \sum_{l=-(N-1)}^{N-1} \rho^l(a) e^{-j\omega l} = \frac{1}{1 - 2\rho a \cos(\omega) + (\rho a)^2}$$

Again considering $\rho_1 = \rho a$, we have above equation in the following form as:

$$P_x(\omega) = \frac{1}{1 - 2\rho_1 \cos(\omega) + (\rho_1)^2} \quad (31)$$

Maximum value of the eigenvalue when $\cos(\omega) = 1$, we have

$$\lambda_{\max} = \frac{1}{1 - 2\rho_1 + \rho_1^2} = \frac{1}{(1 - \rho_1)^2} \quad (32)$$

$$\mathbf{D}_N = \frac{1}{8} \begin{pmatrix} c_1 + c_2 + c_1 \rho_1 + c_2 \rho_2 & c_1 + c_2 - (c_1 \rho_1 + c_2 \rho_2) & 0 & \dots & 0 & 0 \\ c_1 + c_2 + c_1 \rho_1 + c_2 \rho_2 & -(c_1 + c_2) + c_1 \rho_1 + c_2 \rho_2 & & & & \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & & & & \\ 0 & 0 & & & & \end{pmatrix} \quad (42)$$

if $c = c_1 + c_2$, and $s = c_1 \rho_1 + c_2 \rho_2$ then (44) can be written as:

The minimum value when $\cos(\omega) = -1$, and we have

$$\lambda_{\min} = \frac{1}{1 + 2\rho_1 + \rho_1^2} = \frac{1}{(1 + \rho_1)^2} \quad (33)$$

The Eigenvalue spread of the following form

$$= \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{(1 + \rho_1)^2}{(1 - \rho_1)^2} = \frac{\left[1 + \frac{\rho}{2} \{1 + (-1)^l \rho\}\right]^2}{\left[1 - \frac{\rho}{2} \{1 + (-1)^l \rho\}\right]^2} = \left(\frac{2 + \rho - \rho^2}{2 - \rho + \rho^2}\right)^2 \quad (34)$$

3.2.2. Eigenvalue Spread for Second Order AR Process

The characteristic equation for second order AR (2) process is given as[32]:

$$1 + a_1 z^{-1} + a_2 z^{-2} = 0 \quad (35)$$

The pole of the filters ρ_1 , and ρ_2 is defined by the characteristics equation with the following form of transfer function as:

$$H(z) = \frac{1}{(1 - \rho_1 z^{-1})(1 - \rho_2 z^{-2})} \quad (36)$$

The $N \times N$ correlation matrix of the AR(2) process can be defined as:

$$\mathbf{R}_N = c_1 \mathbf{R}_N^1 + c_2 \mathbf{R}_N^2 \quad (37)$$

Where,

$$\mathbf{R}_N^i \approx \begin{pmatrix} 1 & \rho_i & \rho_i^2 & \dots & \dots & \rho_i^{N-1} \\ \rho_i & 1 & \rho_i^2 & \dots & \dots & \rho_i^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \rho_i^{N-1} & \rho_i^{N-2} & \dots & \dots & \dots & 1 \end{pmatrix} \quad (38)$$

$$c_1 = \frac{\rho_1(1 - \rho_2^2)}{(\rho_1 - \rho_2)(1 + \rho_1 \rho_2)}, \quad c_2 = -\frac{\rho_2(1 - \rho_1^2)}{(\rho_1 - \rho_2)(1 + \rho_1 \rho_2)} \quad (39)$$

The following matrix is defined as[25]:

$$\mathbf{D}_N = c_1 \mathbf{D}_N^1 + c_2 \mathbf{D}_N^2 \quad (40)$$

Where,

$$\mathbf{D}_N^i \approx \mathbf{W}_N^H \left(\text{diag}(\mathbf{W}_N \mathbf{R}_N^i \mathbf{W}_N^H) \right) \mathbf{W}_N; i=1,2 \quad (41)$$

and \mathbf{W}_N is the N^{th} wavelet transform matrix defined in (B. 10) of appendix-B. The numerical value of the matrix \mathbf{D}_N of (40) has been calculated in appendix-C and is given as:

$$\mathbf{D}_N^{DWT} = \begin{pmatrix} c+s & c-s & 0 & . & 0 & 0 \\ c+s & c-s & 0 & . & 0 & 0 \\ . & . & & & & \\ . & . & & & & \\ 0 & 0 & 0 & 0 & c+s & c-s \\ 0 & 0 & 0 & 0 & c+s & c-s \end{pmatrix} \quad (43)$$

Now, we compute the simple matrix;

$$\mathbf{X}_N^{DWT} = \mathbf{R}_N^{-1} \mathbf{D}_N^{DWT} \quad (44)$$

The wavelet transform-domain autocorrelation matrix \mathbf{R}_N^{DWT} for AR(2) process can be written in the following form as:

$$\mathbf{R}_N^{DWT} = c_1 \mathbf{R}_N^{1,1} + c_2 \mathbf{R}_N^{2,2} \quad (45)$$

$$\mathbf{R}_N^{DWT} = \frac{c_1}{2} \begin{pmatrix} \rho_a & \rho_b & \rho_1^2 \rho_a & \rho_1^2 \rho_b & . & \rho_1^{N-1} \rho_a & \rho_1^{N-1} \rho_b \\ \rho_a & -\rho_b & \rho_1^2 \rho_a & -\rho_1^2 \rho_b & . & \rho_1^{N-2} \rho_a & (-1)^{N-1} \rho_1^{N-2} \rho_b \\ \rho_1^2 \rho_a & -\rho_1 \rho_b & (1+\rho_1^2) & -(1-\rho_1^2) & . & \rho_1^{N-4} \rho_a & (-1)^{N-3} \rho_1^{N-1} \rho_b \\ . & . & . & . & . & . & . \\ \rho_1^{N-2} \rho_a & -\rho_1^{N-2} \rho_b & . & . & . & \rho_a & -\rho_b \end{pmatrix} + \frac{c_2}{2} \begin{pmatrix} \rho_x & \rho_y & \rho_2^2 \rho_x & \rho_2^2 \rho_y & . & \rho_2^{N-1} \rho_x & \rho_2^{N-1} \rho_y \\ \rho_x & -\rho_y & \rho_2^2 \rho_x & -\rho_2^2 \rho_y & . & \rho_2^{N-2} \rho_x & (-1)^{N-1} \rho_2^{N-2} \rho_y \\ \rho_2^2 \rho_x & -\rho_2 \rho_y & (1+\rho_2^2) & -(1-\rho_2^2) & . & \rho_2^{N-4} \rho_x & (-1)^{N-3} \rho_2^{N-1} \rho_y \\ . & . & . & . & . & . & . \\ \rho_2^{N-2} \rho_x & -\rho_2^{N-2} \rho_y & . & . & . & \rho_x & -\rho_y \end{pmatrix} \quad (46)$$

where, $\rho_a=1+\rho_1$, $\rho_b=1-\rho_1$, and $\rho_x=1+\rho_2$, $\rho_y=1-\rho_2$. The power spectrum of the input with AR(2) process can be written as:

$$\mathbf{P}_{xx}(\omega) = \sum_{l=-\infty}^{\infty} \mathbf{R}_N^{DWT}(l, l) e^{-j\omega l} \quad (47)$$

Now, we can find $\mathbf{R}_N^{DWT}(l, l)$ from (44) and is given as:

$$\mathbf{R}_N^{DWT}(l, l) = \frac{c_1 \rho_1^2}{2} (1 + (-1)^2 \rho_1) + \frac{c_2 \rho_2^2}{2} (1 + (-1)^2 \rho_2) = \frac{c_1 \rho_1^l + c_2 \rho_2^l}{2} + (-1)^l \frac{c_1 \rho_1^{l+1} + c_2 \rho_2^{l+1}}{2} \quad (48)$$

If $\tilde{\rho}^l = c_1 \rho_1^l + c_2 \rho_2^l$; $l=0, 1, \dots, N-1$, the (50) can be written as:

$$\mathbf{R}_N^{DWT}(l, l) = \frac{\tilde{\rho}^l}{2} \{1 + (-1)^l \tilde{\rho}\} = \tilde{\rho}^l a; \quad (49)$$

where, $a = \frac{1}{2} \{1 + (-1)^l \tilde{\rho}\}$. Now, (50) can be written as:

$$P_{xx}(\omega) = \sum_{l=-(N-1)}^{N-1} \tilde{\rho}^l a e^{-j\omega l} = \frac{1}{1 - 2\tilde{\rho}a \cos(\omega) + (\tilde{\rho}a)^2} \quad (50)$$

Eigen value maximum when $\cos(\omega)=-1$ and minimum when $\cos(\omega)=1$. The Eigenvalue spread is given as:

$$\text{Eigenvalue spread} = \frac{\lambda_{mas}}{\lambda_{min}} = \frac{(1 - \tilde{\rho}a)^2}{(1 + \tilde{\rho}a)^2}, \lambda_{mas} = \left(\frac{2 - \tilde{\rho}(1 - \rho^2)}{2 + \tilde{\rho}(1 + \rho^2)} \right)^2 \quad (51)$$

where, $\tilde{\rho} = c_1 \rho_1 + c_2 \rho_2$ for $l = 1$.

4. Structure of Wavelet Transform LMS Algorithm

Signal in most of engineering applications are non-stationary, and Fourier transform encountered difficulty to analyze the time-frequency property[33][34]. The wiener filter is assumed to be stationary but the stationary

assumption of the noisy signal cannot completely satisfied in many applications[35]. The SWT domain signal vector $\mathbf{v}(k)$ in the matrix form can be expressed as:

$$\mathbf{v}(k) = \Psi \mathbf{x}(k) \quad (52)$$

where, $\mathbf{x}(k) = [\dots, x(k+1), x(k), x(k-1), \dots]^T$, Ψ is the wavelet transform matrix whose row vector is Ψ_{mn} as:

$$\Psi_{mn} = [\dots, \Psi_{mn}(k-1), \Psi_{mn}(k), \Psi_{mn}(k+1), \dots]^T \quad (53)$$

The SWT-transformed signal vector $\mathbf{v}(k)$ can be expressed in the following form as:

$$\mathbf{v}(n) = [\mathbf{d}^{J-1}, \mathbf{d}^{J-2}, \dots, \mathbf{d}^L, \mathbf{s}^L] \quad (54)$$

The smooth and details components at level j can be expressed as:

$$\mathbf{s}^{j-1} = \mathbf{H}^{[j-1]} \mathbf{s}^j \text{ and } \mathbf{d}^{j-1} = \mathbf{G}^{[j-1]} \mathbf{s}^j \quad (55)$$

The low pas filter \mathbf{H} and high pass filter \mathbf{G} denoted by the sequence $\{h_n\}_{n \in \mathbb{Z}}$ and $\{g_n\}_{n \in \mathbb{Z}}$ with $g_n = (-1)^n h_{1-n}$; where h_n 's satisfy the orthogonality condition $h_n h_{n+2j} = 0$; $j=0, 1, 2, \dots$, SWT on N samples of signal $\mathbf{x}(n)$ yields[35]:

$$\mathbf{v}(n) = \{\mathbf{d}_k^{J-1}(0), \dots, \mathbf{d}_k^{J-1}(N-1), \dots, \mathbf{d}_k^{J-2}(0), \dots, \mathbf{d}_k^{J-2}(N-1), \dots, \mathbf{d}_k^L(0), \dots, \mathbf{d}_k^L(N-1), \dots, \mathbf{s}_k^L(0), \dots, \mathbf{s}_k^L(N-1)\} \quad (56)$$

where, $\mathbf{x}(n) = \{x_0[0], \dots, x_0[N-1], \dots, x_k[0], \dots, x_k[N-1], \dots, x_{k-1}[0], \dots, x_{k-1}[N-1]\}$. The SWT provides the same length of both smooth and details components and this property is very useful in time-varying application of wiener filter in block by block basis. The SWT-domain based LMS algorithm provides the system output signal $\mathbf{y}(k)$ that can be represented in the following form:

$$\mathbf{y}(n) = \mathbf{W}^J(n) \mathbf{v}^J(n) \quad (57)$$

where, $J=1, 2, \dots, L$; L is the level of decomposition. The weighting vector for each level can be denoted as:

$$\mathbf{W}^J(k) = [\mathbf{W}^1(0), \dots, \mathbf{W}^1(N-1), \dots, \mathbf{W}^2(0), \dots, \mathbf{W}^2(N-1), \dots, \mathbf{W}^L(0), \dots, \mathbf{W}^L(N-1)] \quad (58)$$

The SWT domain ($\mathbf{v}(n)$) of the input signal $\mathbf{x}(n)$ as in (57) having the following form as:

$$\mathbf{v}(n) = \{\mathbf{d}^1(0), \dots, \mathbf{d}^1(N-1), \dots, \mathbf{d}^2(0), \dots, \mathbf{d}^2(N-1), \dots, \mathbf{d}^L(0), \dots, \mathbf{d}^L(N-1), \dots, \mathbf{s}^L(0), \dots, \mathbf{s}^L(N-1)\} \quad (59)$$

The SWT based LMS algorithm according to (5) can be represented as:

$$\mathbf{W}^J(n) = \mathbf{W}^J(n-1) + 2\mu \mathbf{e}^J(n) \hat{\mathbf{R}}_J^{-1}(n) \mathbf{v}^J(n) \quad (60)$$

where, $\mathbf{e}^L(n) = \mathbf{v}_d^L(n) - \mathbf{W}^L(n-1) \mathbf{v}^L(n)$

$\mathbf{v}_d^L(n)$ = L level SWT domain signal of the desired signal $\mathbf{d}(n)$, in vector form $\mathbf{v}_d^L(n) = [\mathbf{v}_d^L(0), \mathbf{v}_d^L(1), \dots, \mathbf{v}_d^L(N-1)]$.

The convergence factor and misadjustment for each level of the SWT domain signal is governed according to the algorithm as discussed in section 2 of this paper. The MSE cost function of each level can be written in the following form as:

$$J(m) = E\{\mathbf{e}^L(m) \mathbf{e}^L(m)\} \quad (61)$$

5. Result

The misadjustment of the adaptive filter measures the amount of noise in the filter output and increases with the step-size parameter. Lower amount of misadjustment indicates the better denoising performance of the filtering algorithm. Let us consider a multi-tone signal of the following form as:

$$\mathbf{y}(n) = \sin(0.1\pi n) + \cos(0.1\pi n) + \sin(0.2\pi n) \quad (62)$$

The toeplitz matrix \mathbf{R}_y of the autocorrelation matrix of the multi-tone signal $\mathbf{y}(n)$ in (62) before any transformation is shown in Fig. 1(a). The toeplitz matrix after application of DCT and DWT in the multi-tone signal $\mathbf{y}(n)$ is shown in Fig. 1(b) and Fig. 1(c) respectively. The misadjustment curve in Fig. 1(d) shows that the misadjustment increases of the three cases as input toeplitz matrix, DCT based input toeplitz matrix, and DWT based input autocorrelation matrix. The DWT-based curve shows the better performance because it provides less misadjustment value for all values of μ than other two.

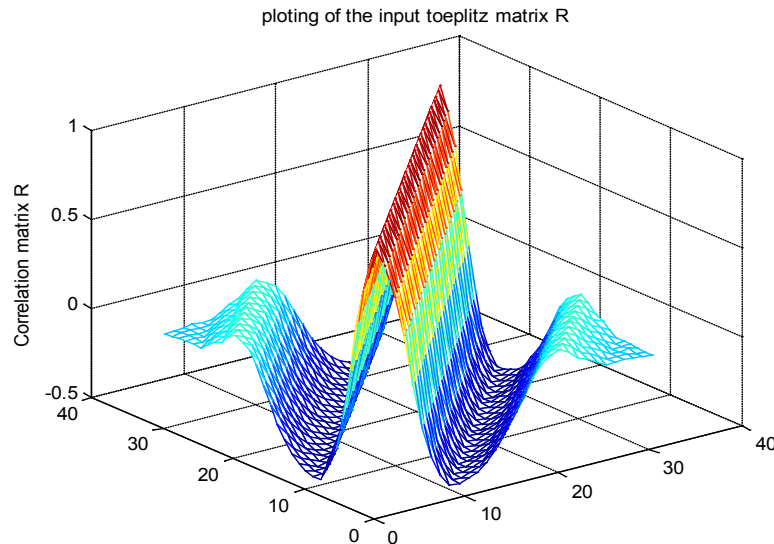


Figure 1(a). Autocorrelation matrix of the input multi-tone signal without any transformation

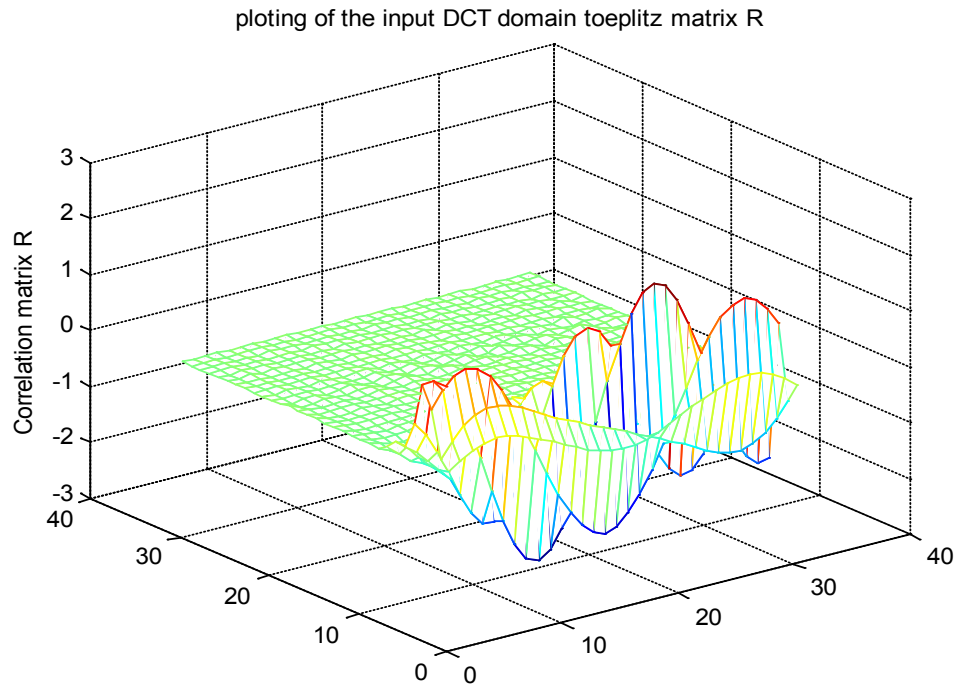


Figure 1(b). Autocorrelation matrix of the DCT transformed based multi-tone signal

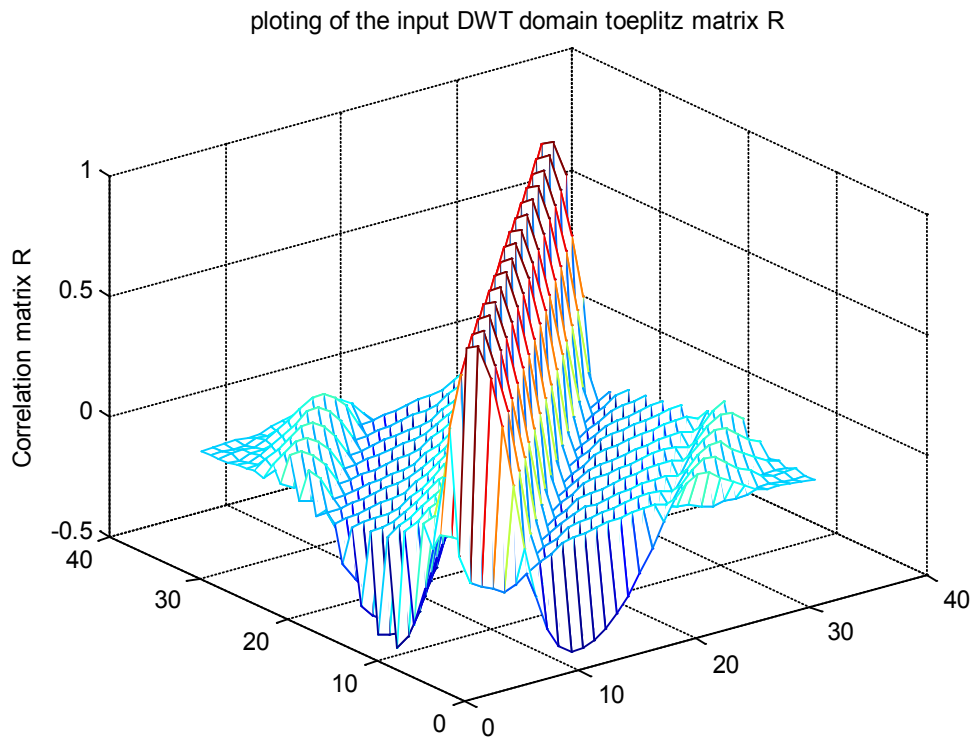


Figure 1(c). DWT based autocorrelation matrix of the input multi-tone signal

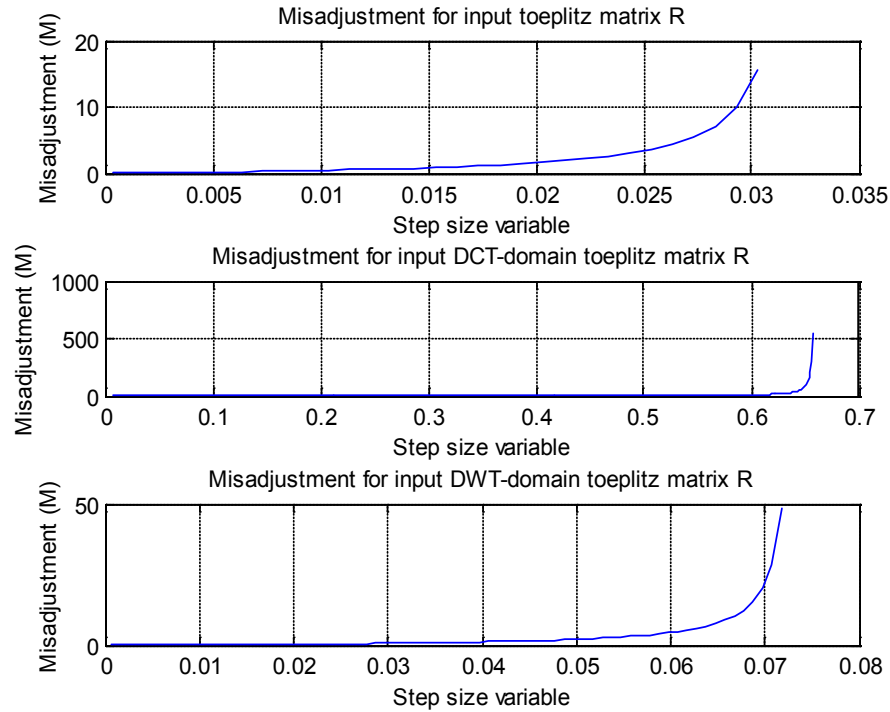


Figure 1(d). Misadjustment result of input autocorrelation matrix, DCT and DWT based input autocorrelation matrix

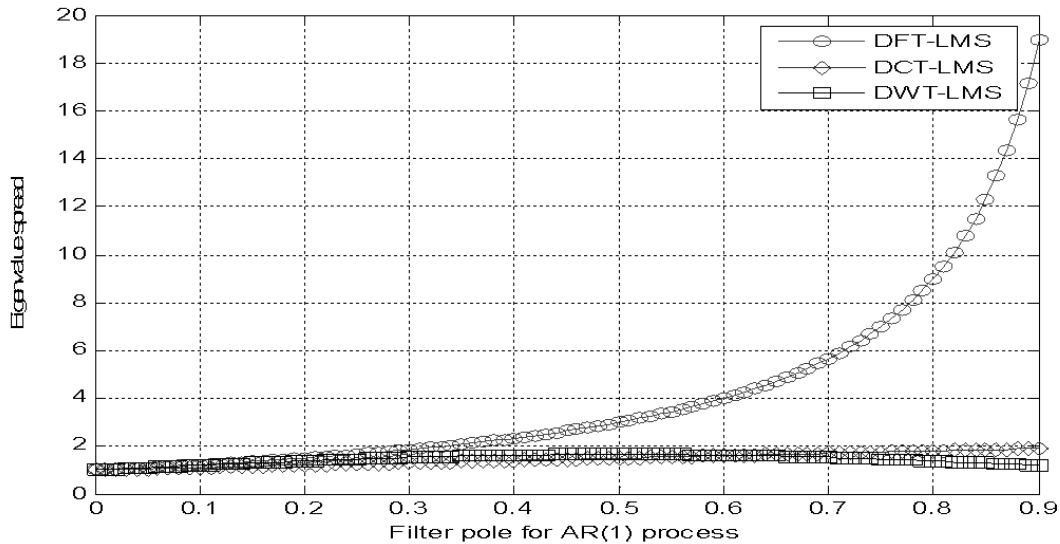


Figure 2. Eigenvalue spread of DFT-LMS, DCT-LMS, and DWT-LMS for AR (1) input process

The eigenvalue spread of DFT-LMS and DCT-LMS for AR(1) process is shown in Fig. 2, details of the work has been discussed in the work[23][36]. The eigenvalue spread of DWT-LMS analyzed in this paper with the same procedure is also shown in Fig. 2, the result shows that DWT-LMS achieves the smaller eigenvalue spread than DFT-LMS and DCT-LMS at some upper region of the area of (ρ). Hence, DWT-LMS gives the best convergence performance for the AR(1) input process. The eigenvalue spread for DFT-LMS and DCT-LMS for AR(2) input process has been analyzed in the work[25] and the result is also shown in Fig. 3(a) and Fig. 3(b). In the work[25], DCT-LMS achieves the smaller eigenvalue spread than other and provides the best convergence performance for the AR(2) process.

The autocorrelation matrix \mathbf{R}_N for LMS with AR(2) process without any transformation is calculated in (37), and the same calculation for the multi-tone signal is calculated in this paper with ρ_1 and ρ_2 , and the 3D-plot of the matrix is shown in Fig. 4(a).

The matrix \mathbf{D}_N^1 , \mathbf{D}_N^2 , and \mathbf{D}_N in (41) and (40) after applying DWT in the input AR(2) process is shown in Fig. 4(b), Fig. 4(c), and Fig. 4(d) respectively. Now, we compute the matrix $\mathbf{X}_N^{DWT} = \mathbf{R}_N^{-1} \mathbf{D}_N^{DWT}$; where \mathbf{R}_N is given in (37) and \mathbf{D}_N^{DWT} is

given in (43) and plotted in Fig. 4(d). The distribution of \mathbf{X}_N^{DWT} is shown in Fig. 5(a) for the sample value of $N=30$, the result shows that most elements of \mathbf{X}_N^{DWT} is very close to zero and the only the diagonal elements having values close to 0.6. The distribution of \mathbf{X}_N^{DWT} is very important factor to compute the eigenvalue spread of the autocorrelation matrix \mathbf{V}_N ; that obtained after DWT.

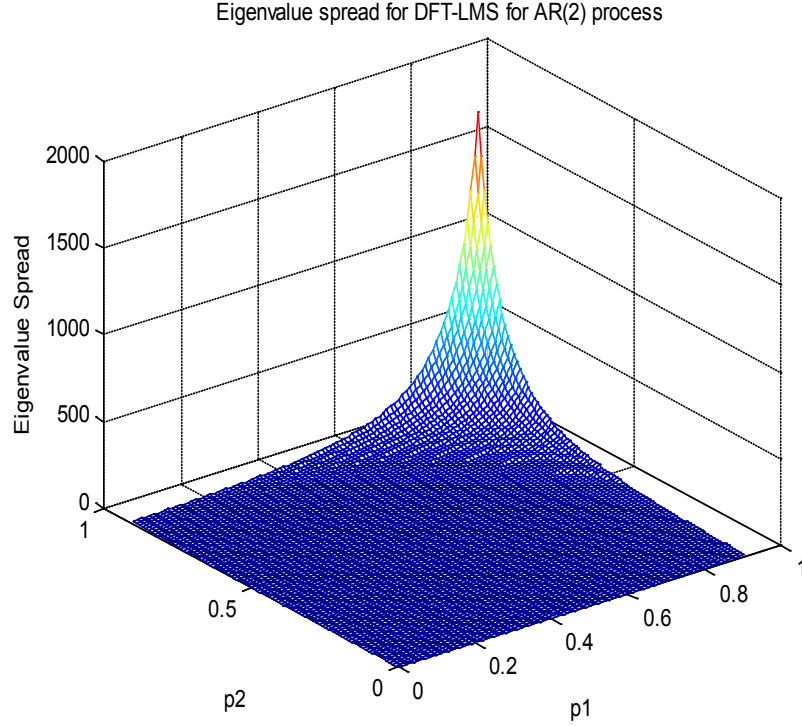


Figure 3(a). Eigenvalue spread of DFT-LMS for second-order lowpass AR process

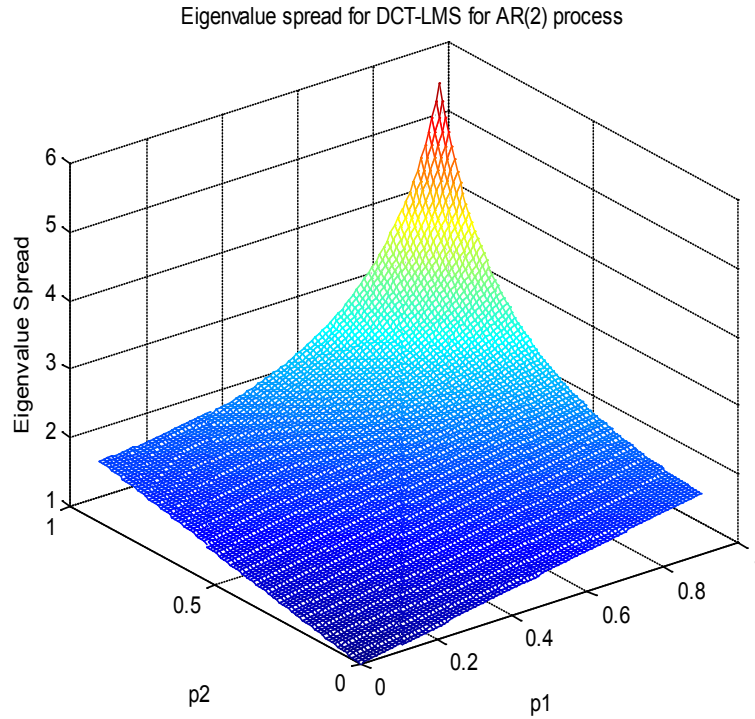


Figure 3(b). Eigenvalue spread of DCT-LMS for second-order lowpass AR process

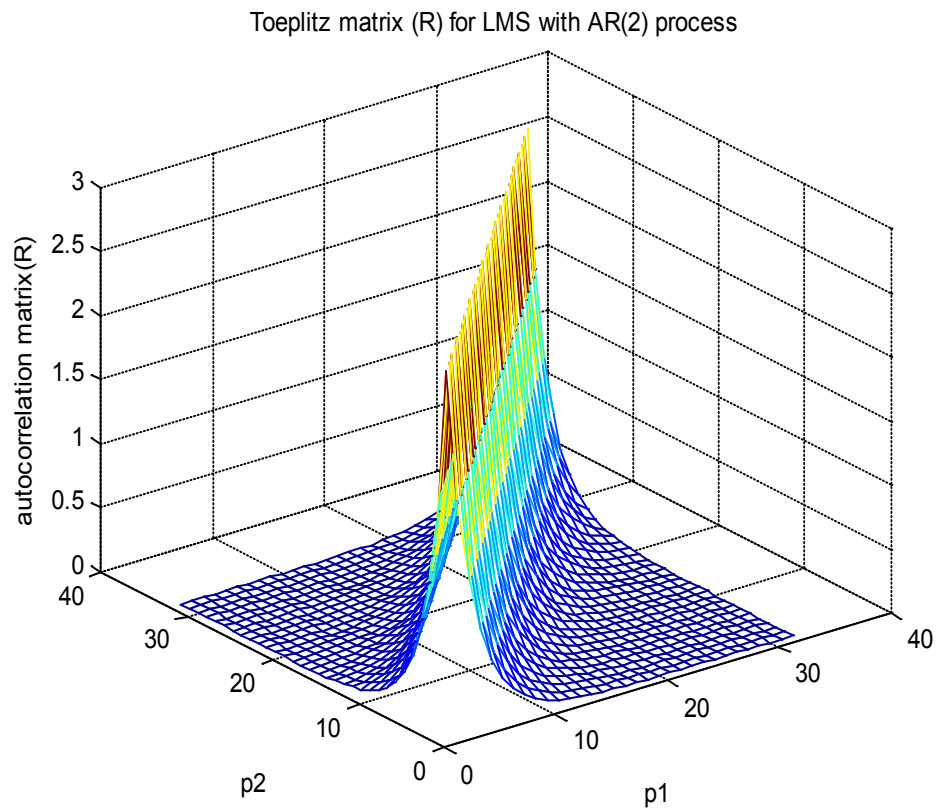


Figure 4(a). Autocorrelation matrix \mathbf{R}_N before any transform for AR(2) input process

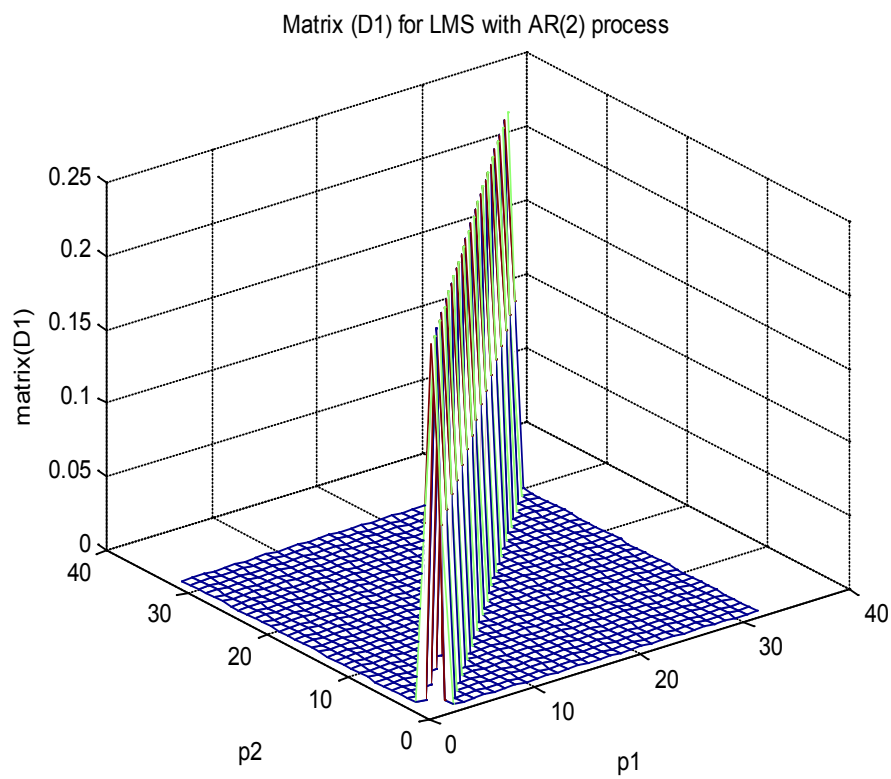


Figure 4(b). Matrix \mathbf{D}_N^1 after DWT in the AR(2) process

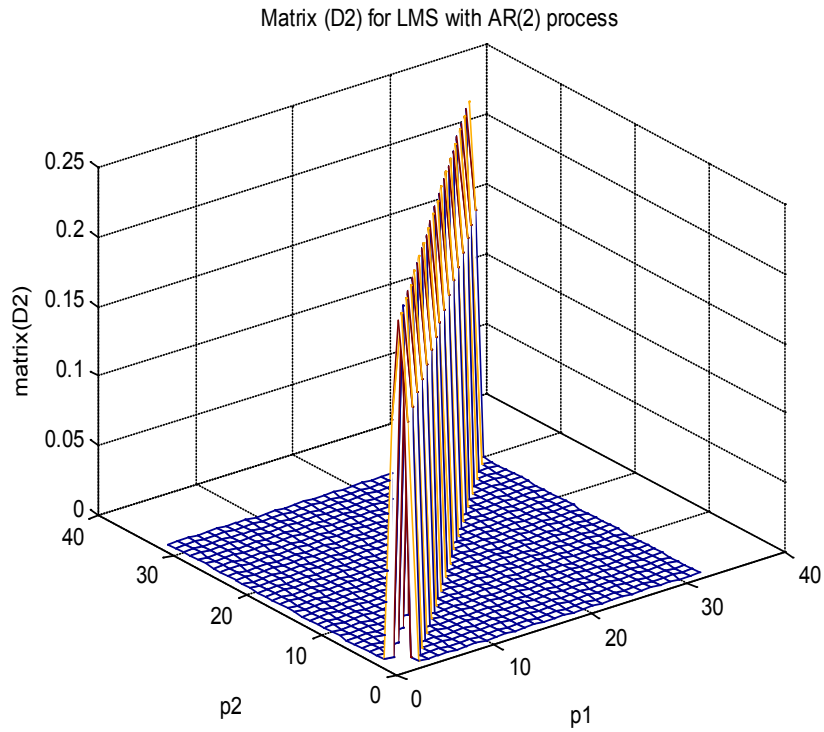


Figure 4(c). Matrix \mathbf{D}_N^2 after DWT in the AR (2) process

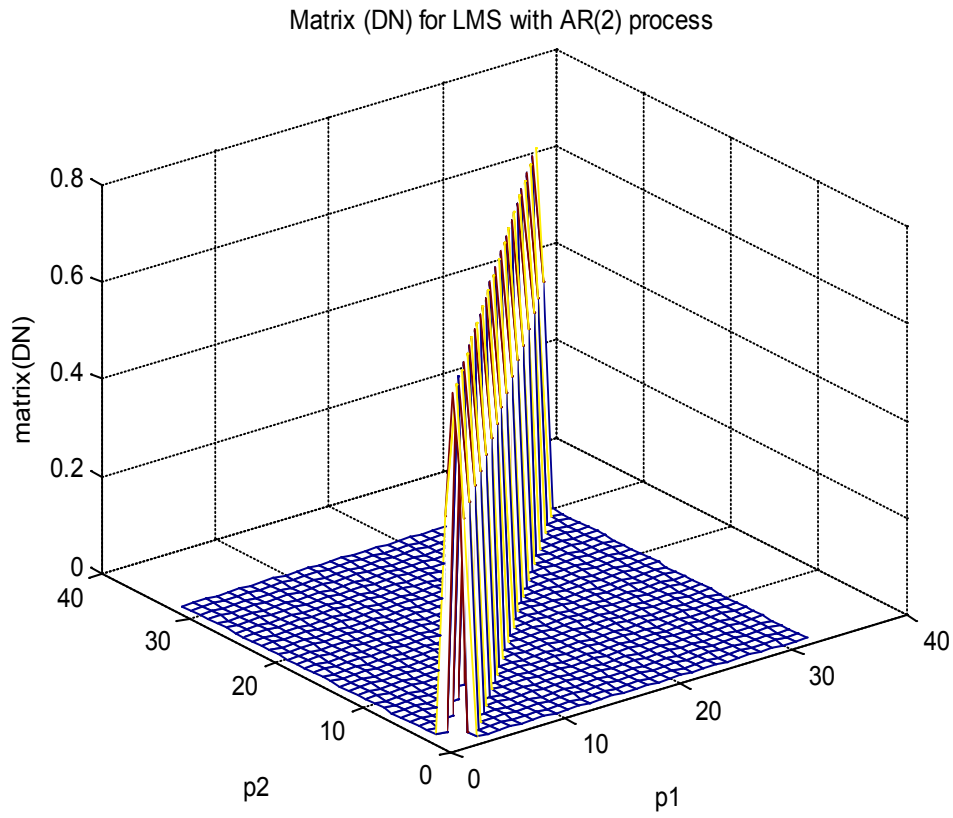


Figure 4(d). Matrix \mathbf{D}_N after DWT in the AR(2) process

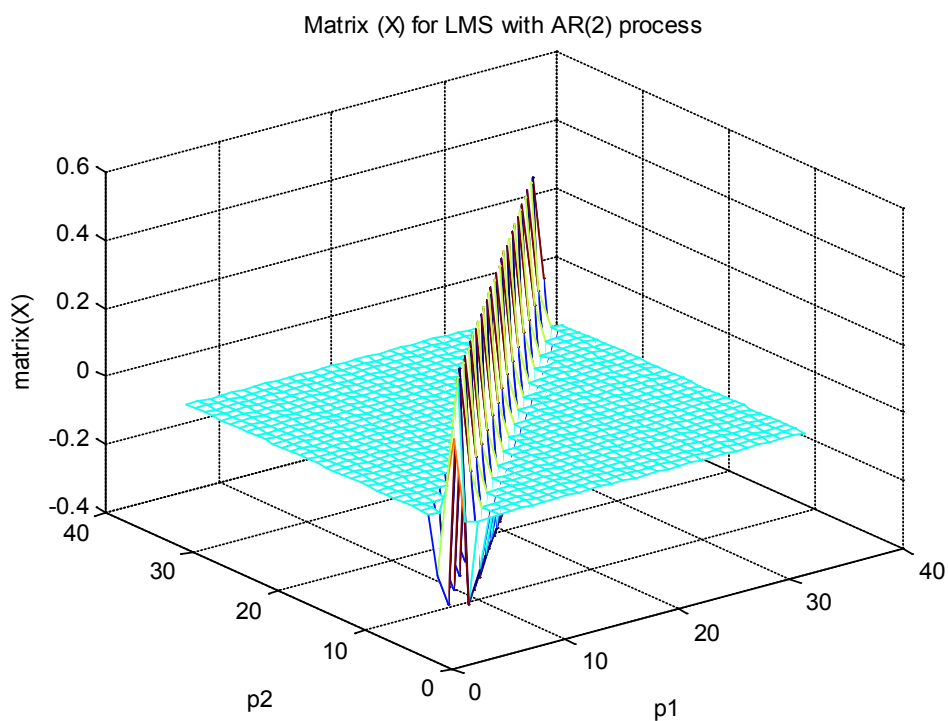


Figure 5(a). 3D plot of the matrix \mathbf{X}_N^{DWT} involved in DWT-LMS eigenvalue spread calculation

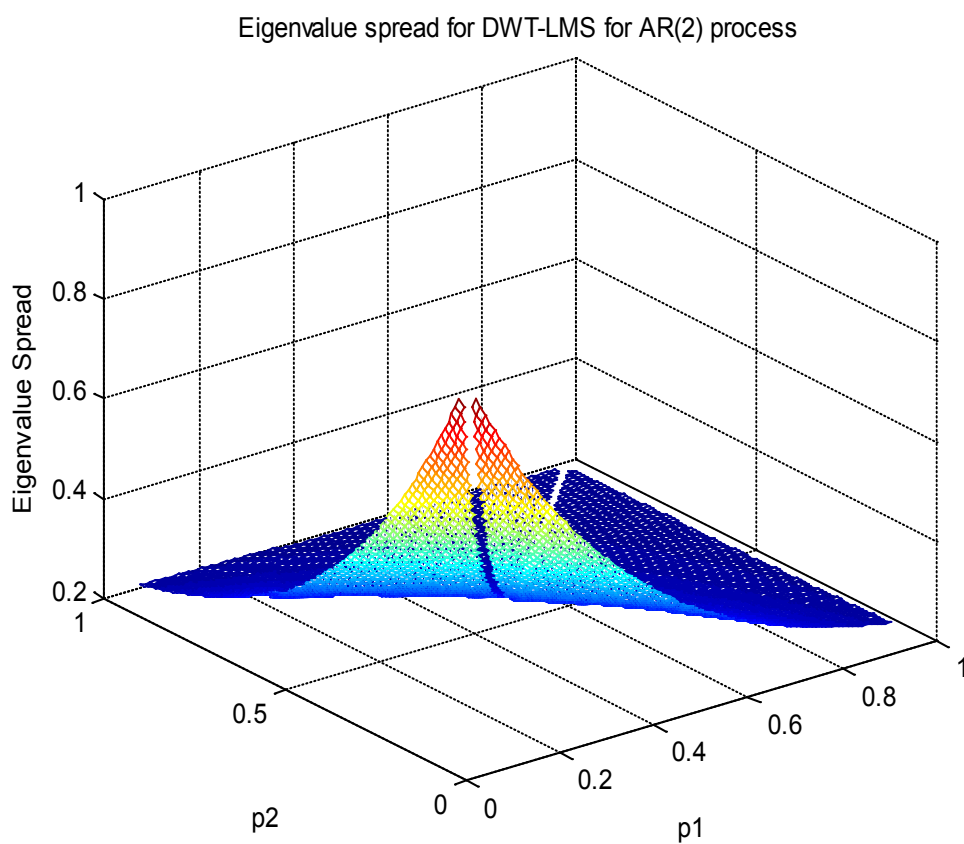


Figure 5(b). Eigenvalue spread of DWT-LMS for second-order lowpass AR process

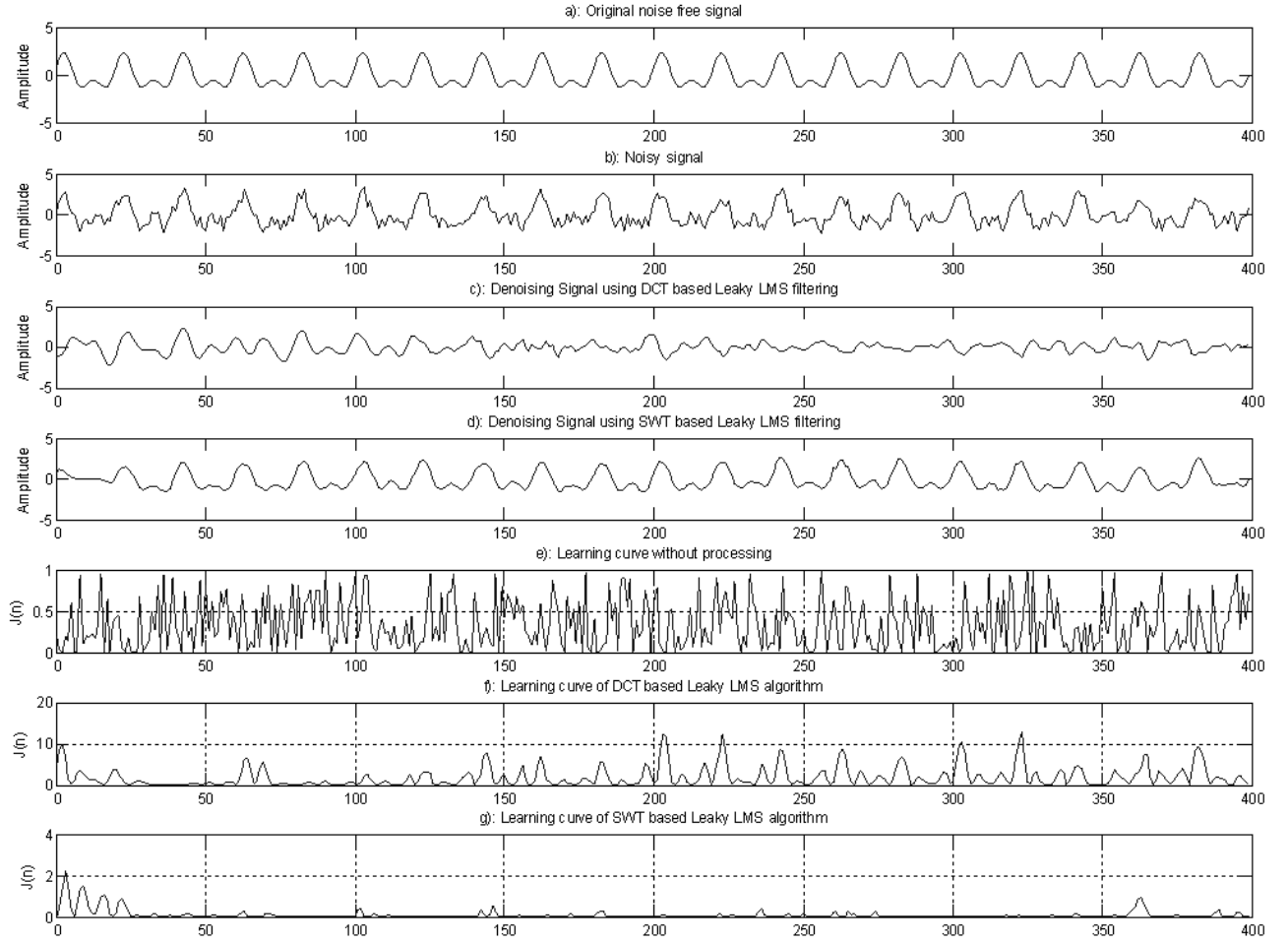


Figure 6. Denoising performance of DCT-Leaky LMS and SWT-Leaky LMS for AR(2) process

The convergence relation for DWT-LMS can be obtained from (51) for real roots of $|\rho_1|, |\rho_2| < 1$ with second-order input case. The eigenvalue spread can be plot for DWT-LMS with given real or complex values of ρ_1 and ρ_2 . The 3D plot of the eigenvalue spread of DWT-LMS is in Fig. 5(b), the result shows that the value converges to zero when both roots deviate from the origin. The eigenvalue spread of DFT-LMS, DCT-LMS, and DWT-LMS for second-order AR process as in Fig. 3(a), 3(b), and 5(b), and the result shows that DWT-LMS provides the best convergence performance among them. Hence, the lower value of the eigenvalue spread of DWT-LMS provides the lowest noise term in the output of the denoise signal. To support the above analytical result, the signal in (62) is passed through a 2nd order AR filter as given below:

$$y(n) = y(n) + 0.5y(n-1) + 0.33y(n-2) + v(n) \quad (63)$$

where, $v(n)$ is white gaussian noise with mean zero and variance σ_2 ; $\mathbf{a} = [1 \ 0.5 \ 0.33]$ is the AR coefficients. The original multi-tone and output of the AR filter as noisy signal is shown in Fig. 6(a) and Fig. 6(b). The DCT algorithm as in [40] is used in Leaky LMS-Newton adaptive filter [36] which is known as DCT-Leaky LMS-Newton algorithm which is actually same as LMS with better denoising performances. In this paper, the AR(2) output signal is

passed through DCT-Leaky LMS and SWT-Leaky LMS with considering $\mu=0.01$ and $\gamma=0.1$, the value of μ is considered depending on the stability condition, details of the mathematical analysis is discussed in section II. The denoising signal of DCT-Leaky LMS and SWT-Leaky LMS is shown in Fig. 6(c) and Fig. 6(d), the result shows that SWT-Leaky LMS provides better denoising performance than DCT-Leaky LMS. The error of the denoised signal from the actual is given in the below Table-1. From the table it can be concluded that SWT-Leaky LMS provides more denoising than DCT-Leaky LMS for each sample value.

Table 1. Denoising Error of DCT-Leaky LMS and SWT-Leaky LMS

Number of sample	Error for DCT-Leaky LMS	Error for SWT-Leaky LMS
0	2.1893	0.1746
1	2.9146	0.6288
2	3.1273	1.1566
3	2.5611	1.5034
4	1.2481	1.2635
5	-0.0587	0.6573
6	-1.0500	0.0807
7	-1.7193	-0.7298
8	-1.8562	-1.1723
9	-1.6414	-1.2298

The learnign curve $J(n)$ or MSE of both DCT-Leaky LMS and SWT-Leaky LMS with the same parameter through the same noisy channel is shown in Fig. 6(f) and Fig. (g). From the result it is shown that the maximum value of the learning curve of SWT-Leaky LMS is close to 2 at initial sampling point an it converge to zero after 10 sample points, in the other hand DCT-Leaky LMS provides high value of $J(n)$ as upto 10 and it repeated to the maximum after 30 sample points. So, SWT-Leaky LMS provides better convergence performance than DCT-Leaky LMS algorithm.

6. Conclusions

The stability and convergence performance of DFT-LMS, DCT-LMS, DHT-LMS, and DST-LMS has been analyzed in details in work[25], and the result shows that DCT-LMS and DHT provides better convergence performance than other. In this work, we have designed DWT-LMS Newton adaptive or SWT-LMS Newton adaptive, or in simply DWT-LMS or

SWT-LMS adaptive filtering algorithm and the performances is compared with the DCT-LMS adaptive filtering algorithm for first-order and second-order AR process as developed in the early work[25]. From the comparison result it is shown that DWT-LMS provides better misadjustment, convergence, and denoising performance than that of DCT-LMS adaptive filtering algorithm for both AR(1) and AR(2) process. For example, if $\rho=0.8$ then eigenvalue spread of DCT-LMS is 1.8 and for DWT-LMS is 1.37 for AR(1) process. The eigenvalue spread of DCT-LMS is 2.63 and for DWT-LMS is 0.25 for $\rho_1=0.59$ and $\rho_2=0.79$ respectively with AR(2) process. From talble-1, it is also concluded that SWT-LMS provides less denoising error in each sample value than DCT-LMS adaptive algorithm. Hence, the DWT-LMS provides better misadjustment, convergence or eigenvalue spread, and denoising performance than others as presented in work[25]. In future the SWT-LMS algorithm can be applied for higher order AR process and other type of input process like MA and ARMA process.

APPENDIX A: MSE of Wavelet Packet Domain Signal

$$J(\mathbf{W}) = E\{e^2(n)\} = E\{[\mathbf{d}(n) - \mathbf{r}(n)]^2\} = E\{[\mathbf{d}(n) - \mathbf{W}^T(n)\mathbf{y}(n)]^2\}$$

The error function in wavelet domain can be written by replacing the input signal in wavelet transform domain as:

$$J(\mathbf{W}) = E\{[\mathbf{d}(n) - \mathbf{W}^T(n)\mathbf{V}_p\mathbf{x}(n)]^2\} \quad (\text{A.1})$$

$$J(\mathbf{W}) = E\{[\mathbf{d}(n) - \mathbf{W}^T(n)\mathbf{V}_p\mathbf{x}(n)] \times [\mathbf{d}(n) - \mathbf{W}^T(n)\mathbf{V}_p\mathbf{x}(n)]\}$$

$$J(\mathbf{W}) = E\{\mathbf{d}^2(n)\} - 2\mathbf{W}^T(n)\mathbf{V}_p\mathbf{P}_x + \mathbf{W}^T(n)\mathbf{V}_pE\{\mathbf{d}(n)\mathbf{x}(n)\} + \mathbf{W}^T(n)\mathbf{V}_pE\{\mathbf{x}(n)\mathbf{x}^T(n)\}\mathbf{W}(n)\mathbf{V}_p$$

Now, $E\{\mathbf{d}^2(n)\} = \sigma_d^2$ is the variance of the desired signal $\mathbf{d}(n)$.

$\mathbf{P}_x = E\{\mathbf{d}(n)\mathbf{x}(n)\}$; the cross-correlation between $\mathbf{d}(n)$ and $\mathbf{x}(n)$

$\mathbf{P}_y = \mathbf{V}_p\mathbf{P}_x$; Cross-correlation vector in wavelet domain

$\mathbf{R}_x = E\{\mathbf{x}(n)\mathbf{x}^T(n)\}$; autocorrelation matrix of $\mathbf{x}(n)$

$\mathbf{R}_y = \mathbf{V}_p\mathbf{R}_x\mathbf{V}_p$; autocorrelation matrix in wavelet domain

$\mathbf{W}(n) = \mathbf{R}_y^{-1}\mathbf{P}_y$; filtering weighting vector in wavelet domain

Substituting those parameters we have the solution as:

$$J(\mathbf{W}) = \sigma_d^2 - 2\mathbf{W}^T(n)\mathbf{V}_p\mathbf{P}_x + \mathbf{W}^T(n)\mathbf{V}_p\mathbf{R}_x\mathbf{W}\mathbf{V}_p \quad (\text{A.2})$$

APPENDIX B: Algorithm in Generating a Wavelet Matrix

Let us assume \mathbf{W} be an N by N matrix of the following form:

$$\mathbf{W} = \mathbf{W}\mathbf{I} \quad (\text{B.1})$$

Where \mathbf{I} is also N by N unity matrix, the alternative form of (B.1) can be written as:

$$\mathbf{W} = \mathbf{W}[\delta_1 \ \delta_2 \ \dots \ \delta_N] = [\mathbf{W}\delta_1 \ \mathbf{W}\delta_2 \ \dots \ \mathbf{W}\delta_N] \quad (\text{B.2})$$

Where δ_i for $i=1, 2, \dots, N$ is the i^{th} column of \mathbf{I} . Equation (B.2) implies that the wavelet transform on δ_i ; $i=1, 2, \dots, N$ provides N column vectors and the vectors form a wavelet matrix. The 1 scale wavelet transform of a column vector \mathbf{x} of length N can be expressed as[31][38]:

$$\mathbf{y}_1 = \mathbf{W}_1^{(N)} \mathbf{x} \quad (\text{B.3})$$

The two scale wavelet coefficients can be represented as:

$$\mathbf{y}_2 = \mathbf{W}'_2^{(N)} \mathbf{x} \quad (\text{B.4})$$

where $\mathbf{W}'_2^{(N)}$ can be represented of the following form as:

$$\mathbf{W}'_2^{(N)} = \begin{bmatrix} \mathbf{W}_1^{(N/2)} & 0 \\ 0 & \mathbf{I}_{N/2} \end{bmatrix} \quad (\text{B.5})$$

where $\mathbf{W}_1^{(N)}$ and $\mathbf{W}_1^{(N/2)}$ corresponds to 1 scale wavelet transform matrix of length N and $N/2$. Now, the 2 scale wavelet transform matrix for the input vector \mathbf{x} can be represented by integrating (B.4) and (B.5) as:

$$\mathbf{W}_2^{(N)} = \mathbf{W}_2^{*(N)} \mathbf{W}_1^{(N)} = \begin{bmatrix} \mathbf{W}_1^{(N)} & 0 \\ 0 & \mathbf{I}_{N/2} \end{bmatrix} \mathbf{W}_1^{(N)} \quad (\text{B.6})$$

A k scale wavelet transform matrix can be generated by the procedure in (B.6) and can be represented in the following form as:

$$\mathbf{W}_k^{(N)} = \begin{bmatrix} \mathbf{W}_1^{(N/2^{k-1})} & 0 \\ 0 & \mathbf{I}_{N-N/2^{k-1}} \end{bmatrix} \begin{bmatrix} \mathbf{W}_1^{(N/2^{k-2})} & 0 \\ 0 & \mathbf{I}_{N-N/2^{k-2}} \end{bmatrix} \begin{bmatrix} \mathbf{W}_1^{(N)} & 0 \\ 0 & \mathbf{I}_{N/2} \end{bmatrix} \mathbf{W}_1^{(N)} \quad (\text{B.7})$$

Now, assume the high pass and low pass filter response of the form, $\mathbf{h}=[h_0, h_1, \dots, h_{m-1}]$, and $\mathbf{g}=[g_0, g_1, \dots, g_{m-1}]$, then we have the wavelet two functions of the form as:

$$\begin{aligned} \psi &= [0 \ h_1 \ h_0 \ 0]; \quad \Psi = \begin{bmatrix} 0 & h_1 & h_0 & 0 \\ h_0 & 0 & 0 & h_1 \end{bmatrix} \\ \phi &= [0 \ g_1 \ g_0 \ 0]; \quad \Phi = \begin{bmatrix} 0 & g_1 & g_0 & 0 \\ g_0 & 0 & 0 & g_1 \end{bmatrix} \end{aligned}$$

The first scale matrix $\mathbf{W}_1^{(N)}$ can be written in the form as:

$$\mathbf{W}_1^{(N)} = \begin{bmatrix} \Psi \\ \Phi \end{bmatrix} \quad (\text{B.8})$$

The transform matrix for the Haar wavelet can be written of the form as[39][40]:

$$\mathbf{W} = \begin{bmatrix} h_0 & h_1 & 0 & 0 & \cdot & \cdot & \cdot \\ g_0 & g_1 & 0 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & h_0 & h_1 & \cdot & \cdot & \cdot \\ 0 & 0 & g_0 & g_1 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & h_0 & h_1 & \cdot \\ 0 & 0 & 0 & 0 & g_0 & g_1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (\text{B.9})$$

In the case of Haar transform the scaling and wavelet values have been given in[41] with $[h_0, h_1]=[1/2, 1/2]$ and $[g_0, g_1]=[1/2, -1/2]$. The wavelet transform matrix can be written as:

$$\mathbf{W} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & \cdot & \cdot \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 & \cdot & \cdot \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & \cdot & \cdot \\ 0 & 0 & 1/2 & -1/2 & 0 & 0 & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 1/2 & -1/2 & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot \end{bmatrix} \quad (\text{B.10})$$

APPENDIX C: Determination of Wavelet Domain Auto Correlation Matrix with AR(2) Process

Using (B.10) and (39), we have

$$\mathbf{W}_N \mathbf{R}_N^{-1} = \begin{pmatrix} 1/2 & 1/2 & 0 & \cdot & 0 & 0 \\ 1/2 & -1/2 & 0 & \cdot & 0 & 0 \\ \cdot & \cdot & & & & \\ \cdot & \cdot & & & & \\ 0 & 0 & 0 & \cdot & 1/2 & 1/2 \\ 0 & 0 & 0 & \cdot & 1/2 & -1/2 \end{pmatrix} \times \begin{pmatrix} 1 & \rho_1 & \rho_1^2 & \cdot & \cdot & \rho_1^{N-1} \\ \rho_1 & 1 & \rho_1^2 & \cdot & \cdot & \rho_1^{N-2} \\ \rho_1^2 & \rho_1 & 1 & \cdot & \cdot & \rho_1^{N-3} \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & & \cdot & \cdot \\ \rho_1^{N-1} & \rho_1^{N-2} & \cdot & \cdot & \cdot & 1 \end{pmatrix} \quad (\text{C.1})$$

$$\mathbf{W}_N \mathbf{R}_N^1 = \frac{1}{2} \begin{pmatrix} (1+\rho_1) & (1+\rho_1) & \dots & \rho_1^{N-2}(1+\rho_1) \\ (1-\rho_1) & -(1-\rho_1) & \dots & (-1)^{N-1} \rho_1^{N-1}(1-\rho_1) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_1^{N-2}(1+\rho_1) & \rho_1^{N-3}(1+\rho_1) & \dots & (1+\rho_1) \\ \rho_1^{N-2}(1-\rho_1) & \rho_1^{N-3}(1-\rho_1) & \dots & -(1-\rho_1) \end{pmatrix} \quad (\text{C.2})$$

$$\mathbf{W}_N \mathbf{R}_N^1 \mathbf{W}_N^H = \frac{1}{4} \begin{pmatrix} 2(1+\rho_1) & 0 & \dots & \rho_1^{N-2}(1+\rho_1)^2 \\ 0 & 2(1-\rho_1) & \dots & (-1)^{N-2} \rho_1^{N-3}(1+\rho_1)^2 \\ \vdots & \vdots & \ddots & \vdots \\ \rho_1^{N-3}(1+\rho_1)^2 & -\rho_1^{N-2}(1-\rho_1)^2 & \dots & 0 \\ \rho_1^{N-2}(1-\rho_1)^2 & -\rho_1^{N-3}(1-\rho_1)^2 & \dots & 2(1-\rho_1) \end{pmatrix} \quad (\text{C.3})$$

$$\text{Diag}(\mathbf{W}_N \mathbf{R}_N^1 \mathbf{W}_N^H) \mathbf{W}_N = \frac{1}{4} \begin{pmatrix} 1+\rho_1 & 0 & 0 & \dots & 0 \\ 0 & 1-\rho_1 & 0 & \dots & 0 \\ 0 & 0 & 1+\rho_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1-\rho_1 \end{pmatrix} \quad (\text{C.4})$$

Again, $\mathbf{D}_N^1 = \mathbf{W}_N^H \{ \text{Diag}(\mathbf{W}_N \mathbf{R}_N^1 \mathbf{W}_N^H) \mathbf{W}_N \}$, and we have the numerical value as:

$$\mathbf{D}_N^1 = \frac{1}{8} \begin{pmatrix} 1+\rho_1 & 1-\rho_1 & 0 & \dots & 0 \\ 1+\rho_1 & -(1-\rho_1) & 0 & \dots & 0 \\ 0 & 0 & 1+\rho_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1+\rho_1 & -(1-\rho_1) \end{pmatrix} \quad (\text{C.5})$$

Similarly, $\mathbf{D}_N^2 = \mathbf{W}_N^H \{ \text{Diag}(\mathbf{W}_N \mathbf{R}_N^2 \mathbf{W}_N^H) \mathbf{W}_N \}$, and we have the numerical value as:

$$\mathbf{D}_N^2 = \frac{1}{8} \begin{pmatrix} 1+\rho_2 & 1-\rho_2 & 0 & \dots & 0 \\ 1+\rho_2 & -(1-\rho_2) & 0 & \dots & 0 \\ 0 & 0 & 1+\rho_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1+\rho_2 & -(1-\rho_2) \end{pmatrix} \quad (\text{C.6})$$

REFERENCES

- [1] B. Widrow, Adaptive filters, in Aspects of Network and System Theory, R. Kalman and N. Declaris, Eds-New York, Holt, Rinehart, Winston, 1971, pp. 563-587.
- [2] L. J. Griffiths, "A continuously adaptive filter implemented as a lattice structure," in proc. ICASSP. Hartford, Conn., 1977. pp. 683-686.
- [3] G. Carayannis, D. Manolakis, and N. Kalouptsidis, "A fast sequential algorithm for least squares filtering and prediction," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, no. 6, pp. 1394-1402, Dec. 1983.
- [4] J. Cioffi and T. Kailath, "Fast recursive least square transversal filters for adaptive filtering," IEEE Trans. Acoust., Speech, Signal Proc., vol. ASSP-32, pp. 304-337, Apr. 1984.
- [5] Vasanthan Raghavan, K. M. M. Prabhu, P.C. W. Sommen, "An analysis of real-Fourier domain-based adaptive algorithms implemented with the Hartley transform using cosine-sine symmetries," IEEE Trans., on Signal Processing, vol. 53, No. 2, Feb. 2005.
- [6] D. Mansour and A. H. Gray Jr., "Unconstrained frequency domain adaptive filter," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, no. 5, pp. 726-734, Oct. 1982.
- [7] M. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in frequency domain," Proc. IEEE, Vol. 66, no. 12, pp. 1658-1659, Dec. 1978.
- [8] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, "Transform-domain LMS algorithm," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, no. 3, pp. 609-615, June 1983.
- [9] D. I. Kim and P. De Wilde, "Performance analysis of the DCT-LMS adaptive filtering algorithm," Signal Process., vol. 80, no. 8, pp. 1629-1654, Aug. 2000.
- [10] R. C. Bilcu, P. Kuosmnien, and K. Egiazarian, "A transform domain LMS adaptive filter with variable step-size," IEEE Trans., Signal Processing, vol. 9, no. 2, Feb. 2002.
- [11] B. Widrow, J. R. Glover, et al., "Adaptive noise canceling: Principles and applications," Proc. IEEE, pp. 1692-1716, Dec. 1975.
- [12] R. W. Harris, D. M. Chabries, and F. A. Bishop, "A variable step (VS) adaptive filter algorithm," IEEE Trans., Acoustic, Speech, Signal Processing, ASSP-34, pp. 309-316.

- [13] T. J. Shan and T. Kailath, "Adaptive algorithms with an automatic gain control feature," *IEEE Trans. Circuits Syst.*, CAS-35, pp. 122-127, Jan. 1988.
- [14] Sen M. Kuo and Dennis R. Morgan, *Active noise control systems: algorithms and DSP implementations*, John Wiley and Sons, Inc., New York, NY, 1996.
- [15] Paulo S. R. Diniz, *Adaptive filtering-algorithms and practical implementation*, 3rd Ed., Kluwer Academic Publishers, New York, NY, 2008.
- [16] B. Farhang-Boroujeny, *Adaptive Filters: Theory and applications*, Chichester, U.K.: Wiley Ltd., 1998.
- [17] Yue Wang, Chun Zhang, and Zhihua Wang, "A new variable step-size LMS algorithm with application to active noise control, *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, China, 2003, 6-10 April.
- [18] F. Casco, H. Perex, R. Marcelin, and M. Lopez, "A two-step size NLMS adaptive filter algorithm," *Conference Proc. On Singapore ICCS*, vol. 2, Nov. 1994, pp. 814-819.
- [19] Z. Ramadan and A. Poularikas, "A variable step-size adaptive noise canceller using signal to noise ratio as the controlling factor," *Proc. Of the 36th IEEE Southeastern Symposium on System theory (SSST)*, Atlanta, Georgia, pp. 384-388., March 2004.
- [20] Zhao Shengkui, Man Zhihong, and Khoo Suiyang, "Modified LMS and NLMS algorithms with a new variable step-size," *IEEE ICARCE* 2006.
- [21] Paulo S. R. Diniz, Marcello L. R. de Campos, and Andreas Angeloniou, "Analysis of LMS-Newton Adaptive Filtering Algorithms with Variable convergence factor," *IEEE Trans.*, on Signal Processing, vol. 43, no. 3, March 1995.
- [22] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Englewood cliffs, NJ, Prentice-Hall, 1985.
- [23] Shengkui Zhao, Zhihong Man, Suiyang Khoo, and Hong Ren Wu, "Stability and Convergence Analysis of Transform-Domain LMS Adaptive Filters With Second-Order Autoregressive Process," *IEEE Trans.*, on Signal Processing, Vol. 57, No. 1, January 2009.
- [24] N. Erdol and F. Basbug, "Wavelet transform based adaptive filters: Analysis and new results," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 617-630, Mar. 1997.
- [25] D. Lee Fugal, *Conceptual Wavelets in Digital Signal Processing*, Space & Signals Technologies LLC, 2006.
- [26] D. F. Marshall, W. K. Jenkins, and J. J. Murphy, "The use of orthogonal Transform for improving performance of adaptive filters," *IEEE Trans. Circuits and Systems*, vo. 36(4), pp. 474-483, April 1989.
- [27] F. Beaufays, "Transform-domain adaptive filters: An Analysis approach," *IEEE Trans.*, on Signal Processing, vol. 43(2), pp. 422-431, February 1995.
- [28] S. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*, 3rd Edition, John Wiley & Sons Ltd, 2006, West Sussex.
- [29] Georges Oppenheim, *Wavelets and Their Applications*, ISTE Ltd. 1st edition, London, UK.
- [30] Albert Boggess, Francis J. Narcowich, *A First Course in Wavelet with Fourier Analysis*, Prentice Hall, Upper Saddle River, NJ 07458.
- [31] Lee. J. C., and Un, C. K., "Performance of Transform domain LMS adaptive digital filters," *IEEE Trans.*, on Acoustics, Speech, and Signal Processing, vol. ASSP-34, pp. 499-510, 1986.
- [32] F. Keinert, *Wavelet and multi-wavelets*, CHAPMAN & Hall/CRC, 2004.
- [33] R. M. Reid, "Some eigenvalue properties of persymmetric matrices," *SIAM Review*, vol. 39, no. 2, pp. 313-316, Jun, 1997.
- [34] Erdol, N., and Basbug, F., "Performance of Wavelet Transform Based Adaptive Filters," *Proc. ICASSP* 1993, vol. III, pp. 500-503 (1993).
- [35] Srinath H. and Tewfik, A. H., "Wavelet Transform Domain LMS Algorithm," *Proc. ICASSP* 1993, vol. III, pp. 508-510 (1993).
- [36] Stephen M. Laconte, Shing-Chung Ngan, and Xiaoping Hu, "Wavelet Transform-Based Wiener Filtering of Event - Related fMRI Data," *Magnetic Resonance in Medicine*, Wiley-Liss, Inc., pp 746-757, 2000.
- [37] Doros Lova Cki, M. and Fan, H., "Wavelet based adaptive filtering," *Proc. ICASSP* 1993. vol. III; pp. 488-491, 1993.
- [38] Bernard Widrow and Eugene Walach, *Adaptive inverse control: A signal processing approach*, Reissue Edition, Institute of Electrical and Electronics Engineering, 2008..
- [39] Gilbert Strang and Truong Nguyen, *Wavelets and Filter banks*, Wellesley-Cambridge press, 1997.
- [40] Arne Jensen and Anders la Cour-Harbo, *The discrete Wavelet transform*, Springer- 2001.
- [41] Woolfson MS, Huang XB, and Crow JA, "Time varying wiener filtering of the fetal ECG using the Wavelet transform." *IEE colloquium on Signal Processing in Cardiology*, 1995, pp. 11.1-11. 6.
- [42] Alexander D. Poularikas, and Zayed M. Ramadan, *Adaptive Filtering Primer with MATLAB*, Taylor & Francis Group, 2006.