Comparison of Wiener Filter solution by SVD with decompositions QR and QLP

EDWIRDE LUIZ SILVA AND PAULO LISBOA
Departamento de Matemática e Estatística
Universidade Estadual da Paraíba - UEPB
Rua Juvêncio Arruda, S/N Campus Universitário (Bodocongó).
Cep: 58.109 - 790 Campina Grande - Paraíba
BRASIL
School of Computing and Mathematical Sciences
Liverpool John Moores University
Byrom Street, Liverpool, L3 3AF
ENGLAND

Abstract: - This paper presents a sequential decomposition of the data auto-correlation matrix using SVD, QR-method and QLP as pre-processors from the design of Wiener Filters. It is shown that this approach is effective for noise reduction by improving the SNR while reducing CPU time for a range of input signal lengths, filter length and noise variance. The theoretical and practical aspects of the proposed approach are introduced and compared to those obtained from simulation results.

Keywords: - Wiener Filter; SNR; SVD; QR; QLP.

1 Introduction

In this paper we propose using the pivoted QLP decomposition of the autocorrelation matrix A from estimation of Wiener filter coefficients for noise reduction. The effect of the decomposition on SNR (Signal-to-Noise Ratio) and CPU-time variations with respect to input signal length and noise variance is analyzed and illustrated with simulated data.

The rest the paper is organized as follows. Section 2 presents Wiener Filter, Section 3 presents the detection of the Numerical rank of the QLP, Section 4 presents system models theory and reports on the quantitative performance comparison SVD, QR-method and QLP, Section 5 presents experimental results and Section 6 offers conclusions from this study.

2 Wiener Filter

The Wiener Filter is a well-know adaptive filter considered to be optimal for noise reduction applications. The signal reconstruction with a equation

$$y(x) = (b * \widetilde{h})(x) \qquad (1)$$

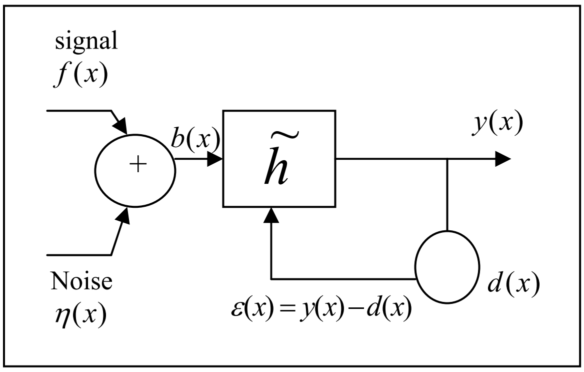

Fig. 2 Wiener Filter

$y(n)$ is the filter output and $d(n)$ is desired, clean signal, and $b(n) = f(n) + \eta(n)$ is the original signal, contaminated by noise The box $\widetilde{h}$ represents the filter operation.

In this equation and in fig. 1, $f(n)$ represents data, $y(n)$ output and $d(n)$ represent the desired or reference. We assume that the signal and the noise correlation matrix are uncorrelated, i.e. $\phi_{f(n),\eta(n)} = 0$. The purpose of the filter is to reconstruct signal $f(x)$ as accurately as possible from $b(x)$ with $d(x)$ being identical signals. Consider the standard linear equation to model a signal with noise

$$b(n) = f(n) + \eta(n) \quad (2)$$

We can formulate the mean square error as

$$\varepsilon(x) = E\{[d(x) - y(x)]^2\} \quad (3)$$

Inserting (2) and (3) in (4) and minimizing this error, we can write

$$\nabla\varepsilon(x) = \underline{0}$$
$$\tilde{h} = E\{\underline{f(x)}\,\underline{f(x)}\}^{-1} E\{d(x)\underline{f(x)}\} \quad (4)$$

where $E\{\underline{f(x)}\,\underline{f(x)}'\} = A_{xx}$ is the autocorrelation matrix of the input.

Defining $\underline{Z} = E\{d(x)\underline{f(x)}\}$, which we will call *Z-vector*, the previous equation can be re-written as

$$\tilde{h} = A_{xx}^{-1}\underline{Z} \quad (5)$$

Where the filter $\tilde{h}$ needs to be estimated from the data. This can be achieved, for instance, via the QLP of A_{xx}^{-1}.

We can expand (3) with known algebraic rules. Then we can well-take the Fourier transform of the expression to find the power spectrum

$$\varepsilon(x) = \sum_{i=1}^{n} (f(x_i) - b(x_i)\tilde{h}_i)^2 \quad (6)$$

Inserting (2) in (6), we can write

$$\varepsilon(x) = \sum_{i=1}^{n} ((f(x_i) - (f(x_i) + \eta(x_i))\tilde{h}_i)^2$$

Where $i = 1,...n$ number of is inputs signal $f(x)$.

The signal to noise ratio is the quotient between the powers of the signal $f(x_i)$ and $\eta(x_i)$ is the power of the noise in the estimative the design signal $d(x)$

$$H(x_i) = \frac{(|f(x_i)|)^2}{(|\eta(x_i)|)^2} \quad (7)$$

Our aim in this study is to combine the use of the Wiener filter with different matrix decompositions applied to the auto correlation matrix A_{xx}.

We also need to define a Toeplitz matrix, which is formed with positive values of the auto correlation matrix A_{xx}. It has constant values along negative-sloping diagonals [7].

$$\begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n+1} \\ a_1 & a_0 & a_{-1} & \ddots & \vdots \\ a_2 & a_1 & a_0 & \ddots & a_{-2} \\ \vdots & \ddots & \ddots & \ddots & a_{-1} \\ a_{n-1} & \cdots & a_2 & a_1 & a_0 \end{bmatrix}.$$

3 Detection of the CPU-time of the QLP, QR and SVD decompositions

It is clear that Singular value decomposition may be easily avoided by computation the rank-r principal subspace of $A_{xx}^{-1} = R_{yx}R_{xx}^{-1}R_{yx}'$. But the product these three matrices, involves additional computational cost and this is made worse by the calculation of the inverse matrix R_{xx}^{-1} [1].

The QR decomposition produces an upper triangular matrix R of the same dimension as A_{xx} and a unitary matrix Q so that $A_{xx} = QR$, where $R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$, where R_1 is an upper triangular matrix. For the full matrix A_{xx}, produces an effective decomposition in which E is a permutation vector, so that $A_{xx}(:,E) = QR$. The diagonal elements of R are called the R-values of A_{xx}; the column permutation E is chosen so that $abs(diag(R))$ is monotonically decreasing.

To motivate the decomposition QLP consider the partitioned R-factor [2].

$$R = \begin{pmatrix} \kappa_{11} & \kappa_{12}^T \\ 0 & R_{22} \end{pmatrix}$$

of the pivoted QR decomposition. [2] Has observed that κ_{11} is an underestimate of $\|A_{xx}\|_2$. A better estimate is the norm $l_{11} = \sqrt{\kappa_{11}^2 + \kappa_{12}^T\kappa_{12}}$ of the first row of R. We can calculate that norm by post multiplying R by a Householder transformation H_1. We can write [3].

$$RH_1 = \begin{pmatrix} l_{11} & 0 \\ l_{12} & \hat{R}_{22} \end{pmatrix}$$

We can obtain an even better value if we interchange (Π) the largest row of R with the first

$$\Pi_1 RH_1 = \begin{pmatrix} l_{11} & 0 \\ l_{12} & \hat{R}_{22} \end{pmatrix} \quad (8)$$

Now if we transpose (8), we see that it is the first step of pivoted Householder traingularization applied to R^T. If we continue this reduction and transpose the result, we obtain a triangular decomposition of the form [2][3].

$$\Pi_L^T Q^T A_{xx} \Pi_R P = \begin{pmatrix} L \\ 0 \end{pmatrix}$$

We will call this the pivoted QLP decomposition of A_{xx} and will cal the diagonal elements of L the L-values of A_{xx}.

The diagonal elements of R are called the R-values of A, those of L are called the L-values of A.

The way we motivated the pivoted QLP decomposition suggests that it might provide better approximations to the singular values of the original matrix A_{xx} than does the pivoted QR decomposition. An advantage of the QLP decomposition is that merely applies the program twice, once to A_{xx} and once to R^T [2].

The implementation of the Matlab package permits the QLP: $[P, Q, L, pr, pl] = qlp(A_{xx})$ to determine the numerical rank of matrix A_{xx}.

Thus, we have a simple QLP algorithm as follows

1. **define** matrix A, which consists of $A_{xx}^{-1} = P.L^{-1}Q^T$
2. **calculate** the orthogonal matrices P and Q which reduce the matrix A_{xx} to lower diagonal form.
3. **identify the** diagonal of lower-triangular matrix L
4. **sort** the diagonal elements by size.

Calculators the CPU-time for the decomposition of matrix of Toeplitz A_{xx} and also the solution of linear systems $\widetilde{h} = A_{xx}^{-1}\underline{Z}$, Table 2 shows us that the QLP and QR are on average 2.3 times faster than decomposition SVD.

Table 1 Comparison between the times required to calculate the matrix of Toeplitz A_{xx} using SVD, QR and QLP decompositions

Decompositions	Time (cputime)
QLP and QR	0.03
SVD	0.07

In summary, the application of QLP and QR decompositions reduces computational complexity while giving similar results to SVD in the calculation of Wiener-Hopf coefficients.

5 System Model and Theory

There are many ways to compute A_{xx}. With QLP, we can write

$$A_{xx}^{-1} = P.\Sigma^{-1}Q^T \Rightarrow w = P\Sigma^{-1}Q^T Z \quad (9)$$

This approach is illustrated with simulated data using coefficients $\beta = [0.8 \;\; 0.5 \;\; -0.1 \;\; -0.3]$ driving a recursive filter to generate the following input signal [4]

$$f(x) = \sum_{i=5}^{1500}\sum_{j=1}^{4} f(i-j)\beta(5-j) + \varepsilon(x),$$

where $\varepsilon(x) \sim N(0, \sigma^2)$. The desired signal is given by the noise-error component

$$d(x) = \sum_{i=5}^{1500}\sum_{j=1}^{4} f(i-j)\beta(5-j)$$

A total of 500 samples were simulated to implement the expectation operator. The input signal length was taken as $N = 2^n$ where $n = 1,2,...,10$ n was limited to 10 to control the matrix size and computational time required for the matrix size and computational time required for the recursive algorithms to work efficiently for such a large number of realizations [4].

The function $\log_2$ for a positive number x, the power y to which some number 2 must be raised to give x. The number 2 is the base of the logarithm.

The input data vector $f(x)$, desired result $d(x)$ and the noise $\eta(x)$ vector were all of size 1x1500. The matrix of auto correlation was created from the input data, and the Toeplitz matrix R was created with 128 positive values of matrix of auto correlation.

Two filters at the same frequency were applied to input signal the data X corrupted by the addition of noise η. The filters described by the numerator coefficient vector $b = \widetilde{h}$ (1:M, 1), where M is signal filter length and denominator coefficient vector formed with ones.

2^{10} samples were taken in order to realize A_{xx} the expectation operator.

The mean achieved SNR was compared for each input length and also for each filter order. Similarly the CPU-time was also compared for each signal length and filter length. All the simulations of these recursive algorithms are carried out in MATLAB 7 environment running in a Pentium(R) 4, 2.40-GHZ CPU.

6 Simulation results

The Figures 1, 6 and 10 contain the results of three decompositions for signal to noise ratio versus variation filter order, shows us that maintaining the same resulting Signal-to-Noise ratio for all decompositions. Here, the L-values and R-values perform essentially as well as the singular value S-values.

The results in the Figure 3, shows us that CPU time versus Signal length in the QLP and QR decompositions (Figure 8 and 12) reduced the CPU time from a value of 0.048 to 0.042. These confirm the results in the table 1, where the QLP and QR are faster that SVD. The computation of SVD decomposition is expensive. For this reason, this paper has proposed others alternatives. Of these the pivoted QLP and QR decompositions is widely recommended because of its simplicity.

The results in the Figures 4, shows us that CPU time w.r.t. filter length in the QLP reduced the CPU time from a value of 0.084 (SVD) and 0.047 (QR) to 0.043. In other words, there is monotonic and exponentially increasing if we keep filter order fixed and vary input signal length [4].

A higher SNR means less noise and better detection, the QLP (figure 1) has better results in comparison with SVD (Figure 10).

The Figure 5 shows us very similar results for all three decompositions. It shows us that if we increase the noise variance, CPU time remains constant because the filter order and input signal length are practically constant independent of decompositions SVD, QR and QLP.

6.1 Results using the decomposition QLP

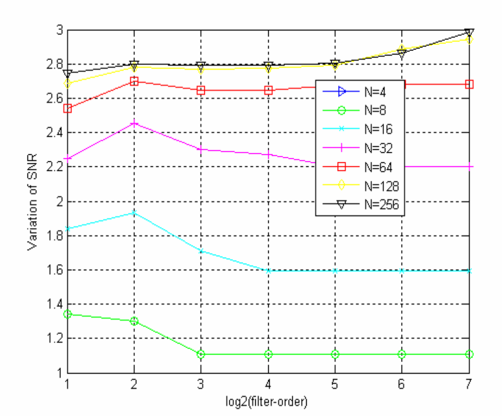

Fig. 1 SNR variation w.r.t filter order

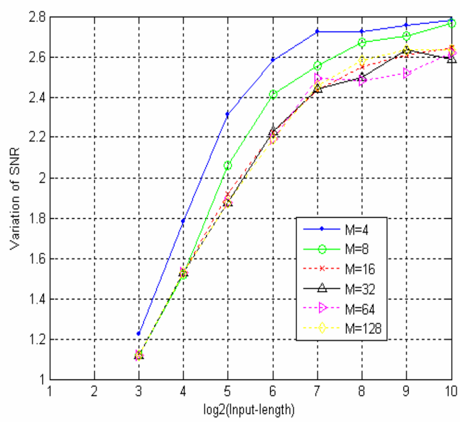

Fig. 2 SNR variation w.r.t. i/p Signal length

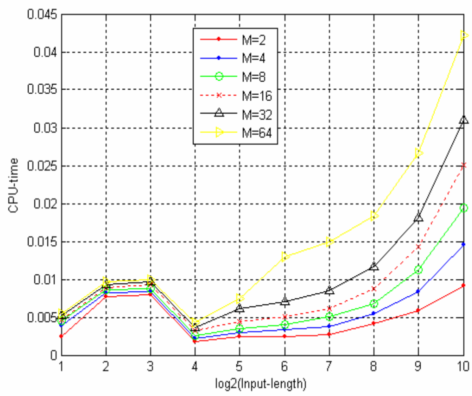

Fig. 3 CPU time w.r.t. i/p Signal length

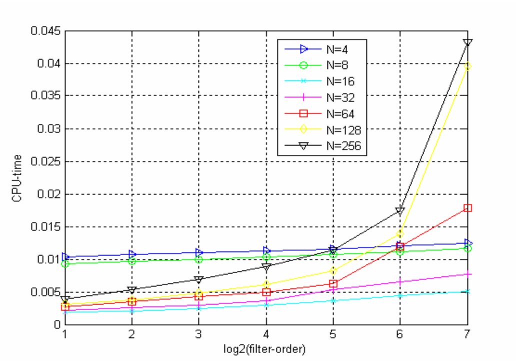

Fig. 4 CPU time w.r.t. filter length

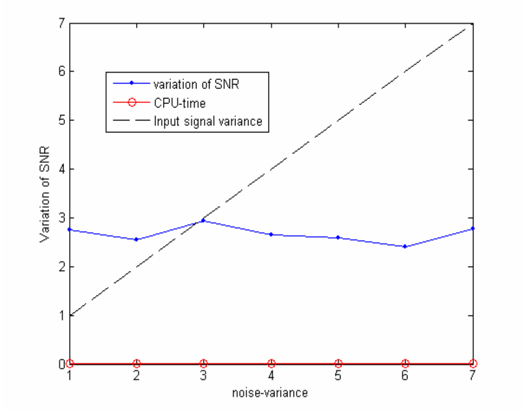

Fig. 5 SNR and CPU time variation w.r.t. noise variance.

6.2 Results using the decomposition SVD

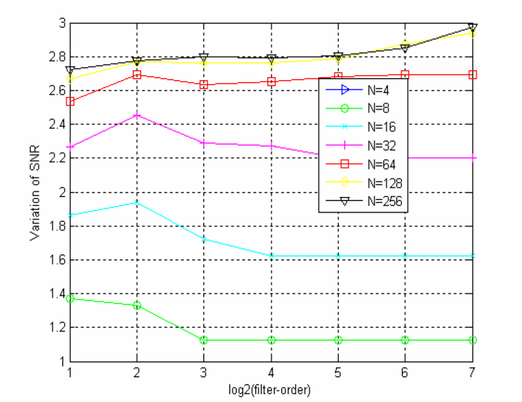

Fig. 6 SNR variation w.r.t filter order

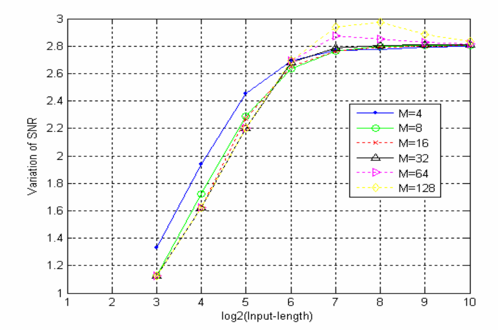

Fig. 7 SNR variation w.r.t. i/p Signal length

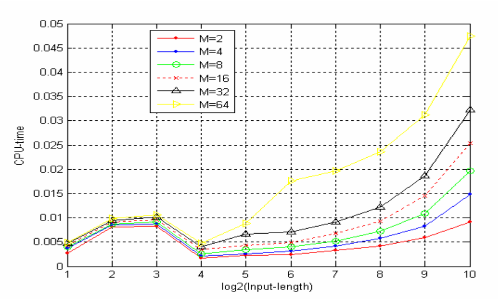

Fig. 8 CPU time w.r.t. i/p Signal length

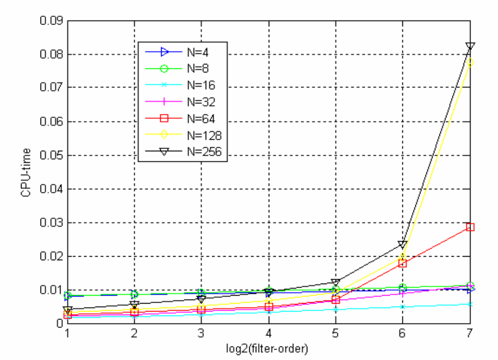

Fig. 9 CPU time w.r.t. filter length

6.3 Results using the decomposition QR

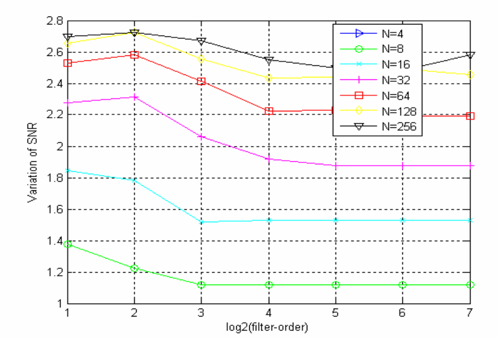

Fig. 10 SNR variation w.r.t filter order

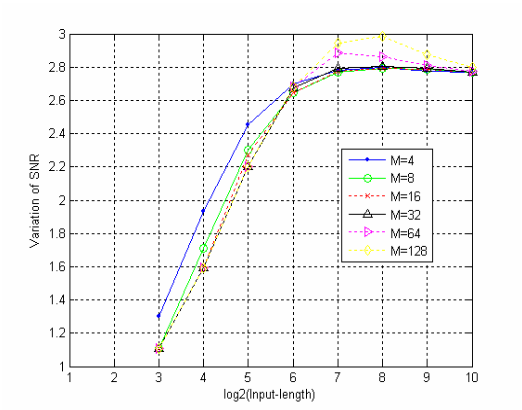

Fig. 11 SNR variation w.r.t. i/p Signal length

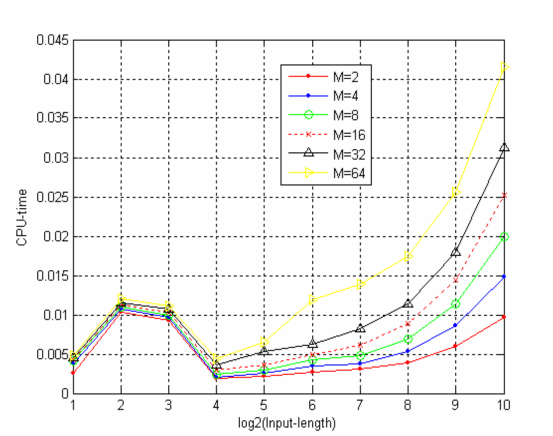

Fig. 12 CPU time w.r.t. i/p Signal length

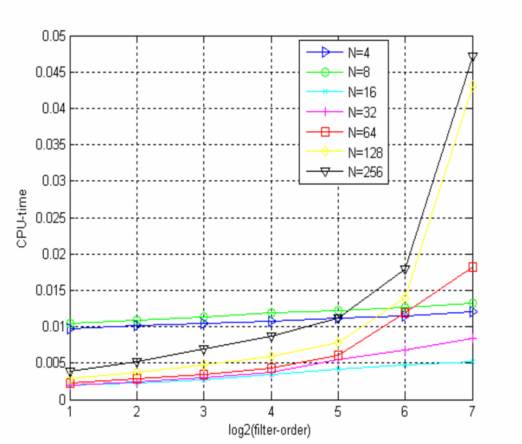

Fig. 13 CPU time w.r.t. filter length

6 Conclusion

The application of QLP decomposition to estimate the autocorrelation coefficient in Wiener-filter design is shown to be effective by reducing computational cost by an order of magnitude compared with the use of SVD, while maintaining or increasing the resulting Signal-to-Noise ratio. The Figures above shows the behaviour of the QLP and QR decompositions reduced the CPU time significantly better that SVD. The QLP and QR decompositions are important when pretends to reduce a matrix size and a computational time required in this our for the recursive algorithm.

The QLP and QR reduces computational complexity drastically to give same results that SVD as given by Wiener equation but it takes very less computational time. This was demonstrated in a noise reduction task using simulated data.

References:

[1] Yingbo Hua and Maziar Nikpour, Computing the Reduced Rank Wiener Filter by IQMD. *Senior Member, IEEE Signal Processing Letters*, Vol.6, No. 9, 1999, pp.240-241.

[2] Stewart, G.W, *Matrix Algorithms: Basic Decompositions*. SIAM Publications. Philadelphia, U.S.A., 1998.

[3] Sterwart, G.W, On an Inexpensive Triangular Approximation to the Singular Value Decomposition, *A Conference in Honor of G. W. (Pete) Stewart*, 1997. pp.1-16.

[4] Heeralal C, Wiener filter solution by Singular Value Decomposition (SVD). *Electrical & Computer Engineering Department.* 2006.