Optimised Wiener Filtering in Overdetermined Systems

Joshua L. Sendall
Council for Scientific and Industrial Research
Pretoria, South Africa

Warren P. du Plessis
University of Pretoria
Pretoria, South Africa

***Abstract*—The Wiener filter is prevalent in many adaptive filtering applications, but can also be a computationally burdensome part of a signal processing chain. This paper presents methods to reduce the computational complexity and memory footprint of Wiener filters where the filter is highly overdetermined. The optimisations are demonstrated in the context of passive radar and result in a reduction of over 40 times in processing time.**

***Index Terms*—Adaptive filter, passive radar, passive coherent location (PCL), direct path interference (DPI), clutter cancellation, radar clutter, radar signal processing.**

I. Introduction

Covariance matrices are an important concept in many adaptive signal processing applications. These include minimum variance distortionless response (MVDR) filtering [1], signal cancellation [2], and space-time adaptive processing (STAP) radar [3]. While these techniques can theoretically result in optimal performance, in practical situations, efficiently estimating the covariance is still an open topic in research. Typically, there are two aspects of covariance matrix estimation that are challenging. Either there is limited information and therefore insufficient samples to accurately estimate the covariance matrix [4], [5], or the number of samples is so large that the estimation of the covariance matrix becomes computationally intensive.

The first case is handled using rank reduction methods [3]. A solution for reducing the computational burden for the second case is presented in this paper. Many signal cancellation environments require delayed versions of a template signal to be removed, although in some instances with a frequency offset [6]. This structure induces redundancy, which can be exploited to reduce computation time and memory requirements.

This paper presents two strategies for efficiently estimating the covariance matrix for highly overdetermined systems. The first exploits the redundancy in the inner product produced by the covariance matrix's tapped delay line structure by breaking up the matrix-matrix multiplication into a matrix-vector multiplication and a set of rank-1 updates. The second method further exploits the redundancy by computing the matrix-vector product using fast Fourier transforms (FFTs).

Both methods significantly reduce the computational time of the filter. The memory requirements of the filters are also significantly reduced as the sample matrix is implicitly represented, rather than having to be stored in memory. Then the computational benefit of using a Toeplitz covariance matrix approximation is analysed. The validity of the Toeplitz approximation on the filter's effectiveness is analysed for filter sizes in overdetermined systems.

II. Wiener Filter

The Wiener filter minimizes the distance between a measured stationary process and a desired process. The exposition below is based on the outline provided by [2].

The filter is described by defining a measured signal as

$$x_n = s_n + z_n \qquad n = 0, 1, \ldots, N-1 \tag{1}$$

where N is the number of samples, and s_n and z_n are the nth samples of the desired and undesired signal components respectively. The output of the filter is then the estimate of the desired signal

$$\widehat{s}_n = x_n - \widehat{z}_n \tag{2}$$

where $\widehat{s}$ is the estimate of the desired signal, and $\widehat{z}$ is the estimated undesired signal obtained from

$$\widehat{z}_n = \sum_{m=0}^{M-1} h_m y_{n-m} \tag{3}$$

with M the number of filter coefficients, y a template of the desired signal, and h_m the filter weights.

The undesired signal can be estimated with the minimum squared error by determining filter weights by solving the Wiener equations for all values of $l = 0, 1, \ldots, M-1$ at sample n [2],

$$\sum_{m=0}^{M-1} E\left[y_{n-m} y_{n-l}^*\right] h_m = E\left[x_n y_{n-l}^*\right] \tag{4}$$

$$\mathbf{R}_{yy}\mathbf{h} = \mathbf{R}_{xy} \tag{5}$$

where $E[\cdot]$, $\cdot^*$, and $\cdot^H$ denote the expectation operator, complex conjugation, and the Hermitian transpose respectively. The $M \times 1$ vector $\mathbf{h}$ is formed from the filter coefficients, $\mathbf{R}_{yy}$ is the $M \times M$ auto-covariance matrix of y, and $\mathbf{R}_{xy}$ is the $M \times 1$ covariance matrix of x and y.

In practical systems the covariance matrix is not usually known and it is necessary to estimate the covariance matrix from sample data. Given N samples from the reference channel,

TABLE I: The approximate computational cost of the cancellation algorithms considered.

Algorithm	Computational cost
LU factorisation	$\frac{2}{3}M^3 + 2NM^2$
Cholesky factorisation	$\frac{1}{3}M^3 + 2NM^2$
QR factorisation	$-\frac{2}{3}M^3 + 2NM^2$
CGLS	$(P+1)(4NM)$

it is possible to form the $N \times M$ matrix

$$\mathbf{Y} = \begin{bmatrix} y_0 & y_1^* & \cdots & y_{M-1}^* \\ y_1 & y_0 & \cdots & y_{M-2}^* \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-1} & y_{N-2} & \cdots & y_{N-M} \end{bmatrix} \tag{6}$$

by delaying each column by an additional sample, thereby allowing $\mathbf{R}_{yy}$ to be estimated as

$$\widehat{\mathbf{R}}_{yy} = \frac{1}{N}\mathbf{Y}^H\mathbf{Y}. \tag{7}$$

Further, placing n samples from the measured signal in the $N \times 1$ vector $\mathbf{x}$ allows $\mathbf{R}_{xy}$ to be approximated by

$$\widehat{\mathbf{R}}_{xy} = \frac{1}{N}\mathbf{Y}^H\mathbf{x}. \tag{8}$$

Substituting (7) and (8) into (5) results in

$$\frac{1}{N}\mathbf{Y}^H\mathbf{Y}\mathbf{h} = \frac{1}{N}\mathbf{Y}^H\mathbf{x} \tag{9}$$

which can be simplified to the form of the least squares (LS) problem

$$\mathbf{Y}\mathbf{h} = \mathbf{x} \tag{10}$$

under the assumption that the inverse of $\mathbf{Y}^H$ exists. An LS method, such as QR decomposition, in used to determine the filter weights. Alternatively, a square system solver, such as LU decomposition, can be employed to solve (5) directly by using the approximate values of $\widehat{\mathbf{R}}_{yy}$ and $\widehat{\mathbf{R}}_{xy}$ from (7) and (8).

In order to achieve an accurate estimate of the covariance matrices, N should be much greater than M [4]. With this in mind, the approximate number of complex operations required to solve the system via a number of methods is shown in Table I. The LU and Cholesky factorisations [7] solve (5), while the QR factorisation [7] and conjugate-gradient least squares (CGLS) [8] solve (10).

The QR filter uses Householder reflections to directly invert a rectangular matrix, where $N > M$. Hence, the terms seen in Table I are both from the factorisation. Due to the negative term, the QR filter requires fewer operations than the Wiener filter (using LU and Cholesky Factorisation). The conjugate-gradient least squares (CGLS) algorithm does not directly solve (10), but instead iterates towards the solution. The number of iterations is P.

The computational complexity of the LU and Cholesky factorisations comprises two terms. The M^3 term is related to the factorisation of $\mathbf{R}_{yy}$ in (5), while the NM^2 term is related to the calculation of $\widehat{\mathbf{R}}_{yy}$ in (7). The majority of the computational load is associated with the calculation of $\widehat{\mathbf{R}}_{yy}$ under the assumption $N \gg M$.

A further point to note is that $\mathbf{R}_{yy}$ is a Hemitian Toeplitz matrix because x is stationary. However, $\widehat{\mathbf{R}}_{yy}$ is only approximately Toeplitz as practical environments are not stationary processes [9]. The effects of this observation are further explored in Section III-E. The use of Toeplitz solvers, such as the Bareiss algorithm [10], results in sub-optimal cancellation.

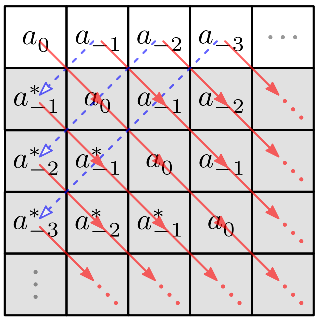

Fig. 1: Procedure for filling $\widehat{\mathbf{R}}_{yy}$.

III. Optimisation

This section shows how the mathematical properties of the computations described in Section II can be exploited to significant reduce the number of computations required to implement a Wiener filter.

A. Hermitian Structure of Covariance Matrix

The first opportunity for optimisation emerges from the exploitation of the Hermitian structure of $\widehat{\mathbf{R}}_{yy}$. Only the lower triangle of $\widehat{\mathbf{R}}_{yy}$ needs be explicitly calculated, while the upper triangle can be filled using

$$\widehat{\mathbf{R}}_{yy}(k,l) = \widehat{\mathbf{R}}_{yy}^*(l,k) \tag{11}$$

where $\mathbf{A}(k,l)$ is the element of $\mathbf{A}$ in kth row and lth column. The complexity in calculating $\widehat{\mathbf{R}}_{yy}$ is reduced from $\mathcal{O}(2NM^2)$ to $\mathcal{O}(NM^2)$.

B. Tapped Delay Line Covariance Matrix

The tapped delay line structure of $\mathbf{Y}$ can be exploited to reduce the computational load. This observation means that is only necessary to store $N + M - 1$ unique samples, and not all $N \times M$ elements, as the elements in $\mathbf{Y}$ are repeated.

This reduces the storage requirements, and the resulting lower number of unique memory accesses also improves cache efficiency, which is especially valuable on processors such as graphics processing units (GPUs) which have a high arithmetic intensity [11].

The tapped delay line structure of $\mathbf{Y}$ gives $\mathbf{R}_{yy}$ a Toeplitz structure. The approximation of $\widehat{\mathbf{R}}_{yy}$ as a Toeplitz matrix reduces the computation of $\widehat{\mathbf{R}}_{yy}$ further by only requiring the full inner dot product of the first row of $\widehat{\mathbf{R}}_{yy}$ ($2NM$ complex operations).

The remainder of $\widehat{\mathbf{R}}_{yy}$ can be populated by following a sequential update approach for each element of the first row, shown in Fig. 1. Each (gray) element is calculated along the diagonal as

$$\begin{aligned}\widehat{\mathbf{R}}_{yy}(k,l) = \widehat{\mathbf{R}}_{yy}(k-1,l-1) - \mathbf{Y}(k,k-1)\mathbf{Y}^*(l,k-1)\\ + \mathbf{Y}(k,k+N)\mathbf{Y}^*(l,k+N). \quad (12)\end{aligned}$$

Therefore, the calculation of $\widehat{\mathbf{R}}_{yy}$ has a complexity of $\mathcal{O}(2NM + (3/2)\,M^2)$.

C. Exploiting Fast Correlation

Fast correlation can be used to calculate $\widehat{\mathbf{R}}_{xy}$ and the first row of $\widehat{\mathbf{R}}_{yy}$. This involves defining,

$$\mathbf{a} = \begin{bmatrix} \mathbf{y}^T & \mathbf{0}_{1,M-1} \end{bmatrix}^T \quad (13)$$

where $\mathbf{y}$ is the $N \times 1$ vector of reference samples, $\cdot^T$ denotes a transpose operation, and $\mathbf{0}_{k,l}$ represents an $k \times l$ matrix of zeros, giving

$$\mathbf{b} = [y_0, y_1, \ldots, y_{N-1},\ y_{M-1}, y_{M-2}, \ldots, y_{-1}]\,. \quad (14)$$

These $1 \times (M+N-1)$ vectors are then correlated using fast correlation which requires $\mathbf{a}$ and $\mathbf{b}$ to be transformed to the frequency domain. The element-wise product is then taken such that

$$\mathbf{c} = \mathscr{F}(\mathbf{a}) \circ \mathscr{F}(\mathbf{b}) \quad (15)$$

where $\mathscr{F}$ represents the Fourier transform, and $\circ$ represents the Hadamard product. The first row of $\widehat{\mathbf{R}}_{yy}$ is calculated by taking the inverse Fourier transform of $\mathbf{c}$. The first row of $\widehat{\mathbf{R}}_{yy}$ can be extracted from the first M elements of the convolved vector.

The calculation of $\widehat{\mathbf{R}}_{yy}$'s first row has a complexity of $\mathcal{O}(6(N+M)[\log_2(N+M)+1])$, compared to the conventional $\mathcal{O}(2NM)$. The correlative method is less complex than the multiplicative method for most practical values of N with $M > 80$. $\mathbf{R_{xy}}$ can be calculated in the same manner.

D. Factorisation Algorithm

Since the computations required to form $\widehat{\mathbf{R}}_{yy}$ can be significantly reduced using (12), the dominant term for computational load becomes the factorisation of $\widehat{\mathbf{R}}_{yy}$.

For a generic matrix, the filter coefficients can be found using LU decomposition. However, the Hermitian nature of $\widehat{\mathbf{R}}_{yy}$ an LDL factorisation [7] can be used. It should be noted that row/column pivoting is required in order for these factorisations to be numerically stable [7]. Due to the low arithmetic intensity of the pivoting, it may be a bottleneck for smaller problem sizes. The Cholesky decomposition will not always successfully factorise $\widehat{\mathbf{R}}_{yy}$, as real data results in $\widehat{\mathbf{R}}_{yy}$ being positive-semi definite. As such, the success of the factorisation must be monitored, and when failure occurs the filter must resort to one of the other factorisation algorithms.

E. Toeplitz Approximation

For a stationary process, the auto-covariance matrix is a Toeplitz positive semi-definite matrix, with each element defined by

$$\mathbf{R}_{ij} = \mathrm{E}\left[(\mathbf{Y}_i - \mu_i)\,(\mathbf{Y}_j - \mu_j)\right] \quad (16)$$

where μ_i is the mean of the ith random vector. For the tapped delay-line case where

$$\mathbf{Y}_i(t_0) = \mathbf{Y}_j(t_0 + T_s(j-i)) \quad (17)$$

where T_s is the sample period, (16) becomes

$$\begin{aligned}\mathbf{R}_{ij} = \mathrm{E}\left[(\mathbf{Y}(t_0 + T_s i) - \mu(t_0 + T_s i))\right.\\ \left.(\mathbf{Y}(t_0 + T_s j) - \mu(t_0 + T_s j))\right] \quad (18)\end{aligned}$$

where $\mathbf{Y}$ is the signal in question and μ is its mean. Considering that practial scenarios are non-stationary processes i.e.

$$\mu(t_0) \neq \mu(t_0 + \tau), \quad (19)$$

it stands that

$$\mathbf{R}_{ij} \neq \mathbf{R}_{(i+n)(j+n)}. \quad (20)$$

It should also be noted that the maximum deviation in time is between $\mathbf{R}_{11}$ and $\mathbf{R}_{MM}$, which is thus $T_s M$.

In a sample covariance matrix there will be $2M$ samples difference between $\widehat{\mathbf{R}}_{11}$ and $\widehat{\mathbf{R}}_{MM}$. The covariance matrix, with $N \gg M$, will therefore be pseudo-Toeplitz. Using this approximation allows $\widehat{\mathbf{R}}_{yy}$ to be represented by an M element vector.

The Toeplitz sample covariance vector is defined as

$$\widehat{T}_{yy} = \mathbf{y}^H \mathbf{Y} \quad (21)$$

where $\mathbf{y}$ is the $N \times 1$ vector containing the reference-channel samples (the first column of $\mathbf{Y}$). This matrix-vector multiplication can be implemented with the same methods described in Section III, except that the fill algorithm is not implemented.

The major advantage of this method is the ability to use a Toeplitz matrix solver which has significantly reduced computational requirements. The Toeplitz solver used in this study is detailed in the appendix, and has a complexity of $\mathcal{O}((11/2)\,N^2)$.

F. Performance Comparison

The performance of each solver is shown in Fig. 2. It can be seen that the QR solver was the slowest for most system sizes, and the Toeplitz solver was the fastest.

The LDL solver was 1.3 times slower than the Cholesky solver for large problem sizes despite having the same dominant complexity term. This demonstrates the performance reduction is caused by row/column pivoting. The LU solver was faster than the LDL solver when $M \leq 1194$.

IV. Numerical Examples

In passive radar, the undesired components are the direct path interference (DPI) and clutter [2]. The Wiener filter can

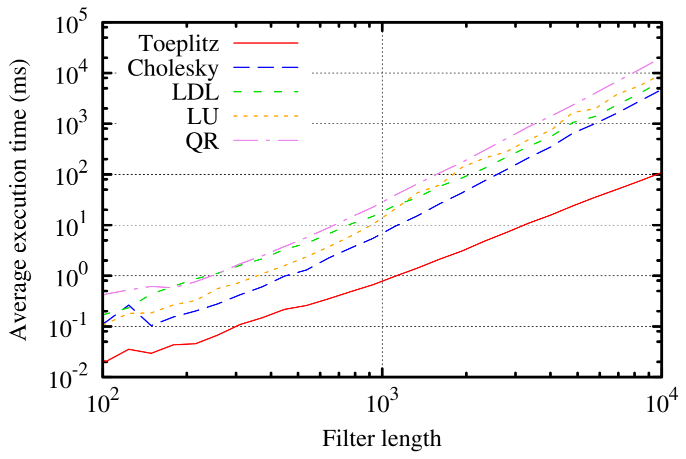

Fig. 2: Mean execution time for square system solvers.

TABLE II: Signal properties of transmission signals.

	FM Radio	DAB	DVB-T
Typical transmission power (kW) [13]	250	10	8
Maximum bandwidth (kHz)	150	1 712	7 600
Sampling frequency (kHz)	200	2 048	8 124

TABLE III: Cancellation filter configuration.

	Estimation samples	Filter taps
FM radio	40×10^3	512
DAB	400×10^3	1 750
DVB-T	800×10^3	4 096

determine these undesired components from the reference signal. The above algorithms were evaluated on three real-world scenarios based on each of the main broadcast signals which have been investigated for passive radar, frequency modulation (FM) terrestrial radio, digital audio broadcast (DAB), and digital video broadcasting – terrestrial (DVB-T) [12]. The most relevant properties of each of these broadcast signals are shown in Table II.

These signal properties translate to three configurations for passive radar systems. Clutter is removed at Doppler frequency offsets lower than 5 Hz, and is implemented using a batched approach [6]. The cancellation filter configuration for each system is shown in Table III.

Each of the implementations was then run over a set of recorded data from an FM based passive radar. While the data set is not representative of the digital transmissions, the execution times of the cancellation algorithms are largely invariant of the underlying data as long as the covariance matrices are non-singular.

For the results, the baselines for comparisons are “QR least-squares,” which is a least-squares solver, and “LU Wiener,” which uses a matrix multiply to form the covariance matrices

TABLE IV: Cancellation filter throughputs for the FM configuration.

Cancellation Filter	Mean throughput (MS/s)
QR least-squares	0.033
CGLS	2.46
LU Wiener	0.564
Rectangular LU Weiner	7.07
FFT LU Weiner	6.37
Toeplitz Weiner	23.7

TABLE V: Cancellation filter throughputs for the DAB configuration.

Cancellation Filter	Mean throughput (MS/s)
QR least-squares	5.40×10^{-3}
CGLS	1.11
LU Wiener	0.118
Rectangular LU Weiner	3.80
FFT LU Weiner	5.01
Toeplitz Weiner	9.91

and an LU factorisation to determine the filter weights. The optimisation in Section III-B is termed “rectangular LU Wiener,” the optimisation scheme presented in Section III-C is termed “FFT LU Wiener,” and the approximation from Section III-E is termed “Toepitz Wiener.”

In order to place the results presented into a more general context, two other methods for performing clutter cancellation are also included in Tables IV to VI. These are the CGLS (implemented with 15 iterations) [2] solver and QR least-squares factorisation.

A. *FM Configuration*

The achieved throughput of the filters for the FM configuration is shown in Table IV.

Aside from the QR least-squares filter, all the filters in Table IV achieved a mean throughput greater than 200 kS/s allowing them to run in real time (i.e. samples were processed at least as fast as they were generated). The rectangular LU Wiener filter achieved the higher throughput than the FFT LU Wiener filter, demonstrating that the overhead incurred using the FFT method outweighs the computational benefits for this configuration.

As expected, the Toeplitz Wiener filter achieved the highest throughput, which was 3.35 times higher than the rectangular LU filter, 42 times higher than the conventional LU Wiener filter, and 9.63 times higher than the CGLS filter.

B. *DAB Configuration*

The results for the DAB filter are shown in Table V.

The increased solution size and data rate meant that only the rectangular Wiener, FFT LU Wiener and Toeplitz Wiener

TABLE VI: Cancellation filter throughputs for the DVB-T configuration.

Cancellation Filter	Mean throughput (MS/s)
CGLS	0.372
Rectangular LU Weiner	1.13
FFT LU Weiner	1.23
Toeplitz Weiner	3.22

filters presented in presented in Section III were able to operate in real-time (mean throughput higher than 2 MS/s). The FFT LU Wiener filter outperformed the rectangular Wiener implementation, demonstrating that the overhead associated with the FFTs is justified by the additional computational benefit for the larger filter size.

Once again, the Toeplitz Wiener filter achieved the highest throughput, 2.6 times higher than the rectangular Wiener filter, 2.0 times higher than the FFT LU Wiener filter, and 84 times higher than the LU Wiener filter.

C. *DVB-T Configuration*

The results for the DVB-T filter are shown in Table VI.

For this configuration, none of the methods which required the explicit storage of the cancellation matrix could be implemented as the memory footprint was larger that the test platform could accommodate; these include QR least-squares and the LU Wiener filters. None of the methods implemented could achieve real-time operation (a mean throughput of at least 8 MS/s), with the fastest throughput achieved being 2.5 times too slow for real-time operation.

Again, the Toeplitz Wiener filter achieved the best throughput, which was 8.7 times higher than the throughput of the CGLS equivalent.

D. *Cancellation effectiveness*

The cancellation effectiveness of sub-optimal cancellation algorithms such as CGLS has already been established in literature [2], [14]. In order to gauge the validity of the Toeplitz approximation, the noise floor for the test data set was compared. The noise floor was estimated by taking the median magnitude of the range-Doppler maps.

Fig. 3 shows the noise floor for the FM configuration, at the batch size where the Toeplitz approximation begins to fail. For the relatively stationary environment for which the data was recorded, this occurred when $N = 2\,000$. With no cancellation, the mean noise floor was -116.0 dBm. The LU Wiener cancellation filter (Wiener in Fig. 3) reduced the noise floor to -155.8 dBm, and the Toeplitz Wiener filter achieved a mean noise floor of -154.8 dBm. In this configuration, the Toeplitz approximation raised the noise floor (i.e. reduced the signal-to-noise ratio (SNR)) by 1.08 dB.

Comparatively, for the original filter length ($N = 20\,000$) for the FM configuration, the noise floor is shown in Fig. 4. Here, the Toeplitz Wiener filter resulted in near identical performance to the LU Wiener filter. The mean noise floor for the data set

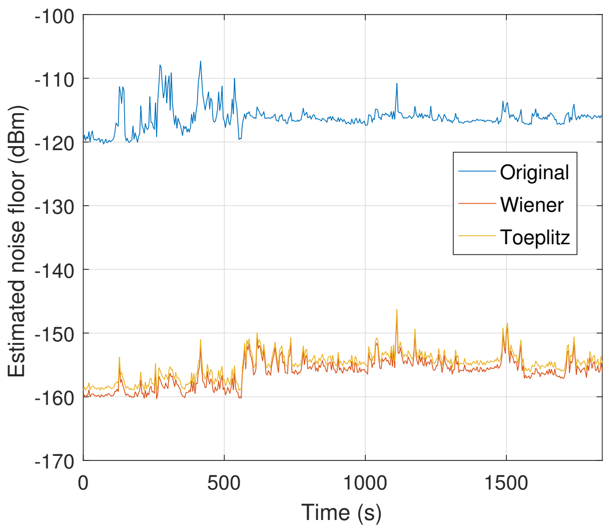

Fig. 3: Noise floor for $M = 512$ and $N = 4\,000$.

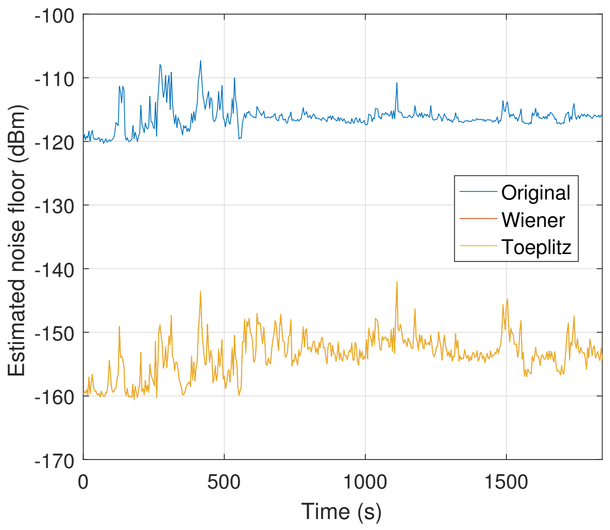

Fig. 4: Noise floor for $M = 512$ and $N = 20\,000$. Note that the Wiener and Toeplitz graphs are indistinguishable.

after the LU Wiener filter was -153.0 dBm, and the Toeplitz Wiener filter displayed a difference of 0.008 dB which is negligible. The Toeplitz approximation would be completely valid in this configuration.

V. Conclusion

Optimised implementations of the required Wiener filtering were presented and evaluated. These implementations exploit the cancellation filters' tapped delay line and overdetermined structure. Two optimisation strategies were presented. The first, rectangular optimisation, was found to be optimal for small problem sizes, while the second, FFT optimisation, was found to be optimal for larger problem sizes. A Hermitian Toeplitz solver was also presented as an approximation to the Wiener filter in order to further accelerate the filters.

The rectangular and FFT approximations were evaluated in three configuration. These filters were found to increase system throughput between 12.5 and 42.5 times. Furthermore, these optimisations allowed a significant decrease in memory requirements. It was shown that for typical passive radar problem sizes, the Toeplitz approximation results in a SNR deterioration of only 1.08 dB and 0.008 dB for FM and DAB passive radars respectively.

Appendix

Hermitian Toeplitz Solver

This section presents the Hermitian Toeplitz solver which is used above. The algorithm is modified from a general Toeplitz solver found in the Toeplitz library [15].

An $M \times M$ Hermitian Toeplitz matrix, $\mathbf{A}$, can be fully described by its first row or column. For this implementation the first row is selected, such that each element of this row is $a_{1,i}$ where $i = 1, 2 \ldots N$. The solution to

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{22}$$

can be computed by Algorithm 1.

In Algorithm 1, vector elements are denoted $x(i)$. The dimension dependant variables required for storage are c_1, c_2, and c_3 which are each $N-1$ element vectors.

By exploiting the Hermitian structure of $\mathbf{A}$, the number of interior multiplications and additions were halved. Furthermore, the loops between 19 and 25 were originally combined (which didn't require an explicit copy), but it was expanded to allow the loop to be parallelised.

Algorithm 1: Hermitian Toeplitz solver

```
Input: $N$ element Toeplitz matrix vector $a$, $N$ element
    solution vector $b$
Output: The $N$ element solution, $x$
    Solve for the first principle minor:
 1: $r_1 = a(1)$
 2: $x(1) = b(1)/r_1$
    Solve for the second principle minor:
 3: $r_6 = a(2)$
 4: $r_3 = -r_6/r_1$
 5: $r_1 = r_1 + r_3 r_6^*$
 6: $c_2(1) = r_3$
 7: $r_6 = (b(2) - a(2)^* x(1))/r_1$
 8: $x(1) = x(1) + c_2(1) r_6$
 9: $x(2) = r_6$
    Solve for the rest of the system:
10: for $i = 3$ to $N$ do
11:     $r_6 = a(i)$
12:     $c_1(i-1) = r_3^*$
13:     for $j = 1$ to $i-2$ do
14:         $r_6 = r_6 + a(j+1)c_2(j)$
15:     end for
16:     $r_3 = -r_6/r_1$
17:     $r_1 = r_1 + r_3 r_6^*$
18:     $c_2(i-1) = 0$
19:     for $j = 1$ to $i-1$ do
20:         $c_3(j) = c_2(j)$
21:     end for
22:     for $j = 2$ to $i-1$ do
23:         $c_2(j) = c_1(j) r_3 + c_3(j-1)$
24:         $c_1(j) = c_1(j) + c_3(j-1) r_3^*$
25:     end for
26:     $c_2(1) = r_3$
27:     $r_5 = 0$
28:     for $j = 1$ to $i-1$ do
29:         $r_5 = r_5 + x(i-j)a(j+1)^*$
30:     end for
31:     $r_6 = (b(i) - r_5)/r_1$
32:     for $j = 1$ to $i-1$ do
33:         $x(j) = x(j) + c_2(j) * r_6$
34:     end for
35:     $x(i) = r_6$
36: end for
37: return $x$
```

References

[1] J. Capon, "Maximum-likelihood spectral estimation," in *Nonlinear Methods of Spectral Analysis*, ser. Topics in Applied Physics, S. Haykin, Ed., vol. 34. Springer Berlin Heidelberg, 1983, pp. 155–179.

[2] J. Palmer and S. Searle, "Evaluation of adaptive filter algorithms for clutter cancellation in passive bistatic radar," in *IEEE Radar Conference (RADAR)*, May 2012, pp. 493–498.

[3] J. R. Guerci, J. S. Goldstein, and I. S. Reed, "Optimal and adaptive reduced-rank stap," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 2, pp. 647–663, Apr 2000.

[4] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-10, no. 6, pp. 853–863, Nov 1974.

[5] B. Kang, V. Monga, and M. Rangaswamy, "Computationally efficient toeplitz approximation of structured covariance under a rank constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 775–785, January 2015.

[6] F. Colone, D. O'Hagan, P. Lombardo, and C. Baker, "A multistage processing algorithm for disturbance removal and target detection in passive bistatic radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 2, pp. 698–722, April 2009.

[7] L. Trefethen and D. Bau, *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[8] Z. Zlatev and H. Nielsen, "Solving large and sparse linear least-squares problems by conjugate gradient algorithms," *Computers & Mathematics with Applications*, vol. 15, no. 3, pp. 185–202, 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0898122188901708

[9] M. A. Richards, J. A. Scheer, and W. A. Holm, "Constant false alarm rate detectors," in *Principles of Modern Radar, Volume I – Basic Principles*. SciTech Publishing, 2010, pp. 589–623.

[10] W. R. Freund, "A look-ahead Bareiss algorithm for general Toeplitz matrices," *Numerische Mathematik*, vol. 68, no. 1, pp. 35–69, 1994.

[11] NVIDIA Corporation, *CUDA C Programming Guide*. NVIDIA Corporation, 2015.

[12] P. Howland, "Editorial: Passive radar systems," *IEE Radar, Sonar and Navigation*, vol. 152, no. 3, pp. 105–106, June 2005.

[13] H. D. Griffiths and C. J. Baker, "Passive coherent location radar systems. Part 1: Performance prediction," *IEE Radar, Sonar and Navigation*, vol. 152, no. 3, pp. 153–159, June 2005.

[14] C. Tong, "A scalable real-time processing chain for radar exploiting illuminators of opportunity," 2014.

[15] O. Arushanian, M. Samarin, V. Voevodin, E. Tyrtyshnikov, B. Garbow, J. Boyle, W. Cowell, and K. Dritz, *The TOEPLITZ Package User's Guide*. Argonne National Laboratory, 1983.