Received 30 March 2023; revised 19 June 2023; accepted 27 June 2023. Date of publication 5 July 2023;
date of current version 28 July 2023. The review of this article was arranged by Associate Editor S. Sun.

Digital Object Identifier 10.1109/OJSP.2023.3292714

Short Paper

Wiener Filter Approximations Without Covariance Matrix Inversion

PRANAV U. DAMALE [1], EDWIN K. P. CHONG [1], AND LOUIS L. SCHARF [2]

[1] Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA
[2] Department of Mathematics, Colorado State University, Fort Collins, CO 80523 USA

CORRESPONDING AUTHOR: PRANAV U. DAMALE (e-mail: pdamale@rams.colostate.edu)

An earlier version of this work has been accepted to be presented at ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

ABSTRACT In this article, we address the problem of ill-conditioning of the Wiener filter, the optimal linear minimum mean square error estimator. Computing the Wiener filter involves the inverse of the observation covariance matrix. In practice, the observation covariance matrix has a large condition number, resulting in unreliable numerical computation of the Wiener filter. To address this issue, we develop four approximate Wiener filter formulas using a truncation technique based on the principal components of a composite covariance matrix. Our approximate filter formulas do not directly involve the inverse of the observation covariance matrix. As a result, our filters are well-conditioned and numerically reliable to compute. We also present an asymptotic analysis of our approximate filter formulas and show that they converge to the Wiener filter as certain approximating terms vanish. Using real data, we evaluate the performance of our filters in terms of accuracy and computation time against the Wiener filter. Our performance-computation tradeoff results show that, unlike the Wiener filter, our filters have stable performance without significantly more computation, even when the covariance matrix is ill-conditioned.

INDEX TERMS Estimation, optimization, random matrix theory, Wiener filters.

I. INTRODUCTION

Estimating a signal of interest based on an observable related signal is of crucial importance. An optimal *estimator* or *filter* (we use these terms interchangeably) is an algorithm that processes the observable signal to yield an estimate such that a certain error criterion between the estimated and desired signal is minimized. The *Wiener filter* is the optimal linear minimum mean square error (LMMSE) estimator. However, computing the Wiener filter involves inverting the observation covariance matrix. In practice, the dimension of the observation signal might be large, or the number of available observation samples might be small relative to the dimension of the observation signal. This results in the observation covariance matrix having a *large condition number*. Computing Wiener filters with ill-conditioned covariance matrices may be numerically unreliable.

We can circumvent the problem of ill-conditioning by approximating the Wiener filter. One approach is by *reconditioning* the covariance matrix using various well-known techniques: ridge regression [2], the minimum eigenvalue method [3], etc. Another approximation approach is to constrain the rank of the Wiener filter to reduce the computation complexity and enhance the performance [4], [5], [6], [7], [8], [9], [10], [11]. However, this does not address the issue of ill-conditioning [12]. More recently, [12] described an alternative approach that explicitly addresses the ill-conditioning instead of constraining the filter rank. The approach described in [12] involves "truncation" based on principal components of a *composite* covariance matrix, leading to two well-conditioned approximate Wiener filters.

To explore the possibility of a more computation-efficient filter without significantly sacrificing performance, in this article, we similarly use truncation to formulate our first two approximate Wiener filters. We approximate our filter formulas even further using the Neumann series expansion and eliminate matrix inversion altogether. The resulting approximate filters are well-conditioned and therefore numerically reliable to compute. The approximations developed here enable a tradeoff between computation and accuracy. We analyze the asymptotic performance of our approximate filters and show that they converge to the Wiener filter as certain approximating terms vanish. To demonstrate the efficacy of our approximation formulas and illustrate the possible computation-accuracy tradeoff, we evaluate the performance of our filters using an empirical dataset: historical daily closing values of the CBOE Amazon VIX Index [13].

The number of principal components of the composite covariance matrix plays a crucial part in truncation and effectively determines the quality of the approximation. To determine the optimal number of principal components used for approximating the filter, [12] uses a line-search procedure to minimize the mean square error. In this article, we use the method of [14], based on results from random matrix theory (RMT), to determine the number of principal components of the covariance matrix. We compare the results for the number of principal components computed using the RMT-based method against the method of [12]. It turns out that the number of principal components, and effectively the performance of our filters, computed using both methods are comparable, while the RMT-based method is more efficient in terms of the computation time than the method of [12].

Even though the practical limitations caused by ill-conditioning are not new to signal processing, and similar problems are faced in other fields, such as statistics and machine learning, only a few known solutions are available, and even fewer solutions that directly address the issue at hand. In signal processing, ill-conditioned covariance matrices can produce filters exhibiting significant errors [15]. In statistics, linear least squares (LLS) problems are often ill-conditioned. In [16], the authors study the stability and accuracy of least squares approximations. In [17], the authors investigate the efficacy of a self-adaptive iterative algorithm for solving severe ill-conditioned LLS problems. In machine learning, ill-conditioning problems occur when dealing with convolutional neural networks [18]. Because the solution to Wiener filters closely relates to the above-mentioned problems, our method of developing well-conditioned filters will be useful for dealing with a variety of ill-conditioned problems.

Our contributions are summarized as follows:

1) To address the problem of ill-conditioning of the Wiener filter, we describe a method based on which we introduce four new approximate Wiener filter formulas that do not directly involve inverting the observation covariance matrix (Section III). We also exploit the peculiar distribution of eigenvalues observed in large covariance matrices using the method of principal component analysis. The approximations developed in this article are justified whenever inverting the observation covariance matrix is ill-conditioned.
2) We prove that the approximate formulas converge to the Wiener filter as certain approximating terms vanish. We also characterize the asymptotic scaling laws (Section IV). Our asymptotic analysis is based on elementary tools accessible to anyone familiar with the application of linear algebra to statistical signal processing problems and will help the reader not accustomed to these types of analyses to develop an intuitive understanding of how to make calculations more tractable without using any esoteric or heavy machinery.
3) We describe two methods for determining the optimal number of principal components used in the approximate filter formulas (Sections V-D). Furthermore, we evaluate how the performance of our approximate filters varies with the number of principal components used for truncation and characterize the tradeoff between the numerical unreliability of filter computation and power loss due to truncation.
4) We evaluate the performance of our approximate Wiener filter formulas using daily closing values of the CBOE Amazon VIX Index (Sections V-E). Our quantitative results with the empirical data demonstrate that the performance of our filters remains stable as we increase the dimension of the associated covariance matrix, while the performance of the Wiener filter deteriorates significantly as expected.

II. THE WIENER FILTER

We wish to estimate a zero-mean random signal $\boldsymbol{X} \in \mathbb{C}^M$ based on a related zero-mean random observation $\boldsymbol{Y} \in \mathbb{C}^N$, where $M, N \in \mathbb{N}$. We use $\mathrm{E}[\cdot]$ as either the expectation or empirical mean from data, $\|\cdot\|$ as the standard complex 2-norm, and $\boldsymbol{A}$ as a matrix representing a linear filter or estimator. Then the LMMSE estimation problem can be stated as follows:

$$\underset{\boldsymbol{A}}{\operatorname{minimize}}\ \mathrm{E}[\|\boldsymbol{A}\boldsymbol{Y} - \boldsymbol{X}\|^2]. \tag{1}$$

The optimal solution, called the Wiener filter, is given as

$$\boldsymbol{A}_{\mathrm{W}} := \boldsymbol{C}_{\boldsymbol{XY}}\boldsymbol{C}_{\boldsymbol{YY}}^{-1}, \tag{2}$$

where $\boldsymbol{C}_{\boldsymbol{YY}} = \mathrm{E}[\boldsymbol{Y}\boldsymbol{Y}'] \in \mathbb{C}^{N\times N}$ is the covariance matrix of the observation $\boldsymbol{Y}$, and $\boldsymbol{C}_{\boldsymbol{XY}} = \mathrm{E}[\boldsymbol{X}\boldsymbol{Y}'] \in \mathbb{C}^{M\times N}$ is the crosscovariance matrix of the desired signal $\boldsymbol{X}$ and the observation $\boldsymbol{Y}$. Throughout this article, we use the superscript prime to denote the Hermitian transpose.

Even though the Wiener filter solves the LMMSE estimation problem, the numerical computation of $\boldsymbol{A}_{\mathrm{W}}$ is unreliable because it involves the inverse $\boldsymbol{C}_{\boldsymbol{YY}}^{-1}$. If the condition number of $\boldsymbol{C}_{\boldsymbol{YY}}$—the ratio of its largest eigenvalue to its smallest eigenvalue—is large, then the computation of $\boldsymbol{C}_{\boldsymbol{YY}}^{-1}$ is numerically unreliable. Notice that we do not need to compute the inverse explicitly. We can simply solve (for $\boldsymbol{A}_{\mathrm{W}}$) the normal equation $\boldsymbol{A}_{\mathrm{W}}\boldsymbol{C}_{\boldsymbol{YY}} = \boldsymbol{C}_{\boldsymbol{XY}}$. Unfortunately, irrespective of the way we compute $\boldsymbol{A}_{\mathrm{W}}$, the reliability of a numerical solution for $\boldsymbol{A}_{\mathrm{W}}$ depends on the condition number of $\boldsymbol{C}_{\boldsymbol{YY}}$. A matrix is *ill-conditioned* if it has a large condition number; otherwise, it is *well-conditioned*. Our primary motivation is to present a solution to the LMMSE estimation problem that is reliable to compute regardless of the condition number of $\boldsymbol{C}_{\boldsymbol{YY}}$.

Now, there are two major factors that cause $\boldsymbol{C}_{\boldsymbol{YY}}$ to have a large condition number. In many practical problems in signal processing, finance, physics, etc., the dimensions of the observation vectors and their corresponding covariance matrices are very large. According to [19], [20], the mean condition number of a random covariance matrix increases with its size. The second reason, which is closely related to the first, is the ratio of the total number of observation samples to the dimension of the observation vector is small (e.g., because the number of observation samples is limited). In such cases, the few largest eigenvalues of the covariance matrix are significantly larger than the remaining eigenvalues [21], [22], [23], [24], [25], [26]. This results in the condition number of the covariance matrix being large as well. Empirical evidence shows that the condition numbers of covariance matrices can easily exceed 10^4 [12], [27]. In practice, these ill-conditioned covariance matrices can cause significant errors when used to solve linear equations.

To address this problem of ill-conditioning, in the next section, we describe approximate Wiener filter formulas that do not directly involve the inverse $\boldsymbol{C}_{\boldsymbol{YY}}^{-1}$. These approximations are justified whenever inverting $\boldsymbol{C}_{\boldsymbol{YY}}$ is ill-conditioned. We introduce multiple filter formulas with varying computational requirements to allow trading off performance for computation. Following that, we show the convergence of these approximate filters to the Wiener filter in terms of their asymptotic scaling laws. Then, we describe two methods for choosing the best approximations to optimally trade off between computation and accuracy. Finally, we evaluate the performance of these filters using empirical data.

III. WIENER FILTER APPROXIMATIONS

To describe our method for approximating the Wiener filter, we start with the method of [12]. First, define $\boldsymbol{Z} \in \mathbb{C}^{(M+N)}$ to be the concatenation of the vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix}. \tag{3}$$

Let $\boldsymbol{C_{ZZ}} \in \mathbb{C}^{(M+N)\times(M+N)}$ be its covariance. Then, the signal $\boldsymbol{X}$ and the observation $\boldsymbol{Y}$ share the composite covariance matrix

$$\boldsymbol{C_{ZZ}} = \begin{bmatrix} \boldsymbol{C_{XX}} & \boldsymbol{C_{XY}} \\ \boldsymbol{C'_{XY}} & \boldsymbol{C_{YY}} \end{bmatrix}. \tag{4}$$

Define the eigendecomposition $\boldsymbol{C_{ZZ}} = \boldsymbol{VSV}'$, where $\boldsymbol{V} \in \mathbb{C}^{(M+N)\times(M+N)}$ is the orthogonal eigenvector matrix composed of eigenvectors of $\boldsymbol{C_{ZZ}}$ as its columns, and $\boldsymbol{S} \in \mathbb{C}^{(M+N)\times(M+N)}$ is the diagonal eigenvalue matrix with diagonal elements $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{M+N}$. The matrices $\boldsymbol{V}$ and $\boldsymbol{S}$ depend jointly on $\boldsymbol{C_{XX}}$, $\boldsymbol{C_{YY}}$, and $\boldsymbol{C_{XY}}$. Next, partition $\boldsymbol{V}$ into $\boldsymbol{V_X}$ (top M rows) and $\boldsymbol{V_Y}$ (bottom N rows). The following identities are easy to verify:

$$\boldsymbol{C_{YY}} = \boldsymbol{V_Y S V'_Y}, \qquad \boldsymbol{C_{XY}} = \boldsymbol{V_X S V'_Y}. \tag{5}$$

The eigendecomposition, or singular-value decomposition (SVD), here is different from the ones usually associated with Wiener filters. For example, in deriving a low-rank version of the Wiener filter, [4] uses the SVD of $\boldsymbol{C_{XY}}\boldsymbol{C_{YY}}^{-1/2}$; see also [7], [8], and [10].

A. TRUNCATION APPROXIMATION

In many real-world problems, such as stock price prediction and direction-of-arrival estimation, we have to deal with large covariance matrices [21], [22], [23], [24], [25], [26], [27]. In these types of problems, the dimension N of $\boldsymbol{Y}$ can be as large as one thousand or even more. The corresponding covariance matrix $\boldsymbol{C_{YY}}$ is of size $N \times N$ and can be transformed using various techniques. The technique of eigendecomposition for transforming a large covariance matrix is of particular interest because the covariance matrix is symmetric by definition, and for a large N, the eigenvalues of covariance matrices vary widely in magnitude. For example, in financial covariance matrices, the first few eigenvalues are often well-separated from the rest of the eigenvalues and account for the bulk of the information, while the remaining eigenvalues decay rapidly toward zero [21], [22], [23], [24], [25], [26], [27]. As the dimension N of $\boldsymbol{Y}$ increases, the proportion of the number of eigenvalues corresponding to this bulk of the information decreases.

As the separation between the first and the last eigenvalue increases, so does the ill-conditioning of the corresponding covariance matrix $\boldsymbol{C_{YY}}$. In such situations, computing a numerical solution in a linear system involving the ill-conditioned $\boldsymbol{C_{YY}}$ becomes unreliable. However, as only the first few eigenvalues carry most of the information, we can safely approximate the relatively small eigenvalues of $\boldsymbol{C_{YY}}$ as 0. To elaborate, suppose the first $L \leq N$ eigenvalues account for the bulk of the eigenvalues of $\boldsymbol{C_{YY}}$; we call these the principal eigenvalues. Next, partition $\boldsymbol{V_X}$, $\boldsymbol{V_Y}$, and $\boldsymbol{S}$ as

$$\boldsymbol{V_X} = \begin{bmatrix} \boldsymbol{V_{X,L}} & \boldsymbol{V_{X,\bar{L}}} \end{bmatrix},$$

$$\boldsymbol{V_Y} = \begin{bmatrix} \boldsymbol{V_{Y,L}} & \boldsymbol{V_{Y,\bar{L}}} \end{bmatrix},$$

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S_L} & \boldsymbol{O}_{L\times(M+N-L)} \\ \boldsymbol{O}_{(M+N-L)\times L} & \boldsymbol{S_{\bar{L}}} \end{bmatrix}, \tag{6}$$

where $\boldsymbol{V_{X,L}} \in \mathbb{C}^{M\times L}$, $\boldsymbol{V_{Y,L}} \in \mathbb{C}^{N\times L}$, $\boldsymbol{S_L} \in \mathbb{C}^{L\times L}$, and $\boldsymbol{O}$ is the all-zero matrix (with dimensions shown in the subscript). Since these L principal eigenvalues constitute the bulk of all the eigenvalues of $\boldsymbol{C_{YY}}$, we can safely approximate $\boldsymbol{S_{\bar{L}}}$ by $\boldsymbol{O}_{(M+N-L)\times(M+N-L)}$, called *truncation* (see, e.g., [7], [12], [27]). Equivalently, we approximate the decomposition by $\boldsymbol{C_{YY}} \approx \boldsymbol{V_{Y,L}S_L V'_{Y,L}}$.

Consider the following transform of $\boldsymbol{Z}$:

$$\tilde{\boldsymbol{Z}} = \boldsymbol{V}'\boldsymbol{Z} = \boldsymbol{V'_X X} + \boldsymbol{V'_Y Y}. \tag{7}$$

Then, the inverse transform is $\boldsymbol{Z} = \boldsymbol{V}\tilde{\boldsymbol{Z}}$, giving $\boldsymbol{X} = \boldsymbol{V_X}\tilde{\boldsymbol{Z}}$ and $\boldsymbol{Y} = \boldsymbol{V_Y}\tilde{\boldsymbol{Z}}$. We now apply the truncation approximation to get the approximate formulas

$$\tilde{\boldsymbol{Z}}_a = \boldsymbol{V'_{X,L}X} + \boldsymbol{V'_{Y,L}Y}, \tag{8}$$

and

$$\boldsymbol{X} = \boldsymbol{V_{X,L}}\tilde{\boldsymbol{Z}}_a. \tag{9}$$

Combining (8) and (9), we get

$$\begin{aligned} &\boldsymbol{X} \approx \boldsymbol{V_{X,L}V'_{X,L}X} + \boldsymbol{V_{X,L}V'_{Y,L}Y} \\ \Rightarrow\ &\boldsymbol{X} \approx (\boldsymbol{I_M} - \boldsymbol{V_{X,L}V'_{X,L}})^{-1}\boldsymbol{V_{X,L}V'_{Y,L}Y}. \end{aligned} \tag{10}$$

This suggests the following filter:

$$\boxed{\boldsymbol{A}_1 = (\boldsymbol{I_M} - \boldsymbol{V_{X,L}V'_{X,L}})^{-1}\boldsymbol{V_{X,L}V'_{Y,L}}.} \tag{11}$$

Notice that $\boldsymbol{C_{YY}^{-1}}$ or its eigenvalues do not appear in (11).

Next, we combine the Neumann series expansion

$$(\boldsymbol{I_M} - \boldsymbol{V_{X,L}V'_{X,L}})^{-1} = \sum_{k=0}^{\infty}(\boldsymbol{V_{X,L}V'_{X,L}})^k \tag{12}$$

with the identity $(\boldsymbol{V_{X,L}V'_{X,L}})^k\boldsymbol{V_{X,L}} = \boldsymbol{V_{X,L}}(\boldsymbol{V'_{X,L}V_{X,L}})^k$ to get $(\boldsymbol{I_M} - \boldsymbol{V_{X,L}}\boldsymbol{V'_{X,L}})^{-1}\boldsymbol{V_{X,L}} = \boldsymbol{V_{X,L}}(\boldsymbol{I_L} - \boldsymbol{V'_{X,L}V_{X,L}})^{-1}$. This leads to an alternative filter formula:

$$\boxed{\boldsymbol{A}_2 = \boldsymbol{V_{X,L}}(\boldsymbol{I_L} - \boldsymbol{V'_{X,L}V_{X,L}})^{-1}\boldsymbol{V'_{Y,L}}.} \tag{13}$$

Though (11) and (13) are equivalent formulas, their computational burden might differ. We investigate this difference later.

The filters $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are similar to the LSJPC filter in [12], which was derived using a similar truncation operation and is well-conditioned. The LSJPC filter was shown to have stable performance, even when the covariance matrix is ill-conditioned [12]. This indicates propitious performance for $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$.

B. INVERSE APPROXIMATION

Though we do not limit ourselves to any particular matrix norm, we assume that the matrix norm is submultiplicative. Many matrix norms, such as the 1-norm, the infinity-norm, and the Frobenius norm, have the submultiplicative property. However, there are some other matrix norms, such as the max norm, that do not possess the submultiplicative property. Furthermore, we assume that $\|\boldsymbol{I_M}\| = 1$ (otherwise we need to normalize by $\|\boldsymbol{I_M}\|$). Considering $\boldsymbol{V_{X,L}V'_{X,L}} + \boldsymbol{V_{X,\bar{L}}V'_{X,\bar{L}}} = \boldsymbol{I_M}$, we see that $\|\boldsymbol{V_{X,L}V'_{X,L}}\| < 1$. This means that $\|\boldsymbol{V_{X,L}V'_{X,L}}\|^k$ can be made arbitrarily small with sufficiently large k. Hence, we can approximate the Neumann series expansion

$$(\boldsymbol{I}_M - \boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L})^{-1} = \sum_{k=0}^{\infty}(\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L})^k \tag{14}$$

by a finite sum:

$$(\boldsymbol{I}_M - \boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L})^{-1} \approx \sum_{k=0}^{K}(\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L})^k. \tag{15}$$

Applying this approximation to (11), we get the filter

$$\boxed{\boldsymbol{A}_3 = \left(\sum_{k=0}^{K}(\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L})^k\right)\boldsymbol{V}_{X,L}\boldsymbol{V}'_{Y,L}.} \tag{16}$$

Similarly, from (13), we get

$$\boxed{\boldsymbol{A}_4 = \boldsymbol{V}_{X,L}\left(\sum_{k=0}^{K}(\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L})^k\right)\boldsymbol{V}'_{Y,L}.} \tag{17}$$

Finally, note that computing the estimate $\hat{\boldsymbol{X}} = \boldsymbol{A}_3\boldsymbol{Y}$ based on (16) can be done iteratively using

$$\begin{aligned}\hat{\boldsymbol{X}}(k+1) &= (\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L})\hat{\boldsymbol{X}}(k) + \boldsymbol{V}_{X,L}\boldsymbol{V}'_{Y,L}\boldsymbol{Y},\\ k &= 0, 1, \ldots, K-1, \quad \hat{\boldsymbol{X}}(0) = \boldsymbol{0}.\end{aligned} \tag{18}$$

Because $\|\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L}\| < 1$, (18) is a contractive fixed-point iteration, guaranteeing convergence of $\hat{\boldsymbol{X}}$ to (11) as $K \to \infty$. Similarly, computing the estimate $\boldsymbol{A}_4\boldsymbol{Y}$ based on (17) can be done iteratively using

$$\begin{aligned}\boldsymbol{U}(k+1) &= (\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L})\boldsymbol{U}(k) + \boldsymbol{V}'_{Y,L}\boldsymbol{Y},\\ k &= 0, 1, \ldots, K-1, \quad \boldsymbol{U}(0) = \boldsymbol{0},\\ \hat{\boldsymbol{X}} &= \boldsymbol{V}_{X,L}\boldsymbol{U}(K).\end{aligned} \tag{19}$$

We use the notation $\boldsymbol{A}_3(K)$ (or $\boldsymbol{A}_4(K)$) to indicate the Kth Neumann-approximation order for the filter $\boldsymbol{A}_3$ (or $\boldsymbol{A}_4$, respectively). The special case of $K = 0$ gives the simple formula $\boldsymbol{V}_{X,L}\boldsymbol{V}'_{Y,L}$, which is difficult to imagine simplifying any further.

In the cases where $\boldsymbol{C}_{YY}$ is ill-conditioned and its eigenvalues decrease very quickly, it is possible to find an appropriate L for truncation. Similar truncations have been used in low-rank versions of the Wiener filter [4], [5], [7], [8], [10], [12], [28]. Naturally, in choosing L there is a tradeoff between the quality of the approximation and the computational burden. Therefore, there is no universally applicable value for L. Indeed, this tradeoff affords flexibility in implementing filters under different computational constraints.

It is not yet entirely clear how well our approximate filters perform in practice. Our derivations for the approximate filters described above are found in simple heuristics. To ensure the viability of these filters, we need to carefully analyze and evaluate them further. In Section IV, we show how these approximations converge to $\boldsymbol{A}_{\mathrm{W}}$ as the truncation and inverse approximations become exact. Later, in Section V, we evaluate these formulas using empirical data.

Our approach to addressing the problem of ill-conditioning involves dimension reduction. An added benefit of dimension reduction is that we can expect the computational burden of applying the filters to be reduced relative to the Wiener filter. Our empirical results in Sections V-E corroborate this expectation.

IV. ASYMPTOTIC PERFORMANCE

We now show how each of the approximation formulas derived in Section III converges to $\boldsymbol{A}_{\mathrm{W}}$ using the *Bachmann-Landau* notation $\mathcal{O}(\cdot)$. To explain, consider a matrix $\boldsymbol{M}(\rho)$ that depends on some parameter $\rho \to 0$. If, for some c, $\|\boldsymbol{M}(\rho)\| \leq c\rho$ for sufficiently small ρ, then we write $\boldsymbol{M}(\rho) = \mathcal{O}(\rho)$. If $\boldsymbol{C}$ and $\boldsymbol{D}$ are bounded as $\rho \to 0$, then the following algebraic rules help us to simplify the calculations:

$$\boldsymbol{C}\mathcal{O}(\rho) = \mathcal{O}(\rho), \tag{20}$$

$$(\boldsymbol{C} + \mathcal{O}(\rho))^{-1} = \boldsymbol{C}^{-1} + \mathcal{O}(\rho), \text{ and} \tag{21}$$

$$(\boldsymbol{C} + \mathcal{O}(\rho))(\boldsymbol{D} + \mathcal{O}(\rho)) = \boldsymbol{C}\boldsymbol{D} + \mathcal{O}(\rho). \tag{22}$$

Whenever we use the truncation operation for some values of L and K, we essentially lose some of the power in the observed signal $\boldsymbol{Y}$. We measure this *truncation-power loss* as ρ_L. The amount of power lost due to truncation is small by design. The parameter ρ_L is a measure of this amount of power being discarded. The exact expression for the power being discarded depends on the filter. For $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, $\rho_L = \|\boldsymbol{S}_{\bar{L}}\|$. For $\boldsymbol{A}_3$, $\rho_L = \|\boldsymbol{S}_{\bar{L}}\| + \|\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L}\|^{K+1}$. And, for $\boldsymbol{A}_4$, $\rho_L = \|\boldsymbol{S}_{\bar{L}}\| + \|\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L}\|^{K+1}$. The truncation approximation relies on $\|\boldsymbol{S}_{\bar{L}}\| \approx 0$. Moreover, the inverse approximation relies on either $\|\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L}\| \approx 0$ or $\|\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L}\| \approx 0$. Thus, in our analysis, we consider $\boldsymbol{S}_{\bar{L}}$ and $\boldsymbol{V}_X$ to be variables such that $\|\boldsymbol{S}_{\bar{L}}\| \to 0$, $\|\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L}\| \to 0$, and $\|\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L}\| \to 0$. We treat $\boldsymbol{S}_L$ as fixed. Note that $\boldsymbol{V}_X$ and $\boldsymbol{V}_Y$, and their submatrices $\boldsymbol{V}_{X,L}$ and $\boldsymbol{V}_{Y,L}$, are bounded as $\|\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L}\| \to 0$ and $\|\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L}\| \to 0$. This is relevant for applying the Bachmann-Landau rules.

Theorem 1: Given a positive integer $L \leq N$,

$$\boldsymbol{A}_{\mathrm{W}} - \boldsymbol{A}_2 = \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|). \tag{23}$$

Proof: We first write

$$\begin{aligned}\boldsymbol{C}_{XY} &= \boldsymbol{V}_X\boldsymbol{S}\boldsymbol{V}'_Y = \boldsymbol{V}_{X,L}\boldsymbol{S}_L\boldsymbol{V}'_{Y,L} + \boldsymbol{V}_{X,\bar{L}}\boldsymbol{S}_{\bar{L}}\boldsymbol{V}'_{Y,\bar{L}}\\ &= \boldsymbol{V}_{X,L}\boldsymbol{S}_L\boldsymbol{V}'_{Y,L} + \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|)\\ \boldsymbol{C}_{YY} &= \boldsymbol{V}_Y\boldsymbol{S}\boldsymbol{V}'_Y = \boldsymbol{V}_{Y,L}\boldsymbol{S}_L\boldsymbol{V}'_{Y,L} + \boldsymbol{V}_{Y,\bar{L}}\boldsymbol{S}_{\bar{L}}\boldsymbol{V}'_{Y,\bar{L}}\\ &= \boldsymbol{V}_{Y,L}\boldsymbol{S}_L\boldsymbol{V}'_{Y,L} + \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|).\end{aligned} \tag{24}$$

Therefore, using the Bachmann-Landau rules,

$$\begin{aligned}\boldsymbol{A}_{\mathrm{W}} &= \boldsymbol{V}_{X,L}\boldsymbol{S}_L\boldsymbol{V}'_{Y,L}(\boldsymbol{V}_{Y,L}\boldsymbol{S}_L\boldsymbol{V}'_{Y,L})^{-1} + \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|)\\ &= \boldsymbol{V}_{X,L}(\boldsymbol{V}'_{Y,L}\boldsymbol{V}_{Y,L})^{-1}(\boldsymbol{V}'_{Y,L}\boldsymbol{V}_{Y,L})\boldsymbol{S}_L\boldsymbol{V}'_{Y,L}\\ &\quad(\boldsymbol{V}_{Y,L}\boldsymbol{S}_L\boldsymbol{V}'_{Y,L})^{-1} + \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|)\\ &= \boldsymbol{V}_{X,L}(\boldsymbol{V}'_{Y,L}\boldsymbol{V}_{Y,L})^{-1}\boldsymbol{V}'_{Y,L} + \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|)\\ &= \boldsymbol{V}_{X,L}(\boldsymbol{I}_L - \boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L})^{-1}\boldsymbol{V}'_{Y,L} + \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|),\end{aligned} \tag{25}$$

which completes the proof. ■

Corollary 1: Given a positive integer $L \leq N$,

$$\boldsymbol{A}_{\mathrm{W}} - \boldsymbol{A}_1 = \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|). \tag{26}$$

Theorem 2: Given positive integers $L \leq N$ and K,

$$\boldsymbol{A}_{\mathrm{W}} - \boldsymbol{A}_4 = \mathcal{O}(\|\boldsymbol{S}_{\bar{L}}\|) + \mathcal{O}(\|\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L}\|^{K+1}). \tag{27}$$

Proof: Using the Neumann series expansion

$$\begin{aligned}(\boldsymbol{I}_L - \boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L})^{-1} &= \sum_{k=0}^{\infty}(\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L})^k \\ &= \sum_{k=0}^{K}(\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L})^k \\ &\quad + \mathcal{O}(\|\boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L}\|^{K+1}). \qquad (28)\end{aligned}$$

Combining this with (23), we get the desired result.

Corollary 2: Given positive integers $L \le N$ and K,

$$\boldsymbol{A}_{\mathrm{W}} - \boldsymbol{A}_3 = \mathcal{O}(\|\boldsymbol{S}_{\overline{L}}\|) + \mathcal{O}(\|\boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L}\|^{K+1}). \qquad (29)$$

The results above, stated in terms of the Bachmann-Landau notation, provide only an asymptotic characterization of the difference between each filter and $\boldsymbol{A}_{\mathrm{W}}$, not an exact expression for it. However, inspecting the calculations above gives some idea of how impractical an exact analysis would be.

V. RESULTS AND DISCUSSION

We now compare, in terms of accuracy and computation time with real data, the performance of the different approximate filters developed in Section III relative to the Wiener filter. Our performance-complexity evaluation has two parts. First, we test the approximate filters as we vary the conditioning of the observation covariance matrix. Second, we quantify the price paid for well conditioning of $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$, and $\boldsymbol{A}_4$ in the preprocessing steps relative to the Wiener filter. We expect the performance of the Wiener filter to deteriorate as the conditioning of the associated observation covariance matrix worsens while the performance of the approximate filters does not deteriorate. Further, we expect the approximate filters to be computationally more efficient than the Wiener filter because of the reduced dimensions involved. We also describe and compare two methods for selecting the optimal filter orders. The results from the computational comparison elucidate how the complexities of the approximate filters scale with different dimensions of the associated covariance matrices and approximation orders. Although some of the specific numerical values are data-dependent, they illustrate the general qualitative features of our methods.

A. DATASET

To demonstrate the performance-complexity tradeoff of each of our approximate filters relative to the Wiener filter, we use the CBOE Amazon VIX index (VXAZN) daily closing values. The historical VXAZN data used in this article is freely available at [13]. To check for stationarity in time of the VXAZN data, following the method in [29], [30], we implemented the augmented Dickey-Fuller (ADF) test [31] for up to 40 lags. The ADF test checks for the null hypothesis that the process follows a unit root, i.e., it is non-stationary. Our ADF test results for the VXAZN data, produced using the HypothesisTests package in Julia [32], strongly reject the null hypothesis with a maximum p-value of 0.0006. Therefore, we can safely assume that the VXAZN data sequence is stationary.

B. DATA PROCESSING

For the purpose of our estimation problem, we use a sequence of 2974 daily closing values of the VXAZN dataset, starting on January 7, 2011, and ending on November 4, 2022. Our estimation problem can be stated as follows: given the VXAZN values for N consecutive days, predict the values for M consecutive days immediately after. To do this, we consider vector-valued samples of $M+N$ consecutive days. We get a total of $T = 2974-(M+N)+1$ such vector-valued samples, each one shifted by one lag from the previous one. We center the data by subtracting the empirical mean of the data matrix from each sample. We then construct a data matrix

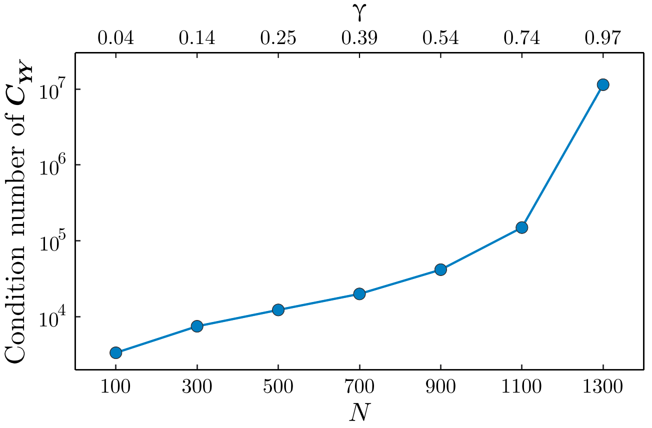

FIGURE 1. Condition number of C_{YY} vs. N and γ.

$$\begin{aligned}\boldsymbol{W} &= \begin{bmatrix}\boldsymbol{W}_1 & \boldsymbol{W}_2 & \cdots & \boldsymbol{W}_T\end{bmatrix} \\ &= \begin{bmatrix} W(1) & W(2) & \cdots & W(T) \\ W(2) & W(3) & \cdots & W(T+1) \\ \vdots & \vdots & \ddots & \vdots \\ W(M+N) & W(M+N+1) & \cdots & W(2974)\end{bmatrix}, \qquad (30)\end{aligned}$$

where $W(i)$ represents the centered closing value of the VXAZN dataset for the day i. Because the dataset is stationary, the columns of $\boldsymbol{W}$ have a common $(M+N)$-variate mean and covariance. Then, we compute the sample covariance matrix $\boldsymbol{C_{WW}}$ as

$$\boldsymbol{C_{WW}} = \frac{1}{T-1}\sum_{i=1}^{T}\boldsymbol{W}_i\boldsymbol{W}'_i. \qquad (31)$$

We use 80% of the data samples for computing the sample covariance matrix and reserve the remaining 20% for conducting out-of-sample test experiments. We obtain the 80% above randomly from the data matrix.

C. ILL-CONDITIONING OF C_{YY}

Our aim for the performance-complexity analysis is to test how the different approximate filters derived in Section III perform relative to the Wiener filter when the conditioning of the observation covariance matrix is varied. To test this, we fix $M = 7$ and vary N from 100 to 1300. Because the total number of data values is fixed (2974 in our case), as we increase the dimension of our vector-valued data sample, the total number of available data samples T decreases. We use $\gamma = (M+N)/T$ to denote their ratio. As the dimension of the data sample approaches the total number of available data samples (i.e., $\gamma \to 1$), the resulting covariance matrix becomes more ill-conditioned. Fig. 1 shows how the condition number of $\boldsymbol{C_{YY}}$ changes as $\gamma \to 1$. We can see that, as γ rises from 0.04 to 0.97, the resulting condition number

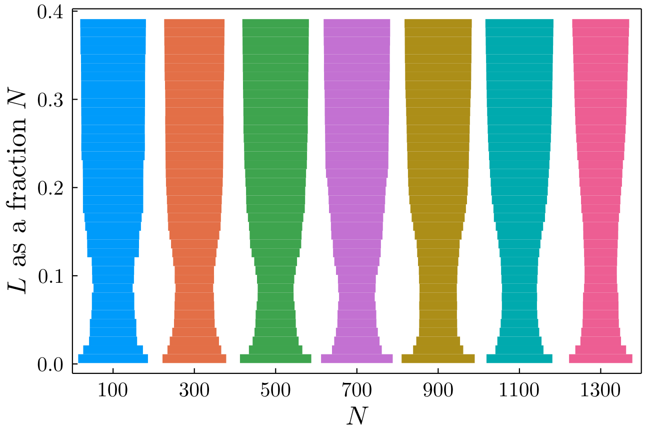

FIGURE 2. **Change in filter performance as *L* varies.**

of $\boldsymbol{C}_{YY}$ increases by more than three orders of magnitude. This allows performance evaluation across many scenarios.

To keep the performance metric directly comparable as we vary N, we evaluate the performance according to the normalized root-mean-square error (nRMSE), calculated as

$$\sqrt{\frac{1}{T}\sum_{i=1}^{T}\|\boldsymbol{A}\boldsymbol{W}_i(\boldsymbol{Y})-\boldsymbol{W}_i(\boldsymbol{X})\|^2}\bigg/\sqrt{\frac{1}{T}\sum_{i=1}^{T}\|\boldsymbol{W}_i(\boldsymbol{X})+\overline{\boldsymbol{W}}\|^2},$$

where $\boldsymbol{W}_i \in \mathbb{R}^{(M+N)}$ $(i = 1, \ldots, T)$ is the vector-valued sample, $\boldsymbol{W}_i(\boldsymbol{Y})$ is the N-subvector of $\boldsymbol{W}_i$ corresponding to $\boldsymbol{Y}$, $\boldsymbol{W}_i(\boldsymbol{X})$ is the M-subvector corresponding to $\boldsymbol{X}$, and $\overline{\boldsymbol{W}}$ is the average M-vector. Better performance values are closer to 0.

D. BEST *L*

The performance of $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$ varies with the number of principal components L used for truncation. To illustrate the effect of different values of L on the performance, Fig. 2 depicts the performance of $\boldsymbol{A}_3(5)$ for different values of N as we vary L. Here, the thickness of each column rung for different values of N indicates the performance in terms of nRMSE, with the maximum thickness of a column rung showing an nRMSE value of 0.28. We can see that the performance of $\boldsymbol{A}_3(5)$ first improves as we increase L and then deteriorates as we keep increasing L. Clearly, there is some optimal value of L for which the filter performs best. This optimal value is governed by the tradeoff between the numerical unreliability of filter computation and the truncation-power loss as we increase L. The filter performance is more sensitive to this optimal value of L for smaller values of N, but for larger values of N, we can vary L by a relatively wide margin without affecting its performance by much. The performance of $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$ also varies with the Neumann-approximation order. In our experiments, their performance is essentially constant beyond $K = 5$. Even when we change the dimension of M, $K = 5$ is the smallest Neumann-approximation order of $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$ that matches the best performance of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, beyond which there is little to separate between their performance.

Next, we describe two methods for computing the best L value to get the optimal filter performance.

1) LINE-SEARCH METHOD

Our objective function value, the mean square error, for any given $\boldsymbol{A}$ can also be expressed as

$$\mathrm{E}[\|\boldsymbol{A}\boldsymbol{Y}-\boldsymbol{X}\|^2] = \mathrm{tr}(\boldsymbol{C}_{XX} - 2\boldsymbol{C}_{XY}\boldsymbol{A}' + \boldsymbol{A}\boldsymbol{C}_{YY}\boldsymbol{A}'). \tag{32}$$

Following the method in [12], we can find a suitable value of L for any filter $\boldsymbol{A}$ using a simple line-search (LS) procedure [28] together with the mean-square-error formula in (32).

The LS method for choosing the best L is computationally costly because we need to compute $\boldsymbol{A}$ multiple times for different values of L.

2) MARCHENKO-PASTUR METHOD

Following the method in [14], we can find the best L from the data matrix without the explicit need for computing $\boldsymbol{A}$. Recall that $\boldsymbol{W} \in \mathbb{R}^{(M+N)\times T}$ is the mean-subtracted data matrix and $\boldsymbol{C}_{WW}$ is its covariance matrix. Without loss of generality, we assume $(M+N) < T$. If $\boldsymbol{W}$ is a random matrix, i.e., $\boldsymbol{W}$ has independent and identically distributed (i.i.d.) entries with mean 0 and variance σ^2, we can view the eigenvalues $\lambda_1, \ldots, \lambda_{M+N}$ of $\boldsymbol{C}_{WW}$ as random variables. Then, in agreement with an asymptotic universal law from RMT, the eigenvalues of $\boldsymbol{C}_{WW}$ have the Marchenko-Pastur (MP) probability density function [33], which is fully characterized by the dimensions of the data matrix ($(M+N)$ and T) and the variance (σ^2). According to [34], if $\boldsymbol{C}_{WW}$ has a non-zero number of eigenvalues that are well-separated from the rest of the eigenvalues, then we can infer that the entries of $\boldsymbol{W}$ are not i.i.d. According to [14], [34], [35], [36], [37], [38], if a data matrix can be synthesized by $L < (M+N)$ principal components, then the MP distribution still applies to the remaining $M+N-L$ eigenvalues of its covariance matrix. In many real-world problems, the largest L eigenvalues, corresponding to the principal components of the covariance matrix, are well-separated from the remaining eigenvalues. The latter are packed together within the support of the MP density [34], [38], [39], [40], [41], [42], [43], [44], [45], [46]. For such covariance matrices, we can jointly estimate L and σ^2 by determining the minimum value of L such that the λ_{L+1} to λ_{M+N} eigenvalues are well described by the MP distribution. Following the method in [14], we can easily find the minimum L that satisfies the inequality

$$\frac{\lambda_{L+1}-\lambda_{M+N}}{4\sqrt{(M+N-L)/T}} \le \frac{1}{(M+N-L)}\sum_{i=L+1}^{M+N}\lambda_i. \tag{33}$$

This method works because if at least one of the eigenvalues corresponding to the principal components is part of the eigenvalues grouped together in the support of the MP density, then the range (left-hand side of (33)) will exceed the mean (right-hand side of (33)). Once we identify the minimum L for which the test inequality in (33) is satisfied, we can substitute $\lambda_{L+1}, \ldots, \lambda_{M+N}$ with 0 and approximate the Wiener filter using the formulas derived in Section III.

Fig. 3 shows the best L values for $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(5)$ as we vary N, calculated using the LS and MP methods. Because the LS method depends on the individual filter, we need to calculate the best L values for $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(5)$ separately. In contrast, the MP method is independent of the filter and only depends on the data matrix. From Fig. 3, we can see that the best L values for both the LS and MP methods are relatively close to each other.

To better differentiate these methods, we analyze the computation time required for each in calculating the best L values. Table 1 shows the computation times for the LS and MP methods for calculating

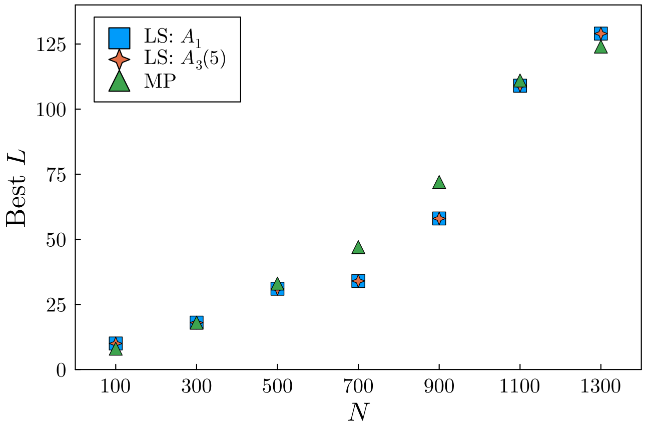

FIGURE 3. Best *L* value vs. *N*.

TABLE 1. Computation Time for the Line Search (LS) Method and the Marchenko-Pastur (MP) Method for Calculating the Best *L* as *N* Varies

N	LS:$\boldsymbol{A}_1$ Time (ms)	LS: $\boldsymbol{A}_3(5)$ Time (ms)	MP Time (ms)
100	3.61	4.12	0.0039
300	17.1	17.9	0.0101
500	48.8	50.0	0.0256
700	91.5	92.4	0.0352
900	161.0	163.0	0.0930
1100	252.0	253.0	0.1305
1300	383.0	386.0	0.1655

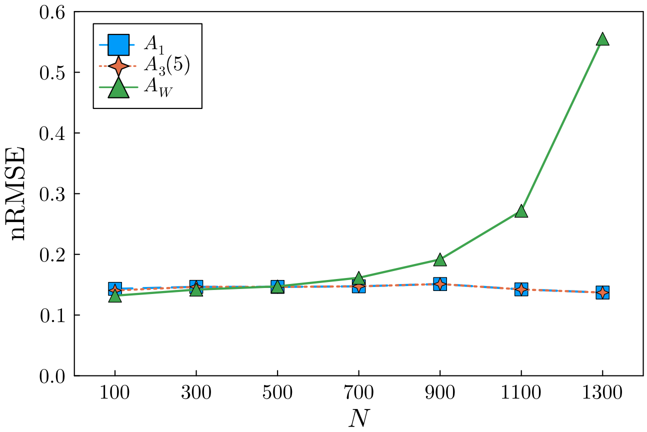

FIGURE 4. nRMSE vs. *N*.

the best L value. We can see that the MP method is approximately three orders of magnitude faster than the LS method. As seen earlier in Fig. 3, the optimal L values computed using both of these methods are comparable. Therefore, henceforth we will use the MP method for computing the optimal L value without compromising the filter performance.

E. PERFORMANCE-COMPLEXITY TRADEOFF

Fig. 4 compares the performance of $\boldsymbol{A}_1$, $\boldsymbol{A}_3(5)$, and $\boldsymbol{A}_{\mathrm{W}}$ in terms of nRMSE as N increases and $\boldsymbol{C_{YY}}$ becomes more ill-conditioned (see Fig. 1). For $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(5)$, we used the best L values computed according to the MP method. All filters have comparable performance for smaller values of N, or until the total number of samples is more than double the dimension of the data vector. After $N = 700$, when the ill-conditioning of $\boldsymbol{C_{YY}}$ starts to increase rapidly, the computation of $\boldsymbol{A}_{\mathrm{W}}$ becomes more unreliable. This is evidenced by the deteriorating performance of $\boldsymbol{A}_{\mathrm{W}}$ after $N = 700$. But the same is untrue for filters $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(5)$. The nRMSE values for $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(5)$ never exceed 0.15. This demonstrates that our filters have consistently stable performance even when $\boldsymbol{C_{YY}}$ is ill-conditioned.

TABLE 2. Condition Numbers of Matrices Involved in Computing A_{W}, A_1, and A_2, Respectively

N	Cond. no. of $\boldsymbol{C_{YY}}$	Cond. no. of $\boldsymbol{I}_M - \boldsymbol{V}_{\boldsymbol{X},L}\boldsymbol{V}'_{\boldsymbol{X},L}$	Cond. no. of $\boldsymbol{I}_L - \boldsymbol{V}'_{\boldsymbol{X},L}\boldsymbol{V}_{\boldsymbol{X},L}$
100	3349.86	7.35	7.35
300	7484.85	2.53	2.53
500	12289.47	2.53	2.53
700	19972.04	2.50	2.50
900	41678.75	4.07	4.07
1100	149494.81	5.38	5.38
1300	1.13e+07	10.79	11.05

We also implemented the recursive least squares (RLS) algorithm with a growing window, which is a well-known adaptive filter and employs an iterative approach for filter computation. However, the resulting RLS filter converges to $\boldsymbol{A}_{\mathrm{W}}$ and has the same deteriorating performance as the Wiener filter after $N = 700$, as illustrated in Fig. 4.

Note that, although there are matrices that involve inverses in computing filters $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ (namely $\boldsymbol{I}_M - \boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L}$ and $\boldsymbol{I}_L - \boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L}$), these matrices are well-conditioned. Table 2 compares the condition numbers of these matrices in computing filters $\boldsymbol{A}_{\mathrm{W}}$, $\boldsymbol{A}_1$, and $\boldsymbol{A}_2$. We can see that, as we increase N from 100 to 1300, the resulting condition number of $\boldsymbol{C_{YY}}$ increases by more than three orders of magnitude while the condition numbers of matrices $\boldsymbol{I}_M - \boldsymbol{V}_{X,L}\boldsymbol{V}'_{X,L}$ and $\boldsymbol{I}_L - \boldsymbol{V}'_{X,L}\boldsymbol{V}_{X,L}$ remain relatively small, which shows that these matrices are well-conditioned.

Using the approximated filters $\boldsymbol{A}_1$ to $\boldsymbol{A}_4$ requires some preprocessing, which includes computing the eigendecomposition of the covariance matrix and selecting the best L using the LS or MP methods. Indeed, this preprocessing involves some computational costs. Fortunately, the preprocessing has to be done upfront only once, after which we can repeatedly use the formulas for $\boldsymbol{A}_1$ to $\boldsymbol{A}_4$ with very little extra computation. i.e., once we obtain $\boldsymbol{V}_{X,L}$ and $\boldsymbol{V}_{Y,L}$, computing the approximate filters $\boldsymbol{A}_1$ to $\boldsymbol{A}_4$ is more efficient than computing $\boldsymbol{A}_{\mathrm{W}}$ due to the reduced dimensions of $\boldsymbol{V}_{X,L}$ and $\boldsymbol{V}_{Y,L}$. Table 3 shows the computation times for $\boldsymbol{A}_1$ with and without the preprocessing. We can see that the computation times for $\boldsymbol{A}_1$ with the preprocessing are greater than that of $\boldsymbol{A}_{\mathrm{W}}$, while the computation times for $\boldsymbol{A}_1$ without the preprocessing are two orders of magnitude smaller than $\boldsymbol{A}_{\mathrm{W}}$. These results show that $\boldsymbol{A}_1$ is computationally more expensive than $\boldsymbol{A}_{\mathrm{W}}$ when we need to use the filters just once, but their marginal computational burden quickly diminishes over multiple filtering operations. In our experiments, it never takes more than 7 applications of $\boldsymbol{A}_1$ to break even with $\boldsymbol{A}_{\mathrm{W}}$. As shown later, the computation times for $\boldsymbol{A}_2$ to $\boldsymbol{A}_4$ are comparable to $\boldsymbol{A}_1$. Because the best L values are approximately 10% of the corresponding N values, computing the approximate filters post-processing is more efficient compared to

TABLE 3. Computation Time for $\boldsymbol{A}_1$ and $\boldsymbol{A}_\mathrm{W}$ as N Varies

N	$\boldsymbol{A}_1$ w/o preprocessing Time (ms)	$\boldsymbol{A}_1$ w/ preprocessing Time (ms)	$\boldsymbol{A}_\mathrm{W}$ Time (ms)	# of applications of $\boldsymbol{A}_1$ to break even
100	0.0039	3.0095	0.4578	7
300	0.0102	16.0639	2.4300	7
500	0.0257	38.4283	5.7700	7
700	0.0351	77.9418	13.900	6
900	0.0859	131.4415	24.000	6
1100	0.1329	209.5738	36.700	6
1300	0.1723	326.5040	56.300	6

TABLE 4. Computation Time for All Approximate Filters as N Varies

N	$\boldsymbol{A}_1$ Time (ms)	$\boldsymbol{A}_2$ Time (ms)	$\boldsymbol{A}_3(5)$ Time (ms)	$\boldsymbol{A}_4(5)$ Time (ms)
100	0.0039	0.0048	0.0300	0.0387
300	0.0102	0.0148	0.0364	0.1011
500	0.0257	0.0419	0.0520	0.3463
700	0.0351	0.0507	0.0608	0.3902
900	0.0859	0.1489	0.1236	1.6000
1100	0.1329	0.7079	0.167	8.4700
1300	0.1723	0.8610	0.1972	11.500

$\boldsymbol{A}_\mathrm{W}$. It is worth emphasizing that our primary purpose is to address the ill-conditioning of $\boldsymbol{C}_{YY}$. The computation times in Table 3 along with the nRMSE performance of the filters in Fig. 4 demonstrate that our filters are numerically reliable to compute without significantly increasing the computational burden (and even decreasing it in some use cases).

Finally, Table 4 compares the computation times for all four approximate filters as we increase N. Table 4 does not include the times to compute eigendecompositions and the MP method to select L. Even though $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(5)$ are algebraically equivalent to $\boldsymbol{A}_2$ and $\boldsymbol{A}_4(5)$, respectively, they differ in terms of their computational burden. We can attribute this added computation to the extra floating point operations (FLOPs) needed to compute $\boldsymbol{A}_2$ and $\boldsymbol{A}_4(5)$ compared to their respective algebraic counterparts. The number of FLOPs for $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(5)$ increases with M, while the number of FLOPs for $\boldsymbol{A}_2$ and $\boldsymbol{A}_4(5)$ increases with only L. For smaller values of N, the values of L and M are comparable. This is demonstrated by their comparable computation times for small values of N. As N increases, so does L, whereas M remains constant throughout. Therefore, for large values of N, the computation times for $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(4)$ are much lower than their respective counterparts. This ordering would reverse whenever $L < M$. The perceptive reader will have noticed that we can easily estimate how the computation cost for the approximate filters would change as M increases by comparing the computation costs for $\boldsymbol{A}_1$ and $\boldsymbol{A}_3$ with those of $\boldsymbol{A}_2$ and $\boldsymbol{A}_4$. The difference between $\boldsymbol{A}_1$ and $\boldsymbol{A}_3(5)$ remains relatively small for all values of N, as illustrated more clearly in Fig. 5. With additional order of approximation (meaning additional computation), $\boldsymbol{A}_3$ becomes more like $\boldsymbol{A}_1$. But recall that computing $\boldsymbol{A}_1$ involves an inversion of a matrix, albeit of a smaller size, whereas computing $\boldsymbol{A}_3$ does not involve inversion of any kind. This tradeoff is beneficial in many cases.

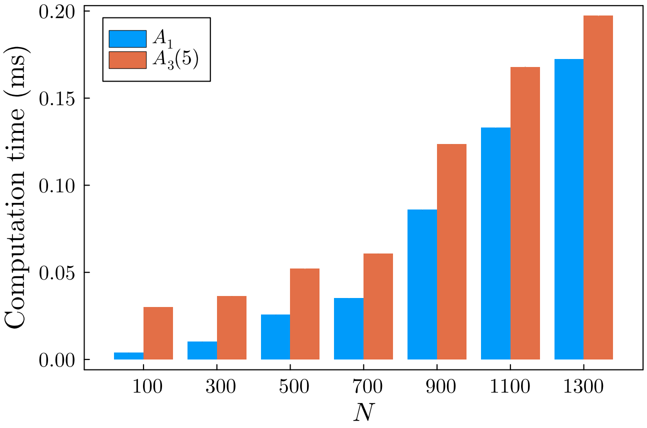

FIGURE 5. Computation time vs. N.

VI. CONCLUSION

In this article, we derive four approximate Wiener filter formulas using a truncation technique based on the principal components of a composite covariance matrix. Our approximate filters do not directly involve the inverse of the covariance matrix. This results in the stable performance of our filters even as the covariance matrix becomes increasingly ill-conditioned, as demonstrated by our results with real data. These results show deteriorating performance for the Wiener filter, demonstrating its numerical unreliability for covariance matrices with large condition numbers. We describe two methods with varying complexity for optimally trading off computation for accuracy using our filters. We also show that the approximate filter formulas converge to the Wiener filter in terms of asymptotic scaling laws. The numerical comparison of computation times demonstrates that the computation of our filters comes at a small extra cost compared to the Wiener filter and is even more efficient in some use cases.

Finally, it has not escaped our notice that our method of approximating Wiener filters in eigensubspaces spanned by the joint principal components (i.e., those of the composite covariance matrix) suggests a comparison with reduced-rank Wiener filters in Krylov subspaces composed by $\boldsymbol{C}_{YY}$ and $\boldsymbol{C}_{YX}$ [9], [10]. Their performance relative to our filters, as well as the two well-conditioned approximate filters derived using the joint principal components in [12], warrants further investigation.

REFERENCES

[1] P. U. Damale, E. K. P. Chong, and L. L. Scharf, "Wiener filtering without covariance matrix inversion," in *Proc. IEEE 48th Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[2] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, Feb. 1970.

[3] J. M. Tabeart, S. L. Dance, A. S. Lawless, N. K. Nichols, and J. A. Waller, "Improving the condition number of estimated covariance matrices," *Tellus A: Dyn. Meteorol. Oceanogr.*, vol. 72, no. 1, pp. 1–19, 2020.

[4] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Reading, MA, USA: Addison-Wesley, 1991.

[5] J. S. Goldstein and I. S. Reed, "Reduced-rank adaptive filtering," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 492–496, Feb. 1997.

[6] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A multistage representation of the Wiener filter based on orthogonal projections," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2943–2959, Nov. 1998.

[7] Y. Hua, M. Nikpour, and P. Stoica, “Optimal reduced-rank estimation and filtering,” *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 457–469, Mar. 2001.

[8] G. Dietl and W. Utschick, “On reduced-rank approaches to matrix Wiener filters in MIMO systems,” in *Proc. IEEE 3rd Int. Symp. Signal Process. Inf. Technol.*, 2003, pp. 82–85.

[9] H. Ge, L. L. Scharf, and M. Lundberg, “Reduced-rank multiuser detectors based on vector and matrix conjugate gradient wiener filters,” in *Proc. IEEE 5th Workshop Signal Process. Adv. Wireless Commun.*, 2004, pp. 189–193.

[10] G. K. Dietl, “Reduced-rank matrix Wiener filters in Krylov subspaces,” in *Linear Estimation and Detection in Krylov Subspaces*, vol. 1. Berlin, Germany: Springer, 2007, ch. 4, pp. 71–109.

[11] L. L. Scharf, E. K. Chong, M. D. Zoltowski, J. S. Goldstein, and I. S. Reed, “Subspace expansion and the equivalence of conjugate direction and multistage Wiener filters,” *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 5013–5019, Oct. 2008.

[12] E. K. P. Chong, “Well-conditioned linear minimum mean square error estimation,” *IEEE Control Syst. Lett.*, vol. 6, pp. 2431–2436, 2022.

[13] *VXAZN Historical Price Data*, Cboe Global Markets, Inc. Accessed: Nov. 6, 2022. [Online]. Available: https://www.cboe.com/tradable_products/vix/vix_historical_data/

[14] J. Veraart, D. S. Novikov, D. Christiaens, B. Ades-Aron, J. Sijbers, and E. Fieremans, “Denoising of diffusion MRI using random matrix theory,” *Neuroimage*, vol. 142, pp. 394–406, Nov. 2016.

[15] R. O’Dowd, “The Wiener-Levinson algorithm and ill-conditioned normal equations,” *Geophys. J. Int.*, vol. 106, no. 2, pp. 399–406, 1991.

[16] A. Cohen, M. A. Davenport, and D. Leviatan, “On the stability and accuracy of least squares approximations,” *Found. Comput. Math.*, vol. 13, pp. 819–834, 2013.

[17] X. Deng, L. Yin, S. Peng, and M. Ding, “An iterative algorithm for solving ill-conditioned linear least squares problems,” *Geod. Geodyn.*, vol. 6, no. 6, pp. 453–459, 2015.

[18] J. Ghafuri, H. Du, and S. Jassim, “Sensitivity and stability of pretrained CNN filters,” *Proc. SPIE*, vol. 11734, 2021, Art. no. 117340B.

[19] S. Smale, “On the efficiency of algorithms of analysis,” *Bull. Amer. Soc.*, vol. 13, no. 2, pp. 87–121, Mar. 1985.

[20] A. Edelman, “Eigenvalues and condition numbers of random matrices,” *SIAM J. Matrix Anal.*, vol. 9, no. 4, pp. 543–560, Oct. 1988.

[21] L. Laloux, P. Cizeau, J.-P. Bouchaud, and M. Potters, “Noise dressing of financial correlation matrices,” *Phys. Rev. Lett.*, vol. 83, no. 7, Aug. 1999, Art. no. 1467.

[22] L. Laloux, P. Cizeau, M. Potters, and J.-P. Bouchaud, “Random matrix theory and financial correlations,” *Int. J. Theor. Appl. Finance*, vol. 3, no. 03, pp. 391–397, 2000.

[23] G. Zumbach, “Empirical properties of large covariance matrices,” *Quant. Finance*, vol. 11, no. 7, pp. 1091–1102, Mar. 2011.

[24] J.-P. Bouchaud and M. Potters, “Financial applications of random matrix theory: A short review,” in *The Oxford Handbook of Random Matrix Theory*. Oxford, U.K.: Oxford Univ. Press, Sep. 2015, ch. 40, pp. 824–850.

[25] A. Bloemendal, A. Knowles, H.-T. Yau, and J. Yin, “On the principal components of sample covariance matrices,” *Probab. Theory Relat. Fields*, vol. 164, no. 1, pp. 459–552, Feb. 2016.

[26] J. Bun, J.-P. Bouchaud, and M. Potters, “Cleaning large correlation matrices: Tools from random matrix theory,” *Phys. Rep.*, vol. 666, pp. 1–109, Jan. 2017.

[27] M. Ghorbani and E. K. P. Chong, “Stock price prediction using principal components,” *PLoS One*, vol. 15, no. 3, Mar. 2020, Art. no. e0230124.

[28] E. K. P. Chong and S. H. Żak, *An Introduction to Optimization*, 4th ed.Hoboken, NJ, USA: Wiley, 2013.

[29] A. Saha, B. G. Malkiel, and A. Rinaudo, “Has the VIX index been manipulated?,” *J. Asset Manag.*, vol. 20, no. 1, pp. 1–14, 2019.

[30] A. Hasanzadeh, X. Liu, N. Duffield, and K. R. Narayanan, “Piecewise stationary modeling of random processes over graphs with an application to traffic prediction,” in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 3779–3788.

[31] D. A. Dickey and W. A. Fuller, “Distribution of the estimators for autoregressive time series with a unit root,” *J. Amer. Statist. Assoc.*, vol. 74, no. 366a, pp. 427–431, 1979.

[32] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, “Julia: A fresh approach to numerical computing,” *SIAM Rev.*, vol. 59, no. 1, pp. 65–98, 2017. [Online]. Available: https://epubs.siam.org/doi/10.1137/141000671

[33] V. A. Marčenko and L. A. Pastur, “Distribution of eigenvalues for some sets of random matrices,” *Math. USSR Sb.*, vol. 1, no. 4, pp. 457–486, 1967.

[34] J. Baik and J. W. Silverstein, “Eigenvalues of large sample covariance matrices of spiked population models,” *J. Multivar. Anal.*, vol. 97, no. 6, pp. 1382–1408, Jul. 2006.

[35] J. Veraart, E. Fieremans, and D. S. Novikov, “Diffusion MRI noise mapping using random matrix theory,” *Magn. Reson. Med.*, vol. 76, no. 5, pp. 1582–1593, Nov. 2016.

[36] W. Zhao, Y. Lv, Q. Liu, and B. Qin, “Detail-preserving image denoising via adaptive clustering and progressive PCA thresholding,” *IEEE Access*, vol. 6, pp. 6303–6315, Dec. 2017.

[37] M. D. Does et al., “Evaluation of principal component analysis image denoising on multi-exponential MRI relaxometry,” *Magn. Reson. Med.*, vol. 81, no. 6, pp. 3503–3514, Feb. 2019.

[38] I. M. Johnstone, “High dimensional statistical inference and random matrices,” 2006. [Online]. Available: arXiv:math/0611589

[39] J. W. Silverstein, “The smallest eigenvalue of a large dimensional Wishart matrix,” *Ann. Probab.*, vol. 13, no. 4, pp. 1364–1368, Nov. 1985.

[40] J. W. Silverstein, “Strong convergence of the empirical distribution of eigenvalues of large dimensional random matrices,” *J. Multivar. Anal.*, vol. 55, no. 2, pp. 331–339, Nov. 1995.

[41] J. W. Silverstein and Z. D. Bai, “On the empirical distribution of eigenvalues of a class of large dimensional random matrices,” *J. Multivar. Anal.*, vol. 54, no. 2, pp. 175–192, Aug. 1995.

[42] J. W. Silverstein and S.-I. Choi, “Analysis of the limiting spectral distribution of large dimensional random matrices,” *J. Multivar. Anal.*, vol. 54, no. 2, pp. 295–309, Aug. 1995.

[43] I. M. Johnstone, “On the distribution of the largest eigenvalue in principal components analysis,” *Ann. Statist.*, vol. 29, no. 2, pp. 295–327, Apr. 2001.

[44] A. Soshnikov, “A note on universality of the distribution of the largest eigenvalues in certain sample covariance matrices,” *J. Statist. Phys.*, vol. 108, pp. 1033–1056, Sep. 2002.

[45] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral, T. Guhr, and H. E. Stanley, “Random matrix approach to cross correlations in financial data,” *Phys. Rev. E*, vol. 65, no. 6, Jun. 2002, Art. no. 066126.

[46] S. Péché, “Universality of local eigenvalue statistics for random sample covariance matrices,” Ph.D. dissertation, Dept. Statist. Stoch. Model., Univ. Paris-Sud, Paris, France, 2003.