Efficient and Accurate Rectangular Window Subspace Tracking

Timothy M. Toolan and Donald W. Tufts
Dept. of Electrical Engineering, University of Rhode Island, Kingston, RI 02881 USA
toolan@ele.uri.edu, tufts@ele.uri.edu

***Abstract*— In this paper, we describe a rectangular window subspace tracking algorithm, which tracks the r largest singular values and corresponding left singular vectors of a sequence of $n \times c$ matrices in $O(nr^2)$ time. This algorithm is designed to track rapidly changing subspaces. It uses a rectangular window to include a finite number of approximately stationary data columns. This algorithm is based on the Improved Fast Adaptive Subspace Tracking (IFAST) algorithm of Toolan and Tufts, but reforms the rth order eigendecomposition with an alternative method that takes advantage of matrix structure. This matrix is a special rank-six modification of a diagonal matrix, so its eigendecomposition can be determined with only a single $O(r^3)$ matrix product to rotate its eigenvectors, and all other computation is $O(r^2)$. Methods for implementing this algorithm in a numerically stable way are also discussed.**

I. Introduction

Let us assume that we have a data source which produces a new length n column vector at regular intervals. This situation applies when we have an n element sensor array and take snapshots at regularly spaced time intervals, it applies when we analyze a digital image by panning across either the columns or rows, and it applies to a single sensor which takes regularly spaced samples in time, then creates a vector from the last n samples.

The goal of subspace tracking, is to apply a sliding window to our data to create a sequence of overlapping matrices, then track the r principal singular values and left singular vectors of that windowed matrix, or an orthogonal set of columns which span that r dimensional principal subspace. The large overlap between successive windowed matrices allows one to reduce computation by determining the new subspace as an update of the subspace determined from the previous matrix. Because the singular value decomposition (SVD) is a full matrix operation, these update methods are generally approximations. One reason we are often only interested in the left singular vectors, is that they are also the eigenvectors of the sample correlation matrix.

Most of the earlier methods of subspace tracking, are based on applying an exponential window to the data [1]–[4], but recently there have been some methods which apply a rectangular window [5]–[8]. The rectangular window methods allow inclusion of a finite amount of data, and can give better performance when the data is highly non-stationary, or there are abrupt changes in the data.

II. Sliding Rectangularly Windowed Data

Letting $\mathbf{x}_i$ be the ith column vector that we received, a length c rectangularly windowed matrix is formed by creating a matrix from the last c column vectors. At time t, we can write our previous matrix, M, and current matrix, $\tilde{M}$, as

$$M = \left[\; \mathbf{x}_{t-c} \;\middle|\; \mathbf{x}_{t-c+1} \;\middle|\; \mathbf{x}_{t-c+2} \;\middle|\; \cdots \;\middle|\; \mathbf{x}_{t-1} \;\right], \tag{1}$$

$$\tilde{M} = \left[\; \mathbf{x}_{t-c+1} \;\middle|\; \mathbf{x}_{t-c+2} \;\middle|\; \cdots \;\middle|\; \mathbf{x}_{t-1} \;\middle|\; \mathbf{x}_{t} \;\right]. \tag{2}$$

Note that M and $\tilde{M}$ share all but one column. We can write the current matrix in terms of the previous matrix as

$$\tilde{M} = \left(M - \mathbf{x}_{t-c}\mathbf{e}_1^H\right) P + \mathbf{x}_t \mathbf{e}_c^H, \tag{3}$$

where $\mathbf{e}_i$ is the ith length c canonical vector (the ith column of a $c \times c$ identity matrix) and $P = [\mathbf{e}_2, \mathbf{e}_3, \cdots, \mathbf{e}_c \,|\, \mathbf{e}_1]$.

Since the left singular vectors of a matrix are the eigenvectors of that matrix times its conjugate transpose, then from (3), we can write

$$\tilde{M}\tilde{M}^H = MM^H - \mathbf{x}_{t-c}\mathbf{x}_{t-c}^H + \mathbf{x}_t\mathbf{x}_t^H, \tag{4}$$

which makes it clear that $\tilde{M}\tilde{M}^H$ is a rank-two perturbation of MM^H. The problem of updating the eigendecomposition of a rank-one perturbation of a symmetric matrix has been well studied both theoretically and numerically, (see [9] and references therein). A rank-two perturbation of this form has been much less studied, even though it is equivalent to two sequential rank-one perturbations.

III. Defining the Problem

We start with two sequential $n \times c$ rectangularly windowed complex matrices M and $\tilde{M}$, along with approximations to the r largest singular values and corresponding left singular vectors of M, which we will call $\Sigma' \in \mathbb{R}^{r \times r}$ and $U' \in \mathbb{C}^{n \times r}$ respectively. Our goal is to determine a good approximation of the r (or possibly $r+1$) largest singular values and corresponding left singular vectors of $\tilde{M}$ by updating U' and Σ'.

From (1) and (2), we can see that M and $\tilde{M}$ share all but one column, and from (4) we can see that the ordering of the columns has no effect on left singular vectors, therefore the column space spanned by U' should be close to the column space spanned by the r dominant left singular vectors of the $n \times (c-1)$ matrix of shared columns, $[\,\mathbf{x}_{t-c+1} \,|\, \mathbf{x}_{t-c+2} \,|\, \cdots \,|\, \mathbf{x}_{t-2} \,|\, \mathbf{x}_{t-1}\,]$. The subspace that is not included in U' which may have a strong influence on the dominant left singular vectors of $\tilde{M}$ is the part of both $\mathbf{x}_{t-c}$ and $\mathbf{x}_t$ that is orthogonal to U'. If we use a Gram-Schmidt like method [10] to augment U' by this subspace, and call this augmentation $Q \in \mathbb{C}^{n\times 2}$, then the subspace spanned by $[\,U' \,|\, Q\,]$ should be close to the subspace spanned by the r largest singular vectors of $\tilde{M}$. In [8], a detailed analytical analysis is presented, which shows why the subspace $[\,U' \,|\, Q\,]$ is a good approximation to the subspace spanned by the r dominant left singular values of $\tilde{M}$, and that it is not trivial to find an $r+2$ dimensional subspace that will give a better approximation.

The idea behind the *improved fast adaptive subspace tracking* (IFAST) algorithm [8], [11], is to approximate the matrix $\tilde{M}$ by the rank $r+2$ matrix $\tilde{M}' = [\,U' \,|\, Q\,][\,U' \,|\, Q\,]^H\tilde{M}$, then determine the singular values and left singular vectors of $\tilde{M}'$ in an efficient way. Approximating $\tilde{M}$ by $\tilde{M}'$ is the *only* approximation involved in the algorithm. Determining the non-zero singular values and corresponding left singular vectors of $\tilde{M}'$, is equivalent to applying the Rayleigh-Ritz procedure [12] to $\tilde{M}\tilde{M}^H$ using the subspace $[\,U' \,|\, Q\,]$. This means that the SVD of $\tilde{M}'$ is the optimal approximation to the SVD of $\tilde{M}$, when limited to the subspace spanned by the columns of $[\,U' \,|\, Q\,]$ [12].

Table I is intended to clearly show how IFAST works, and will produce the exact singular values and left singular vectors of $\tilde{M}'$. Table I will also produce the same result as the algorithm presented in this paper, as well as that in [8], [11]. The only difference between Table I and the IFAST algorithm in [8], [11], is that the $O(ncr)$ computation in step 2 is performed as an equivalent $O(nc)$ computation. The contribution of this paper is to replace the eigendecomposition in step 3 with a single matrix product of two $(r+2)\times(r+2)$ matrices, and all other computation will be $O(r^2)$. This is approximately four times faster than a conventional EVD [10].

Because the efficient computation of the EVD is done using secular equations, [13], we will call the version of the algorithm presented in this paper the *Improved Secular Fast Adaptive Subspace Tracking* (ISFAST) algorithm. The only computation in this algorithm that is not the equivalent of a matrix times a vector, is a the product of two $(r+2)\times(r+2)$ matrices, and the product of a $n\times(r+2)$ matrix with a $(r+2)\times(r+2)$ matrix.

IV. The Details of Σ', U', and Q

Although Σ' and U' are approximations, there are a few properties that they must have. Σ' must be diagonal and nonnegative, the columns of U' must form an orthonormal set (ie. $U'^H U' = I$), and $U'^H MM^H U'$ must equal Σ'^2. These conditions will always be satisfied if U' and Σ' were generated by a previous iteration of this algorithm [8], or from the SVD of M. All these constraints really say, is that U' should be rotated so that $U'^H MM^H U'$ is diagonal. For any matrix $U_x \in \mathbb{C}^{n\times r}$, whose columns form an orthonormal set, we can generate a U' and Σ' that satisfy these constraints by taking the EVD of $U_x^H MM^H U_x = U_y \Sigma_y U_y^H$, then setting $\Sigma' = \sqrt{\Sigma_y}$ and $U' = U_x U_y$.

TABLE I

Theoretical IFAST Algorithm

Step	Calculation
1)	$\dot{\mathbf{q}}_1 = \dfrac{(I - U'U'^H)\,\mathbf{x}_{t-c}}{\|(I - U'U'^H)\,\mathbf{x}_{t-c}\|}$ $\dot{\mathbf{q}}_2 = \dfrac{(I - U'U'^H - \dot{\mathbf{q}}_1\dot{\mathbf{q}}_1^H)\,\mathbf{x}_t}{\|(I - U'U'^H - \dot{\mathbf{q}}_1\dot{\mathbf{q}}_1^H)\,\mathbf{x}_t\|}$
2)	$\tilde{F} = [\,U' \,\vert\, \dot{Q}\,]^H \tilde{M}\tilde{M}^H [\,U' \,\vert\, \dot{Q}\,]$
3)	$U_f \Sigma_f U_f^H = \tilde{F}$
4)	$\tilde{U}' = [\,U' \,\vert\, \dot{Q}\,]U_f$ $\tilde{\Sigma}'^2 = \Sigma_f$

In the last section, we mentioned that we will create the matrix Q using a Gram-Schmidt like method [10] to augment U' with $\mathbf{x}_{t-c}$ (the column we are discarding from M) and $\mathbf{x}_t$ (the column we are adding to $\tilde{M}$). We can augment U' with $\mathbf{x}_{t-c}$ followed by $\mathbf{x}_t$, or $\mathbf{x}_t$ followed by $\mathbf{x}_{t-c}$, as they will both produce a Q which span the same column space. In fact, we can rotate Q by any 2×2 unitary matrix, and it will not change the column space spanned by Q. We will take advantage of this property to make later computation easier. We start by creating the matrix $\dot{Q} = [\,\dot{\mathbf{q}}_1 \,|\, \dot{\mathbf{q}}_2\,]$ using Gram-Schmidt augmentation of U',

$$\dot{\mathbf{q}}_1 = \frac{\left(I - U'U'^H\right)\mathbf{x}_{t-c}}{\|\left(I - U'U'^H\right)\mathbf{x}_{t-c}\|},$$
$$\dot{\mathbf{q}}_2 = \frac{\left(I - U'U'^H - \dot{\mathbf{q}}_1\dot{\mathbf{q}}_1^H\right)\mathbf{x}_t}{\left\|\left(I - U'U'^H - \dot{\mathbf{q}}_1\dot{\mathbf{q}}_1^H\right)\mathbf{x}_t\right\|}.$$

The two vectors $\dot{\mathbf{q}}_1$ and $\dot{\mathbf{q}}_2$ are an orthonormal set that span the desired column space, but in general the 2×2 Hermitian matrix $\hat{T} = \dot{Q}^H MM^H \dot{Q}$ will not be diagonal, therefore we will rotate $\dot{Q}$ to form Q, such that $\hat{\Sigma} = Q^H MM^H Q$ is diagonal.

In order to determine the necessary rotation, we can take the EVD of $\hat{T}$, which we will write as

$$\hat{T} = \left[\begin{array}{c|c} \dot{\mathbf{q}}_1^H MM^H \dot{\mathbf{q}}_1 & \dot{\mathbf{q}}_1^H MM^H \dot{\mathbf{q}}_2 \\ \hline \dot{\mathbf{q}}_2^H MM^H \dot{\mathbf{q}}_1 & \dot{\mathbf{q}}_2^H MM^H \dot{\mathbf{q}}_2 \end{array}\right] = \hat{U}\hat{\Sigma}\hat{U}^H.$$

Because the formation of $\hat{T}$ is required later in the algorithm, there is no additional computational load introduced by forming it now. The eigenvalues of a matrix are the roots of its characteristic equation, and since $\hat{T}$ is a 2×2 matrix, we can use the quadratic equation to find the eigenvalues. Defining $b = (\hat{t}_{1,1} + \hat{t}_{2,2})/2$ and $c = \hat{t}_{1,1}\hat{t}_{2,2} - |\hat{t}_{1,2}|^2$, where $\hat{t}_{i,j}$ is the (i,j)th element of $\hat{T}$, allows us to write the eigenvalues of $\hat{T}$ as

$$\hat{\Sigma} = \left[\begin{array}{c|c} b + \sqrt{b^2 - c} & 0 \\ \hline 0 & b - \sqrt{b^2 - c} \end{array}\right].$$

Now that we have the eigenvalues of $\hat{T}$, we can determine its eigenvectors. We can use the definition of an eigenvector

[10], and write

$$\hat{T}\begin{bmatrix}1\\ \beta\end{bmatrix} = \hat{\sigma}_1\begin{bmatrix}1\\ \beta\end{bmatrix},$$

where $[1,\, \beta]^T$ is an unnormalized eigenvector of $\hat{T}$ corresponding to eigenvalue $\hat{\sigma}_1$. Solving for β, normalizing the eigenvector, then repeating for the second eigenvector, we get

$$\hat{U} = \left[\begin{array}{c|c}\alpha_1 & \alpha_2\\ \hline \phi\alpha_2 & -\phi\alpha_1\end{array}\right],$$

where $\alpha_1 = 1/\sqrt{1+\left|(\hat{\sigma}_1 - \hat{t}_{1,1})/\hat{t}_{1,2}\right|^2}$, $\alpha_2 = \sqrt{1-\alpha_1^2}$, and $\phi = \hat{t}_{2,1}/|\hat{t}_{2,1}|$. This allows us to define

$$Q = \dot{Q}\hat{U},$$

which will have the property $Q^H M M^H Q = \hat{\Sigma}$.

V. Efficiently calculating the SVD of $\tilde{M}'$

Determining the singular values and left singular vectors of $\tilde{M}'$ is equivalent to determining the eigenvalues and eigenvectors of $\tilde{M}'\tilde{M}'^H$ [10], therefore we will start by analyzing the eigendecomposition of

$$\tilde{M}'\tilde{M}'^H = [\,U' \mid Q\,][\,U' \mid Q\,]^H \tilde{M}\tilde{M}^H [\,U' \mid Q\,][\,U' \mid Q\,]^H.$$

If we define

$$\tilde{F} = [\,U' \mid Q\,]^H \tilde{M}\tilde{M}^H [\,U' \mid Q\,] = U_f \Sigma_f U_f^H, \qquad (5)$$

then we can write

$$\tilde{M}'\tilde{M}'^H = \left([\,U' \mid Q\,]U_f\right)\Sigma_f\left(U_f^H [\,U' \mid Q\,]^H\right),$$

and it becomes obvious that the non-zero eigenvalues of $\tilde{M}'\tilde{M}'^H$ will be the eigenvalues of $\tilde{F}$, and the corresponding eigenvectors of $\tilde{M}'\tilde{M}'^H$ will be $[\,U' \mid Q\,]U_f$.

From (4), we know $\tilde{M}\tilde{M}^H = MM^H - \mathbf{x}_{t-c}\mathbf{x}_{t-c}^H + \mathbf{x}_t\mathbf{x}_t^H$, which we can substitute into (5), to get

$$\tilde{F} = [\,U' \mid Q\,]^H MM^H [\,U' \mid Q\,] - \mathbf{a}\mathbf{a}^H + \mathbf{b}\mathbf{b}^H, \qquad (6)$$

where the vectors $\mathbf{a}$ and $\mathbf{b}$ are defined as

$$\mathbf{a} = [\,U' \mid Q\,]^H \mathbf{x}_{t-c}, \qquad \text{and} \qquad \mathbf{b} = [\,U' \mid Q\,]^H \mathbf{x}_t.$$

Multiplying out the matrix blocks in the first term of (6) gives us

$$\tilde{F} = \left[\begin{array}{c|c} U'^H MM^H U' & U'^H MM^H Q\\ \hline Q^H MM^H U' & Q^H MM^H Q\end{array}\right] - \mathbf{a}\mathbf{a}^H + \mathbf{b}\mathbf{b}^H. \quad (7)$$

In section IV we made it clear that $U'^H MM^H U' = \Sigma'^2$ and $Q^H MM^H Q = \hat{\Sigma}$, both of which we already have. If we define

$$Z = U'^H MM^H Q,$$

(remember $M^H Q = (M^H\dot{Q})\hat{U}$, and we already computed $M^H\dot{Q}$ when constructing $\hat{T}$), we can write (7) as

$$\tilde{F} = \left[\begin{array}{c|c}\Sigma'^2 & 0\\ \hline 0 & \hat{\Sigma}\end{array}\right] + \left[\begin{array}{c|c}0 & Z\\ \hline Z^H & 0\end{array}\right] - \mathbf{a}\mathbf{a}^H + \mathbf{b}\mathbf{b}^H. \qquad (8)$$

The matrix $\tilde{F}$, as it is written in (8), is a rank-six modification of a diagonal matrix. The matrices $\mathbf{a}\mathbf{a}^H$ and $\mathbf{b}\mathbf{b}^H$ are each rank-one, and the matrix $\left[\begin{smallmatrix}0 & Z\\ Z^H & 0\end{smallmatrix}\right]$ is rank-four, which can be seen from (16) in the appendix.

Because the eigendecomposition of certain low rank modifications of diagonal matrices can be computed in $O(n^2)$ time, we will take advantage of this to compute the EVD of $\tilde{F}$ more efficiently. We will compute the EVD of $\tilde{F}$ in two steps. The first step is to compute the EVD of

$$\dot{G} = \left[\begin{array}{c|c}\Sigma'^2 & 0\\ \hline 0 & \hat{\Sigma}\end{array}\right] + \left[\begin{array}{c|c}0 & Z\\ \hline Z^H & 0\end{array}\right] = \dot{U}\dot{\Sigma}\dot{U}^H, \qquad (9)$$

which is a rank-four modification of a diagonal matrix. Since $\tilde{F} = \dot{G} - \mathbf{a}\mathbf{a}^H + \mathbf{b}\mathbf{b}^H$, if we rotate $\tilde{F}$ by the eigenvectors of $\dot{G}$, we can define the matrix $\ddot{H} = \dot{U}\tilde{F}\dot{U}^H$. Therefore, defining $\dot{\mathbf{a}} = \dot{U}^H\mathbf{a}$ and $\dot{\mathbf{b}} = \dot{U}^H\mathbf{b}$, the second step is to compute the EVD of

$$\ddot{H} = \dot{\Sigma} - \dot{\mathbf{a}}\dot{\mathbf{a}}^H + \dot{\mathbf{b}}\dot{\mathbf{b}}^H = \ddot{U}\ddot{\Sigma}\ddot{U}^H,$$

which is a rank-two modification of a diagonal matrix.

Finally, we can determine the EVD of $\tilde{F}$ as $\Sigma_f = \ddot{\Sigma}$ and $U_f = \dot{U}\ddot{U}$, allowing us to write the singular values and left singular vectors of $\tilde{M}'$ as $\tilde{\Sigma}'^2 = \ddot{\Sigma}$ and $\tilde{U}' = [U'|Q]\dot{U}\ddot{U}$.

VI. Computing the EVD of $\dot{G}$

The eigenvalues of $\dot{G}$ from (9) are the roots of its characteristic polynomial, $\dot{C}(\lambda) = \det[\dot{G} - \lambda I]$, which can also be written as

$$\dot{C}(\lambda) = \det\left[\begin{array}{c|c}\Sigma'^2 - \lambda I & Z\\ \hline Z^H & \hat{\Sigma} - \lambda I\end{array}\right]. \qquad (10)$$

Using [8, pp. 60-62], the determinant of a bordered diagonal matrix in the form of (10) can be written as

$$\dot{C}(\lambda) = \left(\prod_{i=1}^{r}(\sigma_i'^2 - \lambda)\right)\left(\dot{w}_1(\lambda)\dot{w}_2(\lambda) - |\dot{w}_x(\lambda)|^2\right),$$

where

$$\dot{w}_i(\lambda) = \hat{\sigma}_i - \lambda - \sum_{j=1}^{r}\frac{|z_{j,i}|^2}{\sigma_j'^2 - \lambda}, \qquad \dot{w}_x(\lambda) = \sum_{j=1}^{r}\frac{z_{j,1}^* z_{j,2}}{\sigma_j'^2 - \lambda}, \qquad (11)$$

σ_i' is the ith diagonal element of Σ', $z_{j,i}$ is the (j,i)th element of Z, and $\hat{\sigma}_i$ is the ith diagonal element of $\hat{\Sigma}$. Because the product term in front of $\dot{C}(\lambda)$ has no effect on its roots, we can cancel it to get

$$\dot{w}(\lambda) = \dot{w}_1(\lambda)\dot{w}_2(\lambda) - |\dot{w}_x(\lambda)|^2,$$

which is a rank-four version of the rank-two formula given in [14], and behaves similarly to the rank-two secular function from [8]. We can now get the $r+2$ eigenvalues of $\dot{G}$ by determining the roots of $\dot{w}(\lambda)$, which we will call $\dot{\sigma}_1$ through $\dot{\sigma}_{r+2}$.

If $\dot{\mathbf{u}}$ is the unnormalized ith eigenvector of $\dot{G}$, then from the definition of an eigenvector [10], we get $\dot{G}\dot{\mathbf{u}} = \dot{\mathbf{u}}\dot{\sigma}_i$. Solving for $\dot{\mathbf{u}}$, (the details of which are given in the appendix), we get

$$\dot{\mathbf{u}} = \left[\begin{array}{c|c}\Sigma'^2 - \dot{\sigma}_i I & 0\\ \hline 0 & I\end{array}\right]^{-1}\left(\begin{bmatrix}\mathbf{z}_1\\ -1\\ 0\end{bmatrix} + \frac{\dot{w}_1(\dot{\sigma}_i)}{\dot{w}_x(\dot{\sigma}_i)}\begin{bmatrix}\mathbf{z}_2\\ 0\\ -1\end{bmatrix}\right) \quad (12)$$

where $\mathbf{z}_1$ and $\mathbf{z}_2$ are the first and second columns of Z, respectively. This allows us to write the normalized ith eigenvector of $\dot{G}$ as $\dot{\mathbf{u}}_i = \dot{\mathbf{u}}/\|\dot{\mathbf{u}}\|$.

To stably calculate the eigenvalues of $\dot{G}$, they should be calculated as two sequential updates using [14], combined with the techniques described at the end of the next section.

VII. Computing the EVD of $\ddot{H}$

After determining the EVD of $\dot{G}$, we can determine the EVD of $\ddot{H}$. The eigenvalues of $\ddot{H} = \dot{\Sigma} - \dot{\mathbf{a}}\dot{\mathbf{a}}^H + \dot{\mathbf{b}}\dot{\mathbf{b}}^H$ are the roots of the rank-two secular equation

$$\ddot{w}(\lambda) = \ddot{w}_a(\lambda)\ddot{w}_b(\lambda) + |\ddot{w}_{ab}(\lambda)|^2, \tag{13}$$

which is a rank-two version of the secular function given in [13], and whose derivation is in [8]. The parts of (13) are defined as

$$\ddot{w}_a(\lambda) = 1 - \sum_{j=1}^{r} \frac{|\dot{a}_j|^2}{\dot{\sigma}_j - \lambda}, \qquad \ddot{w}_b(\lambda) = 1 + \sum_{j=1}^{r} \frac{|\dot{b}_j|^2}{\dot{\sigma}_j - \lambda},$$

$$\ddot{w}_{ab}(\lambda) = \sum_{j=1}^{r} \frac{\dot{a}_j^* \dot{b}_j}{\dot{\sigma}_j - \lambda},$$

where $\dot{a}_j$ and $\dot{b}_j$ are the jth elements of $\dot{\mathbf{a}}$ and $\dot{\mathbf{b}}$, respectively.

The unnormalized ith eigenvector of $\ddot{H}$ is a linear combination of $\dot{\mathbf{a}}$ and $\dot{\mathbf{b}}$, and from [8] can be written as

$$\ddot{\mathbf{u}} = \left(\dot{\Sigma} - \ddot{\sigma}_i I\right)^{-1} \left(\dot{\mathbf{a}} + \frac{\ddot{w}_a(\ddot{\sigma}_i)}{\ddot{w}_{ab}(\ddot{\sigma}_i)}\dot{\mathbf{b}}\right). \tag{14}$$

This allows us to write the normalized ith eigenvector of $\ddot{H}$ as $\ddot{\mathbf{u}}_i = \ddot{\mathbf{u}}/\|\ddot{\mathbf{u}}\|$.

To stably calculate the eigenvalues of $\ddot{H}$, they should be calculated as two sequential updates using [15], with the stopping criterion from [16], and the first set of eigenvectors should be calculated using the method in [16], [17]. To avoid an $O(r^3)$ rotation, the eigenvectors of $\ddot{H}$ are calculated using the method in this paper. For closely spaced eigenvalues, there will be some loss of orthogonality, which should be able to be addressed using methods like [18], [19] and [20].

VIII. Conclusion

In this paper, we present a method for efficiently computing the small EVD of the IFAST algorithm. We call this modified version of IFAST the *Improved Secular Fast Adaptive Subspace Tracking* (ISFAST) algorithm. The steps of the ISFAST algorithm are presented in table II, and produce the same result as Table I. The dominant step computationally is step 4-a, as it has one $O(r^3)$ matrix product, and one $O(nr^2)$ matrix product. All other steps in the algorithm only contain terms which are matrices times vectors, or their computational equivalent.

The computational improvement introduced by performing the calculations as presented in this paper are shown in Fig. 1 for a 50×50 complex matrix. The *full decomposition* represents determining the full set of singular values and left singular vectors of $\tilde{M}$ using a conventional SVD [10]. The *ISFAST* algorithm and the *IFAST* algorithm both produce the singular values and left singular vectors of $\tilde{M}'$. The *FAST* algorithm [5] is given for comparison, and is almost computationaly equivalent to the steps in Table I. The *secular update* method uses the method from §VII directly on $\tilde{M}$, and will give the same results as the full decomposition. Plots of approximate FLOPS vs. subspace dimension for other square matrix dimensions look very similar, even for very large n.

We have not mentioned how we determine the subspace dimension, r, but the method in [21], which is based on the energy in the noise subspace works well with this algorithm for determining r at each iteration.

TABLE II
The ISFAST Algorithm

Step	Calculation
1-a)	$\dot{\mathbf{q}}_1 = \dfrac{(I - U'U'^H)\mathbf{x}_{t-c}}{\Vert(I - U'U'^H)\mathbf{x}_{t-c}\Vert}$
b)	$\dot{\mathbf{q}}_2 = \dfrac{(I - U'U'^H - \dot{\mathbf{q}}_1\dot{\mathbf{q}}_1^H)\mathbf{x}_t}{\Vert(I - U'U'^H - \dot{\mathbf{q}}_1\dot{\mathbf{q}}_1^H)\mathbf{x}_t\Vert}$
c)	$\hat{T} = \dot{Q}^H M M^H \dot{Q}$
d)	$\hat{U}\hat{\Sigma}\hat{U}^H = \hat{T}$
e)	$Q = \dot{Q}\hat{U}$
2-a)	$Z = U'^H M M^H Q$
b)	$\dot{G} = \begin{bmatrix} \Sigma'^2 & 0 \\ 0 & \hat{\Sigma} \end{bmatrix} + \begin{bmatrix} 0 & Z \\ Z^H & 0 \end{bmatrix}$
c)	$\dot{U}\dot{\Sigma}\dot{U}^H = \dot{G}$
3-a)	$\dot{\mathbf{a}} = \dot{U}^H [\,U' \mid Q\,]^H \mathbf{x}_{t-c}$
b)	$\dot{\mathbf{b}} = \dot{U}^H [\,U' \mid Q\,]^H \mathbf{x}_t$
c)	$\ddot{H} = \dot{\Sigma} - \dot{\mathbf{a}}\dot{\mathbf{a}}^H + \dot{\mathbf{b}}\dot{\mathbf{b}}^H$
d)	$\ddot{U}\ddot{\Sigma}\ddot{U}^H = \ddot{H}$
4-a)	$\tilde{U}' = [\,U' \mid Q\,]\dot{U}\ddot{U}$
b)	$\tilde{\Sigma}'^2 = \ddot{\Sigma}$

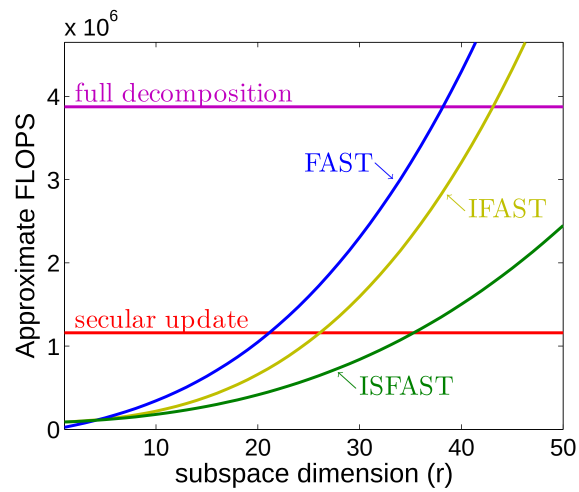

Fig. 1. Computational requirements of various algorithms for a 50×50 complex matrix vs. subspace dimension.

APPENDIX
THE EIGENVECTORS OF $\dot{G}$

In this appendix, we derive the formula for the unnormalized ith eigenvector of $\dot{G}$. If the vector $\mathbf{u}$ is the ith eigenvector of $\dot{G}$ multiplied by some scale factor, then from the definition of an eigenvector [10], we get $\dot{G}\mathbf{u} = \mathbf{u}\dot{\sigma}_i$. Using (9), we can write

$$\left(\left[\begin{array}{c|c} \Sigma'^2 & 0 \\ \hline 0 & \hat{\Sigma} \end{array} \right] + Z_{zz} \right) \mathbf{u} = \mathbf{u}\dot{\sigma}_i,$$

where $Z_{zz} = \left[\begin{smallmatrix} 0 & Z \\ Z^H & 0 \end{smallmatrix} \right]$. After rearranging some terms, we get $D\mathbf{u} = -Z_{zz}\mathbf{u}$, where $D = \left(\left[\begin{smallmatrix} \Sigma'^2 & 0 \\ 0 & \hat{\Sigma} \end{smallmatrix} \right] - \dot{\sigma}_i I \right)$. If we deflated the problem by removing unchanged eigenvalues as in [8], [15], then D is invertible, and we can write

$$\mathbf{u} = -D^{-1} Z_{zz} \mathbf{u}. \tag{15}$$

The matrix Z_{zz} is a rank-four Hermitian matrix, and can be written as the sum of four symmetric rank-one matrices,

$$\begin{aligned} Z_{zz} = & \begin{bmatrix} \mathbf{z}_1 \\ 1/2 \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ 1/2 \\ 0 \end{bmatrix}^H - \begin{bmatrix} \mathbf{z}_1 \\ -1/2 \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ -1/2 \\ 0 \end{bmatrix}^H \\ & + \begin{bmatrix} \mathbf{z}_2 \\ 0 \\ 1/2 \end{bmatrix} \begin{bmatrix} \mathbf{z}_2 \\ 0 \\ 1/2 \end{bmatrix}^H - \begin{bmatrix} \mathbf{z}_2 \\ 0 \\ -1/2 \end{bmatrix} \begin{bmatrix} \mathbf{z}_2 \\ 0 \\ -1/2 \end{bmatrix}^H . \end{aligned} \tag{16}$$

If we partition the length $r+2$ vector $\mathbf{u}$ as $[\ \bar{\mathbf{u}} \mid u_{r+1} \mid u_{r+2}\ ]^T$, and substitute this partitioned $\mathbf{u}$ along with (16) into (15), we get

$$\mathbf{u} = -D^{-1} \left(\begin{bmatrix} u_{r+1}\mathbf{z}_1 \\ \mathbf{z}_1^H \bar{\mathbf{u}} \\ 0 \end{bmatrix} + \begin{bmatrix} u_{r+2}\mathbf{z}_2 \\ 0 \\ \mathbf{z}_2^H \bar{\mathbf{u}} \end{bmatrix} \right). \tag{17}$$

If we partition the $r+2 \times r+2$ diagonal matrix D as

$$D = \begin{bmatrix} \bar{D} & 0 & 0 \\ 0 & d_{r+1} & 0 \\ 0 & 0 & d_{r+2} \end{bmatrix},$$

then from (17), we get the two equalities $u_{r+1} = -\mathbf{z}_1^H \bar{\mathbf{u}}/d_{r+1}$ and $u_{r+2} = -\mathbf{z}_2^H \bar{\mathbf{u}}/d_{r+2}$. Substituting for u_{r+1} and u_{r+2} in (17), we get

$$\mathbf{u} = D^{-1} \left(\mathbf{z}_1^H \bar{\mathbf{u}} \begin{bmatrix} \mathbf{z}_1/d_{r+1} \\ -1 \\ 0 \end{bmatrix} + \mathbf{z}_2^H \bar{\mathbf{u}} \begin{bmatrix} \mathbf{z}_2/d_{r+2} \\ 0 \\ -1 \end{bmatrix} \right). \tag{18}$$

where $\mathbf{z}_1^H \bar{\mathbf{u}}$ and $\mathbf{z}_2^H \bar{\mathbf{u}}$ are scalars. If we multiply both sides of the first r elements of (18) by $\mathbf{z}_1^H$ from the left, we get

$$\mathbf{z}_1^H \bar{\mathbf{u}} = \mathbf{z}_1^H \bar{D}^{-1} \left(\frac{\mathbf{z}_1^H \bar{\mathbf{u}}}{d_{r+1}} \mathbf{z}_1 + \frac{\mathbf{z}_2^H \bar{\mathbf{u}}}{d_{r+2}} \mathbf{z}_2 \right). \tag{19}$$

Solving for $\mathbf{z}_2^H \bar{\mathbf{u}}$ in terms of $\mathbf{z}_1^H \bar{\mathbf{u}}$ gives us

$$\mathbf{z}_2^H \bar{\mathbf{u}} = \mathbf{z}_1^H \bar{\mathbf{u}} \left(\frac{d_{r+2}}{d_{r+1}} \right) \left(\frac{d_{r+1} - \mathbf{z}_1^H \bar{D}^{-1} \mathbf{z}_1}{\mathbf{z}_1^H \bar{D}^{-1} \mathbf{z}_2} \right) \tag{20}$$

From (11), we can see that $\dot{w}_1(\dot{\sigma}_i) = d_{r+1} - \mathbf{z}_1^H \bar{D}^{-1} \mathbf{z}_1$ and $\dot{w}_x(\dot{\sigma}_i) = \mathbf{z}_1^H \bar{D}^{-1} \mathbf{z}_2$. Using these identities, and plugging (20) into (18) gives us

$$\mathbf{u} = \left(\frac{\mathbf{z}_1^H \bar{\mathbf{u}}}{d_{r+1}} \right) D^{-1} \left(\begin{bmatrix} \mathbf{z}_1 \\ -d_{r+1} \\ 0 \end{bmatrix} + \frac{\dot{w}_1(\dot{\sigma}_i)}{\dot{w}_x(\dot{\sigma}_i)} \begin{bmatrix} \mathbf{z}_2 \\ 0 \\ -d_{r+2} \end{bmatrix} \right)$$

Because $\mathbf{u}$ is unnormalized, we can discard the scalar $\mathbf{z}_1^H \bar{\mathbf{u}}/d_{r+1}$, and because the $(r+1)$th and $(r+2)$th diagonal elements of D^{-1} are $1/d_{r+1}$ and $1/d_{r+2}$ respectively, they will cancel the d_{r+1} and d_{r+2} in the two vectors, leaving us with (12).

REFERENCES

[1] I. Karasalo, “Estimating the covariance matrix by signal subspace averaging,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 1, pp. 8–12, Feb. 1986.

[2] B. Yang, “Projection approximation subspace tracking,” *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 95–107, 1995.

[3] D. J. Rabideau, “Fast, rank adaptive subspace tracking and applications,” *IEEE Trans. Signal Processing*, vol. 44, no. 9, pp. 2229–2244, 1996.

[4] P. Strobach, “Bi-iteration SVD subspace tracking algorithms,” *IEEE Trans. Signal Processing*, vol. 45, no. 5, pp. 1222–1240, 1997.

[5] E. C. Real, D. W. Tufts, and J. W. Cooley, “Two algorithms for fast approximate subspace tracking,” *IEEE Trans. Signal Processing*, vol. 47, no. 7, pp. 1036–1045, July 1999.

[6] D. J. Rabideau, “Subspace tracking with quasi-rectangular data windows,” in *Proc. SPIE - The International Society for Optical Engineering*, vol. 5096, Apr. 2003, pp. 646–658.

[7] R. Badeau, G. Richard, and B. David, “Sliding window adaptive SVD algorithms,” *IEEE Trans. Signal Processing*, vol. 52, no. 1, pp. 1–10, 2004.

[8] T. M. Toolan, “Advances in sliding window subspace tracking,” Ph.D. dissertation, University of Rhode Island, 2005.

[9] J. W. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1997.

[10] G. H. Golub and C. F. van Loan, *Matrix Computations, 3rd ed.* Baltimore, MD: Johns Hopkins Univ. Press, 1996.

[11] T. M. Toolan and D. W. Tufts, “Improved fast adaptive subspace tracking,” in *Proc. Thirteenth Annual Adaptive Sensor Array Processing Workshop (ASAP)*, June 2005.

[12] B. N. Parlett, *The Symmetric Eigenvalue Problem*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[13] G. H. Golub, “Some modified matrix eigenvalue problems,” *SIAM Rev.*, vol. 15, no. 2, pp. 318–334, Apr. 1973.

[14] D. R. Fuhrmann, “An algorithm for subspace computation with applications in signal processing,” *SIAM J. Matrix Anal. Appl.*, vol. 9, no. 2, pp. 213–220, 1988.

[15] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, “Rank-one modification of the symmetric eigenproblem,” *Numer. Math.*, vol. 31, pp. 31–48, 1978.

[16] M. Gu and S. C. Eisenstat, “A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem,” *SIAM J. Matrix Anal. Appl.*, vol. 15, no. 4, pp. 1266–1276, Oct. 1994.

[17] ——, “Downdating the singular value decomposition,” *SIAM J. Matrix Anal. Appl.*, vol. 16, no. 3, pp. 793–810, July 1995.

[18] D. C. Sorensen and P. T. P. Tang, “On the orthogonality of eigenvectors computed by divide-and-conquer techniques,” *SIAM J. Numer. Anal.*, vol. 28, no. 6, pp. 1752–1775, Dec. 1991.

[19] R. D. DeGroat and R. A. Roberts, “Efficient, numerically stabilized rank-one eigenstructure updating,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 2, pp. 301–316, Feb. 1990.

[20] M. Moonen, P. van Dooren, and J. Vandewalle, “A note on “Efficient numerically stabilized rank-one eigenstructure updating”,” *IEEE Trans. Signal Processing*, vol. 39, no. 8, pp. 1911–1914, Aug. 1991.

[21] T. M. Toolan and D. W. Tufts, “Detection and estimation in non-stationary environments,” in *Proc. IEEE Asilomar Conference on Signals, Systems & Computers*, Nov. 2003, pp. 797–801.