3
Background

This chapter reports main concepts underlying the proposed method, including a brief literature review on the subspace tracking and rank estimation algorithms with specific details on the relevant PAST and row-Householder algorithms.

3.1
Dimensionality reduction

There are many challenges in identifying patterns in high-dimensional data, and dimensionality reduction is one the most efficient strategies to beat the curse of dimensionality. Furthermore, the process of compression itself encapsulates the very essence of learning in the sense that a model can generalize observations into a reduced number of latent variables.

The motivation for this pursue is driven by the intuition that anomalies can be associated to *changes* in the underlying learnt model, i.e. anomalous data drastically affects the latent structure and thus can be detected. Our approach is based on a variant of principal components analysis (PCA), where changes in the correlation structure can be effective monitored in terms of representation accuracy of the observed data.

3.2
Principal component analysis

One of the most common forms of dimensionality reduction is PCA, which is also known in different areas as: the Karhunen-Loève transform, the empirical orthogonal functions and the Hotelling transform. The main idea is to perform dimensionality reduction while preserving as much of the variance from the high-dimensional space as possible, and it has the distinction that it is the *optimal* linear dimensionality reduction technique in the mean-square error sense.

This is achieved by transforming the original data to a new set of variables, the principal components, which are ordered so that the first few retain most of the variation present in the all original variables. It can be shown

[Jolliffe, 2002] that the optimum basis for such transformation correspond to the eigenvectors of the covariance of the data matrix, sorted by the respective eigenvalues.

Dimensionality reduction is actually achieved by choosing only the r principal components (i.e. those with the corresponding r largest eigenvalues), hence the reduced set of basis vectors span the principal subspace of rank r. In this way, linear combinations between dimensions in the original data can be captured and the original set of high dimensional vectors can be effectively compressed into a set of lower dimensional vectors.

Typically, the eigenvectors of the covariance data matrix are computed through the eigenvalue decomposition[1](EVD) but batch approaches are not suitable for the streaming context. Fortunately, there are many alternatives to tackle the eigenproblem in incremental ways, as we will see later.

Regarding the limitations of PCA-based strategies, it is important to note that it is only when we believe that the observed data has a high signal-to-noise ratio that the principal components with larger variance correspond to interesting dynamics and lower ones correspond to noise. Moreover, PCA only finds the independent axes of the data under the Gaussian assumption. For non-Gaussian or multi-modal Gaussian data, PCA simply de-correlates the axes.

3.3
Principal subspace tracking

Recall r is the rank of subspace and N is dimension of the input vector. Since usually $r \ll N$, the subspace tracking algorithms can be classified with respect to their computational complexity: methods requiring $\mathcal{O}(N^2 r)$ or $\mathcal{O}(N^2)$ will be classified as *high* complexity; $\mathcal{O}(Nr^2)$ as *medium* complexity and finally those with $\mathcal{O}(Nr)$ as *low* complexity. The algorithms in the last class are called *fast subspace trackers* and they are most suitable for real-time computing – thus our main subject of interest. The article of Comon and Golub [1990] constitutes a review of the results up to 1990, treating the first two classes, since the last class was not available at the time. The most complete reviews for the fast subspace trackers are available from Doukopoulos and Moustakides [2008], Doukopoulos [2004] and Strobach [2009a].

[1]Since the singular value decomposition (SVD) handles more general matrices, it can surely be used for the problem as well.

3.3.1
Problem definition

In the context of data streams, the entire data covariance matrix is not available at once and therefore the goal in this stochastic case is the recursive estimation of the principal subspace of the time-recursively updated covariance matrix $\mathbf{\Phi}(t)$ of dimension $N \times N$,

$$\mathbf{\Phi}(t) = \alpha\mathbf{\Phi}(t-1) + \boldsymbol{z}(t)\boldsymbol{z}(t)^\top \tag{3-1}$$

where $0 < \alpha \leq 1$ is a forgetting factor that accommodates for concept drifts and gives higher emphasis on more recent measurements. The input vector $\boldsymbol{z}(t) \in \Re^N$ is the streaming data from Definition 1.

A straightforward way to estimate the subspace is to apply eigenvalue decomposition (EVD) on the $\mathbf{\Phi}(t)$ at every step t. The eigenvalue decomposition can be expressed as follows:

$$\mathbf{\Phi}(t) = \boldsymbol{V}_r(t)\mathbf{\Lambda}_r\boldsymbol{V}_r^\top(t) + \boldsymbol{V}_{N-r}(t)\mathbf{\Lambda}_{N-r}\boldsymbol{V}_{N-r}^\top(t). \tag{3-2}$$

We are interested in the part of this decomposition associated with the principal r eigenvectors:

$$\mathbf{\Phi}_r(t) = \mathbf{\Phi}(t)\boldsymbol{V}_r(t)\boldsymbol{V}_r^\top(t) \tag{3-3}$$

which is obtained by right-multiplication of the covariance with the projection matrix that projects vector orthogonally onto the principal subspace (rank r) spanned by $\boldsymbol{V}_r(t)$.

The direct EVD approach typically requires $\mathcal{O}(N^3)$ operations but approximated schemes were devised to require much less operations. Most of the online algorithms only track the projection matrix without tracking the dominant eigenvectors and eigenvalues directly, that is why they are more precisely defined as principal subspace trackers (i.e. rather than eigenspace trackers). The solution for the problem of finding this orthogonal projector of rank r $\mathbf{\Pi}(t) = \boldsymbol{V}_r(t)\boldsymbol{V}_r^\top(t)$ is equivalent to that of maximizing the generalized Rayleigh quotient $\operatorname{tr}\{\mathbf{\Phi}_r(t)\mathbf{\Pi}(t)\}$ Borga and Borga [1998].

3.3.2
Orthogonal iteration principle

Most (if not all) fast subspace trackers can be analyzed in terms of the orthogonal iteration, which is a generalization of the power method used to compute the eigenvector with largest eigenvalue. The Owsley algorithm [Owsley, 1978] was one of the first applications of the idea in subspace tracking, where a single orthogonal iteration is applied in each time step as follows:

$$\boldsymbol{A}(t) = \boldsymbol{\Phi}(t)\boldsymbol{Q}(t-1) \tag{3-4}$$

$$\boldsymbol{A}(t) = \boldsymbol{Q}(t)\boldsymbol{S}(t) \quad \text{: orthonormal factorization} \tag{3-5}$$

where $\boldsymbol{A}(t)$ is an auxiliary matrix $N \times r$. Owsley used a QR-factorization in (3-5), and in this case, $\boldsymbol{Q}(t)$ is the matrix of estimated dominant eigenvectors and $\boldsymbol{S}(t)$ is an upper-right triangular $r \times r$ matrix with estimated eigenvalues appearing on the main diagonal in descending order. We know from the literature that a batch orthonormal factorization cannot be carried out with less than $\mathcal{O}(Nr^2)$ operations [Golub and Van Loan, 1996], and Strobach [2009a] notes that if only a basis of the principal subspace is sought, then the shape of the square $r \times r$ S-matrix can be left completely undetermined. Therefore, step (3-5) reduces to a QS-decomposition, which can be updated more efficiently than the more restrictive QR-decomposition. This allow most methods to move from $\mathcal{O}(Nr^2)$ to $\mathcal{O}(Nr)$, but the tracked basis vectors in $\boldsymbol{Q}(t)$ will no longer be the tracked eigenvectors – they become an arbitrarily rotated variant of the estimated eigenvector set. This is not a disadvantage, because in most applications one is only interested in tracking the projection matrix, namely

$$\boldsymbol{Q}(t)\boldsymbol{Q}(t)^\top \approx \boldsymbol{V}(t)\boldsymbol{V}(t)^\top \tag{3-6}$$

Strobach [2009a] lays out the following recurrence equation for the auxiliary A-matrix from (3-4) which subsequently allows the discussion of the subspace trackers in terms of its orthonormal factorization:

$$\boldsymbol{A}(t) = \alpha\boldsymbol{A}(t-1)\boldsymbol{\Theta}(t-1) + \alpha\boldsymbol{\Phi}(t-1)\boldsymbol{\Delta}(t-1) + \boldsymbol{z}(t)\boldsymbol{h}^\top(t) \tag{3-7}$$

where

$$\boldsymbol{h}(t) = \boldsymbol{Q}(t-1)^\top\boldsymbol{z}(t) \tag{3-8}$$

is a step common to all fast subspace trackers and $\boldsymbol{h}(t)$ is a r-vector that represents the *latent variables* from the previous subspace model. The matrix $\boldsymbol{\Theta}(t)$ plays the role of a state transition matrix (as introduced in Strobach [1996])

$$\boldsymbol{Q}(t) = \boldsymbol{Q}(t-1)\boldsymbol{\Theta}(t) + \boldsymbol{\Delta}(t) \tag{3-9}$$

and $\boldsymbol{\Delta}(t)$ is an innovations matrix that completes the subspace propagation model satisfying

$$\boldsymbol{Q}^{\top}(t-1)\boldsymbol{\Delta}(t) = \boldsymbol{0} \tag{3-10}$$

hence we can write

$$\boldsymbol{\Theta}(t) = \boldsymbol{Q}(t-1)^{\top}\boldsymbol{Q}(t) \tag{3-11}$$

The innovations term is commonly neglected in all fast subspace trackers, except in the latest row-Householder approach [Strobach, 2008, 2009a]. When neglected, the subspace tracker's goal from step (3-5) can be stated according to the updating of the orthonormal factorization of the A-matrix

$$\boldsymbol{Q}(t)\boldsymbol{S}(t) = \alpha\boldsymbol{Q}(t-1)\boldsymbol{S}(t-1)\boldsymbol{\Theta}(t-1) + \boldsymbol{z}(t)\boldsymbol{h}^{\top}(t) \tag{3-12}$$

3.3.3 PAST

The Projection Approximation Subspace Tracking (PAST) algorithm, originally proposed in Yang [1995a], is probably the best known approach for tracking the principal subspace and it was originally derived by minimizing a *simplification* of the Principal Component Analysis criterion:

$$\sum_{t=1}^{T}\alpha^{T-t}\|\boldsymbol{z}(t) - \boldsymbol{Q}(t)\boldsymbol{h}(t)\|^2 \tag{3-13}$$

In particular, the expectation operator is replaced with the exponentially weighted sum and approximating $\boldsymbol{Q}(t)\boldsymbol{z}(t)$ with $\boldsymbol{Q}(t-1)\boldsymbol{z}(t)$ and substituting for (3-8). This approximated cost function is now quadratic (rather than fourth-order) and recursive least square is applied in order to derive the PAST algorithm as summarized in the listing from Figure 3.1.

In Strobach [2009a], PAST is derived by pre-multiplication of (3-12) by $\boldsymbol{Q}(t-1)^{\top}$ and assuming both *a posteriori* $\boldsymbol{\Theta}(t-1) = \boldsymbol{I}$ and *a priori* $\boldsymbol{\Theta}(t) = \boldsymbol{I}$ projection approximations. These approximations lead to the basic recurrence of the S-matrix, that underlies the PAST algorithm:

$$\boldsymbol{S}(t) = \alpha\boldsymbol{S}(t-1) + \boldsymbol{h}(t)\boldsymbol{h}^{\top}(t) \tag{3-15}$$

The recurrence for the orthonormal Q-matrix is obtained by post-multiplication of the simplified version of (3-12) with $\boldsymbol{S}^{-1}(t)$. If we define $\boldsymbol{P}(t) = \boldsymbol{S}^{-1}(t)$, one can apply the matrix inversion lemma to obtain (3-14f), thus arriving at the same procedure for updating $\boldsymbol{Q}(t)$ as outlined in Figure 3.1. For this reason, PAST-type subspace trackers are denominated as S^{-1}-domain

$$\boldsymbol{Q}(0) \leftarrow \text{random orthonormal} \tag{3-14a}$$
$$\boldsymbol{P}(0) \leftarrow \sigma^{-1}\boldsymbol{I}, \text{ where } \sigma \text{ is a small positive constant} \tag{3-14b}$$
$\texttt{for}\ t = 1, 2, \ldots$
$$\boldsymbol{h}(t) = \boldsymbol{Q}^{\top}(t-1)\boldsymbol{z}(t) \tag{3-14c}$$
$$\boldsymbol{g}(t) = \boldsymbol{P}(t-1)\boldsymbol{h}(t) \tag{3-14d}$$
$$\boldsymbol{f}(t) = \frac{\boldsymbol{g}(t)}{\alpha + \boldsymbol{h}(t)\boldsymbol{g}(t)} \tag{3-14e}$$
$$\boldsymbol{P}(t) = \frac{1}{\alpha}[\boldsymbol{P}(t-1) - \boldsymbol{f}(t)\boldsymbol{g}(t)^{\top}] \tag{3-14f}$$
$$\boldsymbol{z}_{\perp}(t) = \boldsymbol{z}(t) - \boldsymbol{Q}(t-1)\boldsymbol{h}(t) \tag{3-14g}$$
$$\boldsymbol{Q}(t) = \boldsymbol{Q}(t-1) + \boldsymbol{z}_{\perp}(t)\boldsymbol{f}^{\top}(t) \tag{3-14h}$$

Figure 3.1: PAST algorithm.

techniques. Under this framework, one can explain the widely acknowledged loss of orthonormality of the subspace basis matrix $\boldsymbol{Q}(t)$ in PAST in terms of the *a priori* projection approximation that violates the subspace propagation model of (3-9).

In the same influential article [Yang, 1995a], PASTD is presented as a variant of the PAST algorithm using the deflation technique to sequentially estimate the dominant eigenvectors. A complete derivation of PASTD from PAST can be seen in [Ronnie Landqvist, 2005], but here we only display the algorithm in Figure 3.3.3. Yang highlights that the update formula for the one-vector is identical (see 3-16f), except for the step size, to the Oja learning rule, which was designed for extracting the first principal component by means of a single linear unit neural network [Oja et al., 1995]. The connection to biological Hebbian learning is very exciting and is cited in the work of Weng et al. [2003].

$$\left[\boldsymbol{q}_1(0), \ldots, \boldsymbol{q}_r(0)\right] \leftarrow \text{random orthonormal} \tag{3-16a}$$
$\texttt{for}\ t = 1, 2, \ldots$
$$\boldsymbol{z}_1(t) = \boldsymbol{z}(t) \tag{3-16b}$$
$\texttt{for}\ i = 1, 2, \ldots$,r
$$h_i(t) = \boldsymbol{q}_i^{\top}(t-1)\boldsymbol{z}_i(t) \tag{3-16c}$$
$$d_i(t) = \alpha d_i(t-1) + h(t)^2 \tag{3-16d}$$
$$\boldsymbol{e}_i(t) = \boldsymbol{z}_i(t) - \boldsymbol{q}_i(t-1)h_i(t) \tag{3-16e}$$
$$\boldsymbol{q}_i(t) = \boldsymbol{q}_i(t-1) + \boldsymbol{e}_i(t)h_i(t)/d_i(t) \tag{3-16f}$$
$$\boldsymbol{z}_{i+1}(t) = \boldsymbol{z}_i(t) - \boldsymbol{q}_i(t)h_i(t) \tag{3-16g}$$

Figure 3.2: PASTd algorithm.

3.3.4
Recursive row-Householder subspace tracking

The derivation of the algorithm [Strobach, 2009a] follows from the decomposition of the data vector $\boldsymbol{z}(t)$ (recall $\boldsymbol{z}_\perp(t)$ from (3-14g)):

$$\boldsymbol{z}(t) = Z^{1/2}(t)\bar{\boldsymbol{z}}_\perp(t) + \boldsymbol{Q}(t-1)\boldsymbol{h}(t) \tag{3-17}$$

where

$$\bar{\boldsymbol{z}}_\perp(t) = Z^{-1/2}(t)\boldsymbol{z}_\perp(t) \tag{3-18}$$

and

$$Z(t) = \boldsymbol{z}_\perp^\top(t)\boldsymbol{z}_\perp(t) \tag{3-19}$$

Substituting (3-17) into (3-12) yields the following updating expression

$$\boldsymbol{Q}(t)\boldsymbol{S}(t) = \begin{bmatrix}\boldsymbol{Q}(t-1) & \bar{\boldsymbol{z}}_\perp(t)\end{bmatrix} \times \begin{bmatrix}\alpha\boldsymbol{S}(t-1)\boldsymbol{\Theta}(t-1) + \boldsymbol{h}(t)\boldsymbol{h}^\top T(t) \\ Z^{1/2}(t)\boldsymbol{h}^\top(t)\end{bmatrix} \tag{3-20}$$

which is the general structure of the LORAF-type or S-domain techniques. For the original LORAF algorithms, a set of rotations were constructed to enable the recursive update of the Q-matrix. In the new algorithm, a row-Householder reduction is performed as there is no longer the triangular restriction on structure of the S-matrix. The prototype problem of designing a row-Householder reflection $\boldsymbol{H} = \boldsymbol{I} - 2\underline{\boldsymbol{v}}\underline{\boldsymbol{v}}^\top$ where $\underline{\boldsymbol{v}} = \begin{bmatrix}\boldsymbol{v} \\ \varphi\end{bmatrix}$ with $\|\underline{\boldsymbol{v}}\| = 1$ and $\boldsymbol{H}\boldsymbol{H} = \boldsymbol{I}$ has the goal to annihilate the row vector in the appended S-matrix of dimension $(r+1)\times r$ as follows:

$$\begin{bmatrix}\boldsymbol{S}(t) \\ 0\ldots0\end{bmatrix} = \boldsymbol{H}(t)\begin{bmatrix}\alpha\boldsymbol{S}(t-1)\boldsymbol{\Theta}(t-1) + \boldsymbol{h}(t)\boldsymbol{h}^\top(t) \\ Z^{1/2}(t)\boldsymbol{h}^\top(t)\end{bmatrix} \tag{3-21}$$

and the Q-matrix may be updated accordingly

$$\begin{bmatrix}\boldsymbol{Q}(t) & \boldsymbol{z}_q(t)\end{bmatrix} = \begin{bmatrix}\boldsymbol{Q}(t-1) & \bar{\boldsymbol{z}}_\perp(t)\end{bmatrix}\boldsymbol{H}(t) \tag{3-22}$$

The appended column $\boldsymbol{z}_q(t)$ is not interesting and not computed explicitly. The complete derivation is omitted here and we just report the quasicode listing for the fast recursive row-Householder subspace tracking [from Strobach, 2009a] in Figure 3.3. The procedure is very elegant and precise, since the energy in the $(r+1)\times r$ can be transformed into a smaller submatrix of dimension $r\times r$ and the Householder transformation acts essentially as a lossless compressor. The reader can refer to Appendix A for more details.

$\boldsymbol{Q}(0) \leftarrow$ random orthonormal
$\boldsymbol{S}(0) \leftarrow \sigma\boldsymbol{I}$, where σ is a small positive constant
$\boldsymbol{v}(0) \leftarrow \boldsymbol{0}$
`for` $t = 1, 2, \ldots$

$$\boldsymbol{h}(t) = \boldsymbol{Q}^\top(t-1)\boldsymbol{z}(t) \tag{3-23a}$$

$$Z(t) = \boldsymbol{z}^\top(t)\boldsymbol{z}(t) - \boldsymbol{h}^\top T(t)\boldsymbol{h}(t) \tag{3-23b}$$

$$\boldsymbol{u}(t-1) = \boldsymbol{S}(t-1)\boldsymbol{v}(t-1) \tag{3-23c}$$

$$\boldsymbol{X}(t) = \alpha\boldsymbol{S}(t-1) + \boldsymbol{h}(t)\boldsymbol{h}^\top(t) - 2\alpha\psi(t-1)\boldsymbol{u}(t-1)\boldsymbol{v}^\top(t-1) \tag{3-23d}$$

$$\boldsymbol{X}^\top(t)\boldsymbol{b}(t) = Z^{1/2}\boldsymbol{h}(t) \xrightarrow{\text{solve}} \boldsymbol{b}(t) \tag{3-23e}$$

$$\beta(t) = 4(\boldsymbol{b}^\top(t)\boldsymbol{b}(t) + 1) \tag{3-23f}$$

$$\varphi^2(t) = \frac{1}{2} + \frac{1}{\sqrt{\beta(t)}} \tag{3-23g}$$

$$\gamma(t) = \frac{1 - 2\varphi^2(t)}{2\varphi(t)} \tag{3-23h}$$

$$\delta(t) = \frac{\varphi(t)}{Z^{1/2}(t)} \tag{3-23i}$$

$$\boldsymbol{v}(t) = \gamma(t)\boldsymbol{b}(t) \tag{3-23j}$$

$$\boldsymbol{S}(t) = \boldsymbol{X}(t) - \frac{1}{\delta(t)}\boldsymbol{v}(t)\boldsymbol{h}^\top(t) \tag{3-23k}$$

$$\boldsymbol{w}(t) = \delta(t)\boldsymbol{h}(t) - \boldsymbol{v}(t) \tag{3-23l}$$

$$\boldsymbol{e}(t) = \delta(t)\boldsymbol{z}(t) - \boldsymbol{Q}(t-1)\boldsymbol{w}(t) \tag{3-23m}$$

$$\boldsymbol{Q}(t) = \boldsymbol{Q}(t-1) - 2\boldsymbol{e}(t)\boldsymbol{v}^\top(t) \tag{3-23n}$$

Figure 3.3: Fast Recursive row-Householder subspace tracking algorithm.

After further theoretical inspection of the previously neglected term $\boldsymbol{\Phi}(t-1)\boldsymbol{\Delta}(t-1)$ from (3-7), a value of $\psi = -1$ into (3-23d) is found to be nearly optimal [Strobach, 2009a]. A parameter choice of $\psi = +1$ corresponds to estimates from methods like LORAF2 and FAPI and a parameter of $\psi = 0$, which simplifies to the LORAF3 model, is claimed to be safe to use as long as subspace tracking is the only issue. A second variant of the algorithm was proposed in order to directly track the true eigenvalues and eigenvectors, where few steps involving QR-decomposition can be combined to replace the linear system solver from step (3-23e), but it is not needed for our main problem.

Method	**Complexity**	**Stable**
Fast recursive row-Householder [Strobach, 2009a]	$3Nr + O(r^3)$	Yes
FDPM [Doukopoulos and Moustakides, 2008]	$6Nr + O(r)$	Yes
YAST [Badeau et al., 2008]	$5Nr + O(r^2)$	Yes
FAPI [Badeau et al., 2005]	$3Nr + O(r^2)$	Yes
NIC [Miao and Hua, 1998]	$4Nr + O(r^2)$	No
LORAF3 [Strobach, 1996]	$4Nr + O(r^2)$	Yes
PAST [Yang, 1995a]	$3Nr + O(r^2)$	No
PASTd [Yang, 1995a]	$4Nr + O(r)$	No
FRANS [Yang and Kaveh, 1988]	$3Nr + O(r)$	No

Table 3.1: State-of-the-art fast principal subspace trackers.

3.3.5 Summary on subspace trackers

The most relevant algorithms for tracking the principal subspace with complexity dominated by $\mathcal{O}\big(Nr\big)$ flops[2] are portrayed in Table 3.1.

The performance of these various methods is generally evaluated according to several criteria: computational complexity, stability (ability to guarantee orthonormal estimates for the projection matrix $\boldsymbol{Q}(t)$ without diverging) and quality of the estimated subspace according to the true eigenvectors. FAPI (as well as NIC) are based on PAST and have the inherent drawback in terms of numerically robustness due the updating of the inverse power matrix – the situation is literally the same as with the classical RLS adaptive filters. From adaptive filtering literature, we known that S-domain techniques are generally preferable than S^{-1}-domain techniques, if the complexity are similar [Haykin, 1996]. FRANS, FDPM and YAST are also minor subspace trackers. FRANS generalizes the methods proposed in Owsley [1978] using a gradient-descent technique on the Rayleigh quotient mentioned in Section 3.3.1 but it is shown to be unstable [Yang et al., 2008]. HFRANS [Attallah, 2003] was proposed for the minor subspace problem by improving the numerical stability through a Householder transformation. FDPM builds on the HFRANS algorithm to provide both principal and minor subspace estimations with a fast convergence

[2] A flop is a multiply / accumulate (mac) operation. One division is considered one flop.

towards orthonormality, but the awkward step size parameter remains to be tuned. In fact, FRANS and FDPM can be seen as gradient-descent techniques similar to Oja's rule and they inherit the ability to estimate both principal and minor subspace by simply reversing the sign of the learning step parameter. In [Strobach, 2009a], FDPM is explicitly not recommended because it produces degraded subspace estimates due to various heuristical simplifications of the S-matrix. YAST is a faster implementation of the SP algorithm proposed in Davila [2000] that employs similar orthogonalization from the FDPM subspace tracker. It has high complexity in general terms but it is shown here for completeness: it relies on the a shift invariance property in the data, otherwise it is dominated by $\mathcal{O}\big(N^2\big)$ operations and it further relies on the eigendecomposition of a $r+1 \times r+1$ matrix (a inner procedure based on conjugate gradient is offered) which also adds $\mathcal{O}\big(r^3\big)$ flops in general.

LORAF3 assumes $\boldsymbol{\Theta}(t-1) = \boldsymbol{I}$ and applies a sequence of Givens rotations for the orthonormal decomposition direct from (3-12) with $\boldsymbol{S}(t)$ constrained to an upper right triangular matrix, so the algorithm tracks the eigenvectors and eigenvalues directly. LORAF3's estimates are more accurate than methods like FAPI. The new recursive row-Householder method [Strobach, 2009a] follows similar shape from the classical LORAF3 but is motivated by achieving a lower dominant complexity of $3Nr$ by relaxing the upper trapezoidal shape constraint on $\boldsymbol{S}(t)$ for strictly tracking the principal subspace basis. The Householder reflection is very common in the batch algorithms such as in recent implementations of QR decomposition, and a gradient-based algorithm [Douglas, 2000] was presented for adapting the a Householder reflector in the context of subspace trackers. The new row-Householder tracker achieves more exact estimates because fewer projection approximations are made and the entire model of (3-7) is handled. In the same article, a second variant of the algorithm with same complexity shows how to obtain the real eigenvalues and eigenvectors from the tracked basis. The $\mathcal{O}\big(r^3\big)$ term in the total complexity comes from solving a internal system of linear equation in (3-23e) (or from a QR step in the second version). The author suggests that the $\mathcal{O}\big(r^3\big)$ can be brought down to $\mathcal{O}\big(r^2\big)$ by exploiting the inherent recursive 'rank one' updates for the case where the real eigenvalues are not tracked directly. However, the complexity of $\mathcal{O}\big(r^3\big)$ flops for the QR step seems unavoidable in the second variant. Nevertheless, no other low complexity algorithm is known to provide such exact estimates.

It has been postulated that the operations count for a fast orthogonal iteration subspace tracker providing strictly orthonormal principal subspace basis estimates is lower bounded by a dominant complexity of $3Nr$ operations per time update. The reasoning comes from the following three necessary distinct Nr layers that are decoupled, where none can be substituted by any combination of the other two:

- input data compression step according to (3-23a) or (3-14c)
- extraction of the innovation in the actual data snapshot (projection error) according to (3-23m) or (3-14g)
- updating of the Q-matrix according to (3-23n) or (3-14h)

3.3.6
Summary on rank estimation

A disadvantage of nearly all the subspace tracking algorithms is that they require knowledge of r. For instance, none of the algorithms in Table 3.1 estimate the subspace rank. In relative terms, little attention has been devoted to estimating r in an unsupervised manner. PASTD has been extended in Yang [1995b] to provide subspace rank estimates using information theoretic criteria (Akaike information criterion and minimum description length), while SPIRIT [Papadimitriou et al., 2005] extends PASTD to allow rank discovery using energy thresholding given a reconstruction error target which is a common method to determine the number of principal components in PCA [Jolliffe, 2002]. They are nearly identical to PASTD, hence are classified as low complexity algorithms with the same computational complexity and same disadvantages. The KY method [Kavcic and Yang, 1996] is a general purpose adaptive rank estimator and it has been show to typically outperform the information theoretical criteria from [Yang, 1995b]. Champagne et al. [2003] improves the KY algorithm by using different adaptive thresholds for rank increase and decrease tests which reduce the detection time of a rank decrease. In [Perry and Wolfe, 2009], most of the previous methods are criticized by relying on tuning of parameters that depend on the actual data and points to the literature on random matrix theory as a best way to proceed. Cross-validation-based approaches have been suggested [Owen and Perry, 2009] but do not apply to real-time subspace tracking. Kritchman and Nadler [2008] provides a survey of possible approaches with recent results from random matrix theory regarding the behaviour of noise eigenvalues, ultimately recommending one based on eigenvalue thresholding. The idea from this algorithm, called KN, is further developed in [Perry and Wolfe, 2009] that derives a selection rule that minimizes the maximum risk under a set of suitable alternate models.