N91-17090

Downdating a Time-Varying Square Root Information Filter

Ronald J. Muellerschoen †

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

Abstract

A new method to efficiently downdate an estimate and covariance generated by a discrete time Square Root Information Filter (SRIF) is presented. The method combines the QR factor downdating algorithm of Gill [3] and the decentralized SRIF algorithm of Bierman [4]. Efficient removal of either measurements or *a priori* information is possible without loss of numerical integrity. Moreover, the method includes features for detecting potential numerical degradation. Performance on a 300 parameter system with 5800 data points shows that the method can be used in real time and hence is a promising tool for interactive data analysis. Additionally, updating a time-varying SRIF filter with either additional measurements or *a priori* information proceeds analogously.

Introduction

A typical 24 hour data arc for the GPS demonstration on TOPEX/POSEIDON [6] will contain $\approx$ 30,000 data points. To process these data points on a VAX 8530, a sequential SRIF filter will require nearly 3 hours of CPU time. Upon processing, should outliers be discovered in the data, due to cycle slips, atmospheric fluctuations, multipath, *etc* ..., they must be removed from the data arc and another 3 hours of CPU time would then be needed for reprocessing. The method presented here permits efficient removal of these outlying data points without reprocessing the entire data set.

The discussion is organized in the following manner. First, a decentralized approach to updating a time-varying SRIF with a single measurement is presented. From this development, it is evident how to remove a measurement from a time-varying SRIF. Next, the one-component-at-a-time process noise methodology is applied in the actual implementation. The necessary linear combination of the data equation and the smoothing coefficients is then presented. Finally, the efficiency and the numerical integrity of the method is discussed.

Updating a Time-Varying SRIF with a Single Measurement

The goal is to find estimates $(x_0, x_1, \ldots, x_N)$ to minimize the the least-squares performance functional

$$J^N(x_0, x_1, \ldots, x_N) = \|\tilde{R}_x(0)x_0 - \tilde{z}_x(0)\|^2 + \sum_{j=0}^{N-1} \|R_w(j)w_j\|^2 + \sum_{j=0}^{N} \|A_j x_j - z_j\|^2 + \|H_k x_k - y_k\|^2 \tag{1}$$

† Member of Technical Staff, Navigation Systems

for the linear discrete-time dynamic model

$$x_{j+1} = \phi_j x_j + G w_j \qquad j = 0, N-1 \tag{2}$$

with measurements

$$z_j = A_j x_j + \nu_j \qquad j = 0, N \tag{3}$$

and the single measurement

$$y_k = H_k x_k + \nu_k \qquad 0 \le k \le N \tag{4}$$

where $\widetilde{()}_x(j)$ represents *a priori* information between time updates j and j+1. The data noise ν_j and process noise w_j are independent mean zero white Gaussian noise processes with covariances I and $R_w(j)^{-1}R_w(j)^{-T}$, respectively. Of note, when a pseudo-epoch state formulation is used ϕ_j and G have a simplified structure.

$$\begin{pmatrix} x_{j+1} \\ p_{j+1} \end{pmatrix} = \begin{pmatrix} I & V_p(j) \\ 0 & M_j \end{pmatrix} \begin{pmatrix} x_j \\ p_j \end{pmatrix} + \begin{pmatrix} 0 \\ I \end{pmatrix} w_j \qquad j = 0, N-1 \tag{5}$$

Ignoring the additional data equation $y_k = H_k x_k + \nu_k$, the SRIF alternates between measurement updates

$$\begin{aligned} J^N(x_0, x_1, \ldots, x_N) = & \sum_{i=0}^{j-1} \|e_i\|^2 + \| \begin{pmatrix} \tilde{R}_x(j) \\ A_j \end{pmatrix} x_j - \begin{pmatrix} \tilde{z}_x(j) \\ z_j \end{pmatrix} \|^2 + \\ & \sum_{i=j+1}^{N} \|A_i x_i - z_i\|^2 + \|H_k x_k - y_k\|^2 + \\ & \sum_{i=0}^{j-1} \|R_w^*(i) w_i + R_{wx}^*(i+1) x_{i+1} - z_w^*(i)\|^2 + \sum_{i=j}^{N-1} \|R_w(i) w_i\|^2 \end{aligned} \tag{6}$$

$$\begin{aligned} = & \sum_{i=0}^{j} \|e_i\|^2 + \|\hat{R}_j x_j - \hat{z}_j\|^2 + \\ & \sum_{i=j+1}^{N} \|A_i x_i - z_i\|^2 + \|H_k x_k - y_k\|^2 + \\ & \sum_{i=0}^{j-1} \|R_w^*(i) w_i + R_{wx}^*(i+1) x_{i+1} - z_w^*(i)\|^2 + \sum_{i=j}^{N-1} \|R_w(i) w_i\|^2 \end{aligned} \tag{7}$$

for $j = 0, N$ and time updates

$$\begin{aligned} J^N(x_0, x_1, \ldots, x_N) = & \| \begin{pmatrix} R_w(j) & 0 \\ -\hat{R}_j \phi_j^{-1} G & \hat{R}_j \phi_j^{-1} \end{pmatrix} \begin{pmatrix} w_j \\ x_{j+1} \end{pmatrix} - \begin{pmatrix} 0 \\ \hat{z}_j \end{pmatrix} \|^2 + \\ & \sum_{i=0}^{j-1} \|R_w^*(i) w_i + R_{wx}^*(i+1) x_{i+1} - z_w^*(i)\|^2 + \sum_{i=j+1}^{N-1} \|R_w(i) w_i\|^2 + \end{aligned}$$

$$\sum_{i=j+1}^{N} ||A_i x_i - z_i||^2 + ||H_k x_k - y_k||^2 + \sum_{i=0}^{j} ||e_i||^2 \tag{8}$$

$$\begin{aligned}
&= ||\tilde{R}_x(j+1)x_{j+1} - \tilde{z}_x(j+1)||^2 + \\
&\sum_{i=0}^{j} ||R^*_w(i)w_i + R^*_{wx}(i+1)x_{i+1} - z^*_w(i)||^2 + \sum_{i=j+1}^{N-1} ||R_w(i)w_i||^2 + \\
&\sum_{i=j+1}^{N} ||A_i x_i - z_i||^2 + ||H_k x_k - y_k||^2 + \sum_{i=0}^{j} ||e_i||^2
\end{aligned} \tag{9}$$

for $j = 0, N-1$. The e_i term represents the residual sum of squares from the i'th mesurement update; the $()^*$ terms represent the smoothing coefficients. Additionally, the notation $\widehat{()}_j$ represents quantities that include data up to time $j+1$.

To update the filter with the additional data equation $y_k = H_k x_k + \nu_k$, it is necessary to express this equation in terms of x_{k+1} and w_k.

$$\begin{aligned}
||H_k x_k - y_k||^2 &= || \begin{pmatrix} -H_k \phi_k^{-1} G & H_k \phi_k^{-1} \end{pmatrix} \begin{pmatrix} w_k \\ x_{k+1} \end{pmatrix} - y_k ||^2 \\
&= || \begin{pmatrix} H^*_w(k) & H^*_{wx}(k+1) \end{pmatrix} \begin{pmatrix} w_k \\ x_{k+1} \end{pmatrix} - y^*(k) ||^2
\end{aligned} \tag{10}$$

This equation is then merged with the smothing coefficients of the SRIF.

$$\begin{aligned}
J^N(x_0, x_1, \ldots, x_N) = &\sum_{j=0}^{N} ||e_j||^2 + ||\hat{R}_N x_N - \hat{z}_N||^2 + \\
&\sum_{j=0}^{k-1} ||R^*_w(j)w_j + R^*_{wx}(j+1)x_{j+1} - z^*_w(j)||^2 + \\
&\sum_{j=k+1}^{N-1} ||R^*_w(j)w_j + R^*_{wx}(j+1)x_{j+1} - z^*_w(j)||^2 + \\
&|| \begin{pmatrix} R^*_w(k) & R^*_{wx}(k+1) \\ H^*_w(k) & H^*_{wx}(k+1) \end{pmatrix} \begin{pmatrix} w_k \\ x_{k+1} \end{pmatrix} - \begin{pmatrix} z^*_w(k) \\ y^*(k) \end{pmatrix} ||^2
\end{aligned} \tag{11}$$

$$\begin{aligned}
J^N(x_0, x_1, \ldots, x_N) = &\sum_{j=0}^{N} ||e_j||^2 + ||\hat{R}_N x_N - \hat{z}_N||^2 + \\
&\sum_{j=0}^{k-1} ||R^*_w(j)w_j + R^*_{wx}(j+1)x_{j+1} - z^*_w(j)||^2 + \\
&\sum_{j=k+1}^{N-1} ||R^*_w(j)w_j + R^*_{wx}(j+1)x_{j+1} - z^*_w(j)||^2 + \\
&|| \begin{pmatrix} R^{**}_w(k) & R^{**}_{wx}(k+1) \\ 0 & H_{k+1} \end{pmatrix} \begin{pmatrix} w_k \\ x_{k+1} \end{pmatrix} - \begin{pmatrix} z^{**}_w(k) \\ y_{k+1} \end{pmatrix} ||^2
\end{aligned} \tag{12}$$

The $()^{**}$ terms replace the old smoothing coefficients and $y_{k+1} = H_{k+1}x_{k+1} + \nu_{k+1}$ is the ensuing additional data equation. The time updates in equation (10) and the merges in equations (11) and (12) are then repeated for $k+2, k+3, \ldots, N$. In the end, the terminal data equation $y_N = H_N x_N + \nu_N$ is then merged with the terminal SRIF array.

$$
\begin{aligned}
J^N(x_0, x_1, \ldots, x_N) = & \sum_{j=0}^{k-1} \|R^*_w(j)w_j + R^*_{wx}(j+1)x_{j+1} - z^*_w(j)\|^2 + \\
& \sum_{j=k}^{N-1} \|R^{**}_w(j)w_j + R^{**}_{wx}(j+1)x_{j+1} - z^{**}_w(j)\|^2 + \\
& \left\| \begin{pmatrix} \hat{R}_N \\ H_N \end{pmatrix} x_N - \begin{pmatrix} \hat{z}_N \\ y_N \end{pmatrix} \right\|^2 + \sum_{j=0}^{N} \|e_j\|^2
\end{aligned} \qquad (13)
$$

$$
\begin{aligned}
J^N(x_0, x_1, \ldots, x_N) = & \sum_{j=0}^{k-1} \|R^*_w(j)w_j + R^*_{wx}(j+1)x_{j+1} - z^*_w(j)\|^2 + \\
& \sum_{j=k}^{N-1} \|R^{**}_w(j)w_j + R^{**}_{wx}(j+1)x_{j+1} - z^{**}_w(j)\|^2 + \\
& \|\check{R}_N x_N - \check{z}_N\|^2 + \sum_{j=0}^{N} \|e_j\|^2 + \|\check{e}_N\|^2
\end{aligned} \qquad (14)
$$

The notation $\check{()}_N$ represents quantities resulting from merging the terminal data equation with the terminal SRIF array. The filter estimate and covariance are then $\check{R}_N^{-1}\check{z}_N$ and $\check{R}_N^{-1}\check{R}_N^{-T}$, respectively. To obtain the smoothed estimates and covariances, the Dyer-McReynolds smoother [2] would then operate with the $()^*$ terms for $j = 0, k-1$ and with the $()^{**}$ terms for $j = k, N-1$.

Downdating a Time-Varying SRIF with a Single Measurement

The last lines of equations (11) and (12) contain the key to reversing this process. If the data equation $y_k = H_k x_k + \nu_k$ has been included in the SRIF, then the $()^{**}$ terms of (12) are available. Additionally, the data equation $y^*(k) = H^*_w(k)w_k + H^*_{wx}(k+1)x_{x+1} + \nu_k$ in (11) can be generated from the measurement that is to be removed. It is possible, as shown below, to solve for a transformation to produce the upper $()^*$ terms of (11) and the ensuing data equation $y_{k+1} = H_{k+1}x_{k+1} + \nu_{k+1}$ in (12). The upper $()^*$ terms of (11) are then the downdated smoothing coefficients and replace the old smoothing coefficients. This process is repeated for $j = k+2, k+3, \ldots, N$. The terminal data equation $y_N = H_N x_N + \nu_N$ is then downdated from the terminal SRIF array [3].

Implementation

It is efficient to perform the process noise updates one component at at time. For example, in a four state filter with three process noise parameters, the smoothing coefficients from time update $j+1$ can be expressed as

$$
\begin{matrix}
R^{**}_{p1}(j) & R^{**}_{p1,x}(j+1) & R^{**}_{p1}(j+1) & R^{**}_{p1,p2}(j) & R^{**}_{p1,p3}(j) & z^{**}_{p1}(j) \\
R^{**}_{p2}(j) & R^{**}_{p2,x}(j+1) & R^{**}_{p2,p1}(j+1) & R^{**}_{p2}(j+1) & R^{**}_{p2,p3}(j) & z^{**}_{p2}(j) \\
R^{**}_{p3}(j) & R^{**}_{p3,x}(j+1) & R^{**}_{p3,p1}(j+1) & R^{**}_{p3,p2}(j+1) & R^{**}_{p3}(j+1) & z^{**}_{p3}(j)
\end{matrix} \quad j = 0, N-1 \qquad (15)
$$

where the $()^{**}$ terms are a reminder that all data has been processed in the SRIF. Note that the update order of the process noise parameters is p1, p2, p3.

The first step is to express the data equation to be removed in terms of x_{k+1} using equation (5).

$$\|(H_{x_k} \quad H_{p1_k} \quad H_{p2_k} \quad H_{p3_k}) \begin{pmatrix} x_k \\ p1_k \\ p2_k \\ p3_k \end{pmatrix} - y_k\|^2 =$$

$$\|(H_{x_k} \quad H_{p1_k} - H_{x_k}V_{p1}(k) \quad H_{p2_k} - H_{x_k}V_{p2}(k) \quad H_{p3_k} - H_{x_k}V_{p3}(k)) \begin{pmatrix} x_{k+1} \\ p1_k \\ p2_k \\ p3_k \end{pmatrix} - y_k\|^2 =$$

$$\|(H_x^*(k+1) \quad H_{p1}^*(k) \quad H_{p2}^*(k) \quad H_{p3}^*(k)) \begin{pmatrix} x_{k+1} \\ p1_k \\ p2_k \\ p3_k \end{pmatrix} - y^*(k)\|^2 \tag{16}$$

The smoothing coefficients associated with $p1$ are then downdated with an orthogonal transformation.

$$T\begin{pmatrix} R_{p1}^{**}(k) & R_{p1,x}^{**}(k+1) & R_{p1}^{**}(k+1) & R_{p1,p2}^{**}(k) & R_{p1,p3}^{**}(k) & z_{p1}^{**}(k) \\ 0 & \bar{H}_{x_{k+1}} & \bar{H}_{p1_{k+1}} & \bar{H}_{p2_k} & \bar{H}_{p3_k} & \bar{y}_k \end{pmatrix}$$
$$= \begin{pmatrix} R_{p1}^{*}(k) & R_{p1,x}^{*}(k+1) & R_{p1}^{*}(k+1) & R_{p1,p2}^{*}(k) & R_{p1,p3}^{*}(k) & z_{p1}^{*}(k) \\ H_{p1}^*(k) & H_x^*(k+1) & 0 & H_{p2}^*(k) & H_{p3}^*(k) & y^*(k) \end{pmatrix} \tag{17}$$

For the sake of notation, the following variable (re)assignments are necessary:

$$\begin{aligned} H_x^*(k+1) &:= \bar{H}_{x_{k+1}} \\ H_{p1}^*(k+1) &:= \bar{H}_{p1_{k+1}} \\ H_{p2}^*(k) &:= \bar{H}_{p2_k} \\ H_{p3}^*(k) &:= \bar{H}_{p3_k} \\ y^*(k) &:= \bar{y}_k \end{aligned} \tag{18}$$

Next, the smoothing coefficients associated with $p2$ are then downdated.

$$T\begin{pmatrix} R_{p2}^{**}(k) & R_{p2,x}^{**}(k+1) & R_{p2,p1}^{**}(k+1) & R_{p2}^{**}(k+1) & R_{p2,p3}^{**}(k) & z_{p2}^{**}(k) \\ 0 & \bar{\bar{H}}_{x_{k+1}} & \bar{\bar{H}}_{p1_{k+1}} & \bar{\bar{H}}_{p2_{k+1}} & \bar{\bar{H}}_{p3_k} & \bar{\bar{y}}_k \end{pmatrix}$$
$$= \begin{pmatrix} R_{p2}^{*}(k) & R_{p2,x}^{*}(k+1) & R_{p2,p1}^{*}(k+1) & R_{p2}^{*}(k+1) & R_{p2,p3}^{*}(k) & z_{p2}^{*}(k) \\ H_{p2}^*(k) & H_x^*(k+1) & H_{p1}^*(k+1) & 0 & H_{p3}^*(k) & y^*(k) \end{pmatrix} \tag{19}$$

Again for the sake of notation, the following variable (re)assignments are necessary:

$$\begin{aligned} H_x^*(k+1) &:= \bar{\bar{H}}_{x_{k+1}} \\ H_{p1}^*(k+1) &:= \bar{\bar{H}}_{p1_{k+1}} \\ H_{p2}^*(k+1) &:= \bar{\bar{H}}_{p2_{k+1}} \\ H_{p3}^*(k) &:= \bar{\bar{H}}_{p3_k} \\ y^*(k) &:= \bar{\bar{y}}_k \end{aligned} \tag{20}$$

Finally, the smoothing coefficients associated with $p3$ are downdated.

$$T\begin{pmatrix} R^{**}_{p3}(k) & R^{**}_{p3,x}(k+1) & R^{**}_{p3,p1}(k+1) & R^{**}_{p3,p2}(k+1) & R^{**}_{p3}(k+1) & z^{**}_{p3}(k) \\ 0 & H_{x_{k+1}} & H_{p1_{k+1}} & H_{p2_{k+1}} & H_{p3_{k+1}} & y_{k+1} \end{pmatrix}$$
$$= \begin{pmatrix} R^{*}_{p3}(k) & R^{*}_{p3,x}(k+1) & R^{*}_{p3,p1}(k+1) & R^{*}_{p3,p2}(k+1) & R^{*}_{p3}(k+1) & z^{*}_{p3}(k) \\ H^{*}_{p3}(k) & H^{*}_{x}(k+1) & H^{*}_{p1}(k+1) & H^{*}_{p2}(k+1) & 0 & y^{*}(k) \end{pmatrix} \tag{21}$$

Note that all smoothing coefficients before time update $k+1$ remain unchanged. The $()^{*}$ terms are the downdated smoothing coefficients and replace the old smoothing coefficients. The ensuing data equation $y_{k+1} = H_{k+1}x_{x+1} + \nu_{k+1}$ is then used to continue this process to downdate the smoothing coefficients from time updates $k+2$ through N. In the end, the terminal data equation $y_N = H_N x_N + \nu_N$ is then downdated from the terminal SRIF array [3].

Constructing the Orthogonal Transformation

The orthogonal transformation in equations (17), (19), and (21) operates as

$$T\begin{pmatrix} R^{**}(k) & R^{**t} \\ 0 & H^{t} \end{pmatrix} = \begin{pmatrix} R^{*}(k) & R^{*t} \\ H^{*}(k) & H^{*t} \end{pmatrix} \tag{22}$$

where the $()(k)$ terms are scalars and the $()^{t}$ terms represent row vectors. As usual, let the orthogonally packed measurement be stored in the last column of the row vectors. To solve for the transformation T, multiply equation (22) by T^{-1} and take transposes.

$$\begin{pmatrix} R^{**}(k) & 0 \\ R^{**} & H \end{pmatrix} = \begin{pmatrix} R^{*}(k) & H^{*}(k) \\ R^{*} & H^{*} \end{pmatrix} T \tag{23}$$

Let $\tilde{A}^{t} = (A, \alpha)$, where $\alpha = \sqrt{1 - A^2}$ and A is a scalar to be determined. Note that the norm of $\tilde{A}$ is 1. The transformation T is then constructed as an elementary Givens reflection.

$$T = \begin{pmatrix} -\alpha & A \\ A & \alpha \end{pmatrix} \tag{24}$$

With T constructed as such, equation (23) is then post-multiplied by $\tilde{A}$.

$$\begin{pmatrix} R^{**}(k) & 0 \\ R^{**} & H \end{pmatrix}\begin{pmatrix} A \\ \alpha \end{pmatrix} = \begin{pmatrix} R^{*}(k) & H^{*}(k) \\ R^{*} & H^{*} \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{25}$$

The scalar A and vector H can now be determined.

$$A = \frac{H^{*}k}{R^{**}(k)} \tag{26}$$

$$H = \frac{H^{*} - AR^{**}}{\alpha} \tag{27}$$

If instead of downdating, it is desirable to update the filter solution with an additional measurement, an elementary Givens reflection would be applied to the right side of equation (22).

Efficiency

The discussed method was implemented on a VAX 8530. The filter test case required 30 minutes of CPU time to process 5800 data points in a 5.5 hour data arc of a typical GPS data processing scenario for the TOPEX/POSEIDON demonstration [6]; the filter estimated nearly 300 parameters. Downdating points from the resulting filter output required on average 10 seconds of CPU time per point. It proves that the linear combination of the data equation and the smoothing coefficients is relatively fast compared with downdating the terminal data equation from the terminal SRIF array.

Numerical Integrity

Stewart [5] has shown that downdating is stable in the presence of rounding errors. However, as he has also shown, if the spread of singular values is greater than half the computational precision, the precision of the smaller singular values may be lost. Fortunately, the downdating algorithm provides a way of detecting such ill-conditioned problems; the value of α (above and a similar quantity in the terminal downdating algorithm) is a reliable indication of trouble. Experience has shown that if α is less than 10^{-8}, on a machine with a 15 digits of precision, the results of particular estimates may be inaccurate. This situation occurs when all information is removed from an estimated parameter. However, in an operational environment there are generally sufficient measurements or *a priori* information to avoid such situations. In these cases, experience has shown that the estimates are accurate to better than 10 digits.

Applications

In spacecraft orbit determination, often critical measurements, such as Very Long Baseline Interferometry (VLBI) measurements, are not available until long after the usual radio-metric data (*e.g.* doppler, range) has been obtained. Using the discussed method, these critical measurements may be efficiently added after the usual data has been processed. When using optical data for orbit determination, camera pointing is often modeled as a white noise stochastic process. The discussed method permits the analyst to efficiently remove optical frames and replace them with others as desired. For large state systems with many stochastic parameters, such as GPS applications, outliers may be removed more efficiently and just as effectively by downdating rather than reprocessing the entire data set.

Conclusion

A new method to downdate a time-varying SRIF filter is presented. The method combines the algorithm of Gill [3] to downdate a matrix factorization and the decentralized SRIF algorithm of Bierman [4] to combine the results of independent time-varying SRIF filters. This method permits efficient removal of either measurements or *a priori* information. Additionally, updating a time-varying SRIF filter with either additional measurements or *a priori* information proceeds analogously. In both cases, a data equation is propagated by alternating between time mapping the equation and forming a particular linear combination with the SRIF's smoothing coefficients. The terminal data equation that results can then be merged with or removed from the terminal SRIF array. Smoothing then proceeds as usual [2]. For large state systems with many stochastic parameters, the discussed method is expected to be a critical component in the real-time reduction and analysis of large volumes of tracking data.

References

1. G. J. Bierman, Factorization Methods for Discrete Sequential Estimation, Academic Press, 1977.
2. G. J. Bierman, "Sequential Square Root Filtering and Smoothing of Discrete Linear Systems", Automatica, Vol. 10, pp. 147–158.
3. P. E. Gill, G. H.Golub, W. Murray, and M. A. Saunders, "Methods of Modifying Matrix Factorizatrions", Mathematics of Computation, Volume 28, Number 126, April 1974, pp. 505–535.
4. G. J. Bierman, "A Decentralized Square Root Information Filter/ Smoother", Proceedings of 24th Conference on Decision and Control, pp. 1902–1905, Ft. Lauderdale, FL., December 1985.
5. G. W. Stewart, "The Effects of Rounding Error on an Algorithm for Downdating a Cholesky Factorization", University of Maryland, Computer Science Technical Report TR-582, September 1977.
6. B. G. Williams, K. C. McColl, J. R. Guinn, "Navigation Accuracies for GPS Demonstration on TOPEX/POSEIDON", AIAA/AAS Astrodynamics Conference, Portland, OR., August 1990.

Acknowledgement

The work described in this paper was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.