Math Specification Document
for
Extended Square-Root Information Filter

Jason Schmidt

April 29, 2011

License

ODTBX: Orbit Determination Toolbox

Nomenclature

x	state
δx	change in state
w	noise on state estimate
$\boldsymbol{P}$	state covariance matrix such that $\boldsymbol{P} = E(ww^T)$
$\boldsymbol{S}$	square-root of the state covariance matrix such that $\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{P}$
μ	external input
ν	process noise
$\boldsymbol{Q}$	external input covariance matrix such that $\boldsymbol{Q} = E(\nu\nu^T)$
$\boldsymbol{W}$	square-root of the external input covariance matrix such that $\boldsymbol{W}\boldsymbol{W}^T = \boldsymbol{Q}$
y	measurement
δy	change in measurement
h(t,x)	nonlinear function mapping the state to the measurement
$\boldsymbol{H}$	measurement sensitivity matrix
η	noise on measurement
$\boldsymbol{R}$	measurement covariance matrix such that $\boldsymbol{R} = E(\eta\eta^T)$
$\boldsymbol{V}$	square-root of the measurement covariance matrix such that $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{R}$
f(t,x)	nonlinear function mapping the state to state rates
$\boldsymbol{F}$	linearized state function
$\boldsymbol{\phi}$	state transition matrix
$\boldsymbol{\Gamma}$	external input transition matrix
$\boldsymbol{T}_{qr}$	transformation matrix with same effect as the qr algorithm
e	residual vector (true measurement minus predicted measurement)
dt	stepsize

Superscripts

$-$	outdated value
$+$	updated value

Subscripts

w	whitened value
k	current timestep
k-1	previous timestep
o	Initial value

Decorators

˜	modified value
˙	time derivative
ˆ	estimated state

Introduction

The extended square-root information filter (a.k.a. extended square-root inverse covariance filter) is algebraically equivalent to the extended Kalman filter but has a number of advantages. It can be initialized with with poorly known initial conditions and can be more efficient than covariance filters when large numbers of measurements are processed. In addition, because the filter operates on the square-root of the information matrix, the matrix has been essentially normalized. As a result, the algorithm is less sensitive to round-off error and the resulting numerical instability due to finite word length and has twice the precision of a non-square-root algorithm.

1 Filter Models

Before the filter can be built, linear discrete models must be developed for the state, external input, and measurement. Continuous, nonlinear models must be linearized about a state estimate and discretized.

A continuous, nonlinear state model is described by

$$\dot{x} = f(t, x) + \mu(t) \tag{1}$$

This equation can be linearized about the latest state estimate to yield

$$\delta\dot{x} = \boldsymbol{F}\delta x + \mu \tag{2}$$

where $\boldsymbol{F} = \partial f(t, x)/\partial \hat{x}_k$. The discrete form of this continuous model takes the form

$$\delta x_k = \boldsymbol{\phi}_{t_{k-1}\to t_k}\delta x_{k-1} + \boldsymbol{\Gamma}_{t_{k-1}\to t_k}\mu_{k-1} \tag{3}$$

In the discrete form, the state transition matrix $\boldsymbol{\phi} = e^{\boldsymbol{F}dt}$, and $\boldsymbol{\Gamma} = \int_0^{dt} e^{\boldsymbol{F}(dt-\tau)}d\tau$ where dt is the stepsize from t_{k-1} to t_k.

Similarly, the nonlinear measurement model is described by

$$y = h(t, x) + \eta(t) \tag{4}$$

This equation can also be linearized about the latest state estimate to yield

$$\delta y_k = \boldsymbol{H}\delta x_k + \eta_k \tag{5}$$

where $\boldsymbol{H}$ is the measurement sensitivity matrix given by $\partial h(t, x)/\partial \hat{x}_k$.

Variable Statistics

In order to build the filter, the statistical characteristics of the state, external input, and measurement must be characterized. The *a priori* state estimate is modeled by the equation

$$\hat{x} = x + w \tag{6}$$

Thus the *a priori* state estimate has an expected value $E[\hat{x}] = x$, and covariance $E[ww^T] = \boldsymbol{P}$.

Of course, when the model has been linearized about the expected value, the model becomes

$$\delta\hat{x} = \delta x + w \tag{7}$$

where $E[\delta\hat{x}] = \delta x = 0$.

The estimated external input ($\hat{\mu}$) is modeled by

$$\hat{\mu} = \mu + \nu \tag{8}$$

Thus $\hat{\mu}$ has an expected value $E[\hat{\mu}] = \mu$, and covariance $E[\nu\nu^T] = \boldsymbol{Q}$. For the unforced case, $\mu = 0$, and the external input is simply process noise.

As noted above, the measurement is modeled by

$$\delta y = \boldsymbol{H}\delta x + \eta \tag{9}$$

The measurements noise (η) has an expected value $E[\eta] = 0$ and covariance $E[\eta\eta^T] = \boldsymbol{R}$.

$\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ have upper triangular square-root forms such that

$$\begin{aligned} \boldsymbol{S}^T\boldsymbol{S} &= \boldsymbol{P} && (10)\\ \boldsymbol{W}^T\boldsymbol{W} &= \boldsymbol{Q} && (11)\\ \boldsymbol{V}^T\boldsymbol{V} &= \boldsymbol{R} && (12) \end{aligned}$$

This is easily done in Matlab with the following command

```
S=chol(P);
```

The inverse of the square-root covariances can be used to whiten the state, noise, and measurement models such that

$$\begin{aligned} \boldsymbol{S}^{-1}\delta\hat{x} = \boldsymbol{S}^{-1}\delta x + \boldsymbol{S}^{-1}w \quad &\Longrightarrow \quad \delta\hat{x}_{\mathrm{w}} = \boldsymbol{S}^{-1}\delta x + w_{\mathrm{w}} && (13)\\ \boldsymbol{W}^{-1}\hat{\mu} = \boldsymbol{W}^{-1}\mu + \boldsymbol{W}^{-1}\nu \quad &\Longrightarrow \quad \hat{\mu}_{\mathrm{w}} = \boldsymbol{W}^{-1}\mu + \nu_{\mathrm{w}} && (14)\\ \boldsymbol{V}^{-1}\delta y = \boldsymbol{V}^{-1}\boldsymbol{H}\delta x + \boldsymbol{V}^{-1}\eta \quad &\Longrightarrow \quad \delta y_{\mathrm{w}} = \boldsymbol{H}_{\mathrm{w}}\delta x + \eta_{\mathrm{w}} && (15) \end{aligned}$$

The whitened vectors have covariances equal to the identity matrix.

2 Extended Square-root Information Filter Overview

Like most sequential state estimators, the extended square-root information filter (ESRIF) has two basic parts. One part is the state and state covariance propagate, where a dynamic model is used to propagate the state to the next measurement time. The other part is the state and state covariance update, where new measurements are incorporated into the current estimate.

Depending on the setup, either one of these can be thought of as "first," depending on how the filter is initialized, but for this discussion the time propagation will be covered first. A flowchart summarizing the ESRIF algorithm is found in figure 1. For time propagation the subscripts $\cdot_k$ and $\cdot_{k-1}$ will denote the current and previous time steps respectively. For measurements updates, the superscripts $\cdot^-$ and $\cdot^+$ will denote uncorrected and corrected values respectively.

This section is intended to be a quick overview of how to implement the ESRIF. More detailed derivations are found in Section 3.

Initialization

To start, the ESRIF needs some initial or *a priori* conditions. The user must decide on an initial state estimate ($\hat{x}_o$), an associated initial state covariance ($\boldsymbol{P}_o$), and an initial external input estimate ($\hat{\mu}_o$) (which is zero for the unforced case). The initial square-root information matrix ($\boldsymbol{S}_o^{-1}$) is derived from $\boldsymbol{P}_o$ as seen in equation 10. The beauty of this approach is that if some or all of the the initial states are unknown then $\boldsymbol{S}_o^{-1}$ is singular. This singular square-root information matrix is used to generate the whitened state estimate $\delta\hat{x}_{\mathrm{w},o} = \boldsymbol{S}_o^{-1}\delta\hat{x}_o$, which is processed by the ESRIF algorithm without any difficulty. This characteristic of information filters can be very useful, but primarily for linear problems. For the nonlinear case, an initial state estimate is required for linearization.

The initial state transition matrix ($\boldsymbol{\phi}_o$), external input transition matrix ($\boldsymbol{\Gamma}_o$), whitened delta-state estimate ($\delta\hat{x}_{\mathrm{w},o}$), and whitened external input estimate ($\hat{\mu}_{\mathrm{w},o}$) are calculated as follows.

$$
\begin{aligned}
\boldsymbol{F}_o &= \partial f(t,x)/\partial \hat{x}_o && (16)\\
\boldsymbol{\phi}_o &= e^{\boldsymbol{F}_o dt} && (17)\\
\boldsymbol{\Gamma}_o &= \int_0^{dt} e^{\boldsymbol{F}_o(dt-\tau)}d\tau && (18)\\
\delta\hat{x}_{\mathrm{w},o} &= \boldsymbol{S}^{-1}\delta\hat{x}_o && (19)\\
\hat{\mu}_{\mathrm{w},o} &= \boldsymbol{W}^{-1}\hat{\mu}_o && (20)
\end{aligned}
$$

Again, note that many of these values go to zero when the state is linearized about the current estimate and the model is unforced. They are included here to maintain generality and to aid in the derivations found in Section 3.

Time Propagation

The time propagation is accomplished through a transformation such that:

$$
\boldsymbol{T}_{qr}\begin{bmatrix} \boldsymbol{W}^{-1} & \boldsymbol{0} & \hat{\mu}_{\mathrm{w,k\text{-}1}} \\ -\boldsymbol{S}_{k-1}^{-1}\boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1}\boldsymbol{\Gamma}_{t_{k-1}\to t_k} & \boldsymbol{S}_{k-1}^{-1}\boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1} & \delta\hat{x}_{\mathrm{w},k-1}\end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{W}}_{\mu}^{-1} & \widetilde{\boldsymbol{S}}_x^{-1} & \widetilde{\mu}_{\mathrm{w}} \\ \boldsymbol{0} & \boldsymbol{S}_k^{-1} & \delta\hat{x}_{\mathrm{w},k}\end{bmatrix} \tag{21}
$$

$\uparrow$ $\uparrow$

propagated SR information matrix
and whitened state

This transformation can be accomplished mathematically with either a Gram-Schmidt orthogonalization process or a Householder transformation. It is accomplished in Matlab with the qr.m function such that

```
[q,r]=qr(A);
Anew=r;
```

In practice, the transformation is used to propagate the information matrix. The state estimate is propagated through integration such that

$$
\hat{x}_k = \int_{t_{k-1}}^{t_k} f(\tau,\hat{x})d\tau + \hat{x}_{k-1} \tag{22}
$$

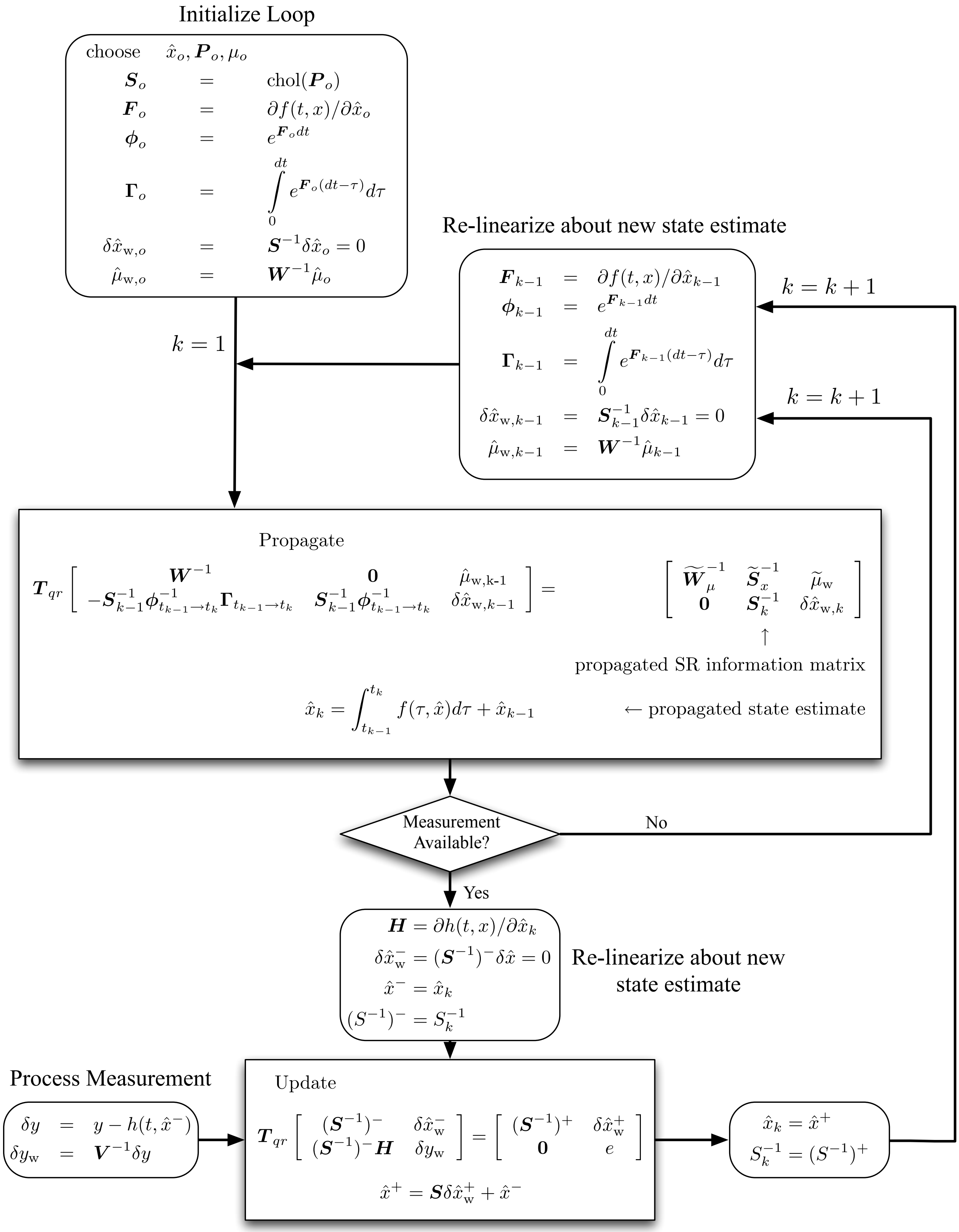

Figure 1: ESRIF Flowchart

Measurement Update

To reduce errors due to nonlinearity, the measurement equation is linearized about the propagated state estimate.

$$\boldsymbol{H} = \partial h(t,x)/\partial \hat{x}^- \tag{23}$$

Second, the whitened delta-state estimate must be defined.

$$\delta\hat{x}_{\mathrm{w}}^- = (\boldsymbol{S}^{-1})^- \delta\hat{x} \tag{24}$$

This is zero because the problem has been re-linearized about the most recent state estimate.

Finally, the actual measurement (y) must be turned into a whitened delta-measurement (δy_{w}). This is done as follows:

$$\delta y = y - h(t, \hat{x}^-) \tag{25}$$

$$\delta y_{\mathrm{w}} = \boldsymbol{V}^{-1}\delta y \tag{26}$$

Now the outdated square-root information matrix $(\boldsymbol{S}^{-1})^-$ and the outdated whitened state estimate $\hat{x}_{\mathrm{w}}^-$ may be updated. The update performs a transformation such that

$$\boldsymbol{T}_{qr}\begin{bmatrix} (\boldsymbol{S}^{-1})^- & \delta\hat{x}_{\mathrm{w}}^- \\ (\boldsymbol{S}^{-1})^-\boldsymbol{H} & \delta y_{\mathrm{w}} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{S}^{-1})^+ & \delta\hat{x}_{\mathrm{w}}^+ \\ \boldsymbol{0} & e \end{bmatrix} \tag{27}$$

where e is the measurement residual (true measurement minus the estimated measurement). This is the same type of orthogonal transformation required for the time propagation step and is accomplished in Matlab with the qr.m function.

Essentially, this transformation adds the additional measurement information to the information matrix and corrects the whitened state estimate with the whitened measurement.

The state estimate is updated thus:

$$\hat{x}^+ = \boldsymbol{S}\delta\hat{x}_{\mathrm{w}}^+ + \hat{x}^- \tag{28}$$

3 Derivations

The following sections contain more detailed derivations of the measurement update and time propagation steps covered above. This derivation is loosely based on Bierman [1].

Measurement Update Derivation

Kalman filtering is an iterative application of Gauss's principle of least squares. The concept is to solve for a delta-state estimate that minimizes the sum square of the residuals. For the measurement update step, we have two relationships to satisfy, *a priori* information and a new measurement. Based on their whitened models (Equations 13 and 15), whitened residuals can be formulated in terms of the *a priori* whitened delta-state ($\delta\hat{x}_{\mathrm{w}}^-$ and usually zero), the *a priori* square-root information matrix $(\boldsymbol{S}^{-1})^-$, the whitened delta-measurement (δy_{w}), and the TBD delta-state estimate ($\delta\hat{x}$).

$$w_{\mathrm{w}} = \delta\hat{x}_{\mathrm{w}}^- - (\boldsymbol{S}^{-1})^-\delta\hat{x} \tag{29}$$

$$\eta_{\mathrm{w}} = \delta y_{\mathrm{w}} - (\boldsymbol{S}^{-1})^-\boldsymbol{H}\delta\hat{x} \tag{30}$$

The cost function for the update step is

$$J = \|w_{\mathrm{w}}\|^2 + \|\eta_{\mathrm{w}}\|^2 \tag{31}$$

$$J = \left\|\delta\hat{x}_{\mathrm{w}}^- - (\boldsymbol{S}^{-1})^-\delta\hat{x}\right\|^2 + \left\|\delta y_{\mathrm{w}} - (\boldsymbol{S}^{-1})^-\boldsymbol{H}\delta\hat{x}\right\|^2 \tag{32}$$

Rearrange equation 32 into the form

$$J = \left\|\begin{bmatrix} (\boldsymbol{S}^{-1})^- \\ (\boldsymbol{S}^{-1})^-\boldsymbol{H} \end{bmatrix}\delta\hat{x} - \begin{bmatrix} \delta\hat{x}_{\mathrm{w}}^- \\ \delta y_{\mathrm{w}} \end{bmatrix}\right\|^2 \tag{33}$$

The Gram-Schmidt process is usually used to produce an orthogonal set of nonzero vectors that span the original vector space. However, it also generates an upper triangular matrix whose columns span the original vector space. The bracketed values in equation 33 can be concatenated into a matrix and put through the Gram-Schmidt process.

$$T_{qr}\begin{bmatrix} (\boldsymbol{S}^{-1})^- & \delta\hat{x}_{\mathrm{w}}^- \\ \boldsymbol{H}_w & \delta y_{\mathrm{w}} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{S}^{-1})^+ & \delta\hat{x}_{\mathrm{w}}^+ \\ \boldsymbol{0} & e \end{bmatrix}$$

Plugging these terms back into equation 33 gives

$$J = \left\|\delta\hat{x}_{\mathrm{w}}^+ - (\boldsymbol{S}^{-1})^+\delta\hat{x}\right\|^2 + \|e\|^2 \tag{34}$$

$\delta\hat{x}_{\mathrm{w}}^+$ is the updated whitened state estimate, and $(\boldsymbol{S}^{-1})^+$ is the updated square-root information matrix. It is now a simple matter to solve for a delta-state estimate $\delta\hat{x}$ that zeros the first term, and minimizes J.

State Propagation Derivation

For state propagation, the cost function J must include the state propagation equation. Solving equation 3 for the previous step and replacing the actual δx and μ with the estimated $\delta\hat{x}$ and $\hat{\mu}$yields

$$\delta\hat{x}_{k-1} = \boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1}(\delta\hat{x}_k - \boldsymbol{\Gamma}_{t_{k-1}\to t_k}\hat{\mu}_{k-1}) \tag{35}$$

In addition, the introduction of external input introduces another residual.

$$\nu_{\mathrm{w}} = \hat{\mu}_{\mathrm{w}} - \boldsymbol{W}^{-1}\hat{\mu} \tag{36}$$

Substituding equation 35 for $\delta\hat{x}$ the update cost function (equation 34) and adding the process noise residual yields the new cost function.

$$J = \|e\|^2 + \left\|\boldsymbol{S}_{k-1}^{-1}\{\boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1}(\delta\hat{x}_k - \boldsymbol{\Gamma}_{t_{k-1}\to t_k}\hat{\mu}_{k-1})\} - \delta\hat{x}_{\mathrm{w,k\text{-}1}}\right\|^2 + \left\|\boldsymbol{W}^{-1}\hat{\mu}_{k-1} - \hat{\mu}_{\mathrm{w,k\text{-}1}}\right\|^2 \tag{37}$$

In this form, the solution for $\delta\hat{x}_k$ is dependent on the external input $\hat{\mu}_{k-1}$. Rearranging this cost function yields

$$J = \|e\|^2 + \left\|\begin{bmatrix} \boldsymbol{W}^{-1} & \boldsymbol{0} \\ -\boldsymbol{S}_{k-1}^{-1}\boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1}\boldsymbol{\Gamma}_{t_{k-1}\to t_k} & \boldsymbol{S}_{k-1}^{-1}\boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1} \end{bmatrix}\begin{bmatrix} \hat{\mu}_{k-1} \\ \delta\hat{x}_k \end{bmatrix} - \begin{bmatrix} \hat{\mu}_{\mathrm{w},k-1} \\ \delta\hat{x}_{\mathrm{w},k-1} \end{bmatrix}\right\|^2 \tag{38}$$

Just like the update step, the left and right bracketed terms are concatenated and a Gram-Schmidt process progressively diagonalizes the matrix until the $-\boldsymbol{S}_{k-1}^{-1}\boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1}\boldsymbol{\Gamma}_{t_{k-1}\to t_k}$ term has been eliminated. The resulting matrix is:

$$T_{qr}\begin{bmatrix} \boldsymbol{W}^{-1} & \boldsymbol{0} & \hat{\mu}_{\mathrm{w},k-1} \\ -\boldsymbol{S}_{k-1}^{-1}\boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1}\boldsymbol{\Gamma}_{t_{k-1}\to t_k} & \boldsymbol{S}_{k-1}^{-1}\boldsymbol{\phi}_{t_{k-1}\to t_k}^{-1} & \delta\hat{x}_{\mathrm{w},k-1} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{W}}_{\mu}^{-1} & \widetilde{\boldsymbol{S}}_x^{-1} & \widetilde{\mu}_{\mathrm{w}} \\ \boldsymbol{0} & \boldsymbol{S}_k^{-1} & \delta\hat{x}_{\mathrm{w},k} \end{bmatrix}$$

Plugging these values back into equation 38 yields the following

$$J = \|e\|^2 + \left\| \widetilde{\boldsymbol{W}}_\mu^{-1} \hat{\mu}_{k-1} + \widetilde{\boldsymbol{S}}_x^{-1} \delta\hat{x}_k - \widetilde{\mu}_{\mathrm{w}} \right\|^2 + \left\| \boldsymbol{S}_k^{-1} \delta\hat{x}_k - \hat{x}_{\mathrm{w},k} \right\|^2 \tag{39}$$

$\boldsymbol{S}_k^{-1}$ is the propagated square-root information matrix and $\delta\hat{x}_{\mathrm{w},k}$ is the propagated whitened state. It is a simple task to solve for $\delta\hat{x}_k$ such that the right hand residual goes to zero, and then pick a $\hat{\mu}_{k-1}$ such that the middle residual goes to zero. Thus minimizing the cost function J. However, during practice, the state estimate is integrated and $\boldsymbol{S}_k^{-1}$ is the only value obtained from the Gram-Schmidt process.

References

[1] Gerald Bierman. *Factorization Methods for Discrete Sequential Estimation.* Dover Publications, NY, 1977.