AAS 94-169

A CONTINUOUS SQUARE ROOT INFORMATION FILTER-SMOOTHER WITH DISCRETE DATA UPDATE

James K. Miller[†]

A differential equation for the square root information matrix is derived and adapted to the problems of filtering and smoothing. The resulting continuous square root information filter (SRIF) performs the mapping of state and process noise by numerical integration of the SRIF matrix and admits data via a discrete least square update. For comparison, the matrix differential equations for the covariance filter or continuous Kalman-Bucy filter, the information filter, and the square root covariance filter are also derived.

Computational efficiency, accuracy, computer memory requirements and simplicity of design are compared with other filter designs. Computational efficiency seems to favor discrete filters since these have been developed to a high degree of efficiency. The continuous SRIF is expected to out perform discrete filters with regard to accuracy and memory requirements because there is no need to compute the state transition matrix and error control may be placed directly on the elements of the SRIF matrix. Elimination of the state transition matrix may also eliminate numerical problems that have been experienced in computing this matrix. Simplicity of design favors the continuous SRIF particularly when the same numerical integration algorithm used to propagate the state is used to integrate the SRIF matrix. Also, the introduction, of process noise to the filter as a differential equation enables the investigation of a wide variety of noise models without the explicit solution of the differential equation.

INTRODUCTION

A data filter processes data in order to obtain an estimate of parameters that are related to the data by a mathematical model. Data filters (at least in many) forms and use the covariance of the state parameters, or some equivalent representation, along with the measurements and a simulation of the measurements including partial derivatives, to obtain the desired estimate. Data filters may be separated into two categories depending on how the state covariance is evolved as a function of time. Continuous data filters evolve the state covariance by integration of a matrix differential equation or Ricatti equation and discrete data filters evolve the state covariance by mapping over a finite time interval. Discrete filters are thus obtained by solving the continuous equations over some finite time interval. The covariance matrix of the state may be represented by its inverse or information matrix or square root factorization of either of these matrices.

In the formulation of the filter described in this paper, system dynamics and process noise are described by differential equations. For the simple case of exponentially time correlated process noise, both the continuous and discrete formulations are described. Data is processed sequentially similar to the discrete Kalman-Bucy filter. The matrix differential equation for the mapping term of the SRIF matrix is derived from the system dynamics. The process noise term and data update term are obtained by transformation of the corresponding terms in either the covariance or information filters using simple matrix identities. Dual relationships are shown for the continuous filters and these are exploited in the development of the filter.

[†] Member of the Technical Staff, Navigation Systems Section, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, Associate fellow AIAA.

In order to complete the continuous SRIF description, the problems of smoothing, discrete process noise update, iteration for solution, and numerical integration of the SRIF matrix are described.

SYSTEM DYNAMICS

‘l’he system dynamics may be described as a linear perturbation of a reference function of the state variables. Given the nominal values of the state variables described by the function $\bar{x}(t)$ and a perturbation of the state (δx) at the initial epoch (t_0), the perturbed state variables arc described by

$$x(t) = \bar{x}(t) + \Phi(t, t_0)\, \delta x(t_0) \tag{1}$$

where the state transition matrix (Φ) is given by

$$\Phi = \frac{\partial x(t)}{\partial x(t_0)}$$

‘l’he state transition matrix may be obtained as a solution of the following differential equation or by numerical integration.

$$\frac{\partial \dot{x}(t\ )}{\partial x(t_0)} = \frac{\partial \dot{x}(t)}{\partial x(t)} \frac{\partial x(t)}{\partial x(t_0)}$$

$$\dot{\Phi}(t, t_0) = F\,\Phi(t, t_0) \tag{2}$$

where

$$F = \frac{\partial \dot{x}(t)}{\partial x(t)}$$

The above differential equation describing the evolution of the state variation may be generalized to include other parameters and process noise.

$$\dot{X} = F\,X \ + \ G\Omega \tag{3}$$

where G is the mapping of Ω, the process noise. Here, the b's have been dropped and the variation δx is represented by X, The state vector variation X may be generalized to include constant parameters (y) and stochastic parameters (p) as well as the dynamic state variables (x). The process noise (Ω) contains white noise (ω) on the stochastic parameters. Thus wc have

$$X = \begin{bmatrix} p \\ x \\ y \end{bmatrix} \qquad \Omega = \begin{bmatrix} \omega \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

The stochastic parameters (p) provide a means of introducing process noise into the state variables. These arc defined by scalar differential equations of the form

$$\dot{p}_i = -\frac{1}{\tau_i}\, p_i + \omega_i \tag{5}$$

where τ_i is the correlation time and ω_i is the white noise associated with the i'th stochastic parameter, “1'bus, white noise is introduced directly to the parameter p and indirectly to the state via the mapping matrix F.

An estimate of the state is obtained from a mathematical model of the system dynamics that include measurements processed by a data filter. The “best” estimate of the variation of the state $(\hat{X})$ is described by the following equations,

$$\text{x} = F\hat{X} + G\hat{\Omega} +\!- K\hat{Z} \tag{6}$$

$$\hat{Z} = Z - H\hat{X} \tag{7}$$

$$H = \frac{\partial Z}{\partial X} \tag{8}$$

where K is the Kalman gain, $\hat{\Omega}$ represents an estimate of the process noise, Z arc the actual measurements and H is the matrix of data partials. The Kalman gain is computed as a function of the measurement error, the data partials and the state error covariance (P). '1'bus, in order to obtain a complete set of equations that would enable the computation of the estimated state wc need an equation for the Kalman gain and an equation for evolving P as a function of time.

DERIVATION OF CONTINUOUS FILTER. EQUATIONS

The covariance of the state estimate is defined by the expected value represented by

$$P = E\{XX^{T}\} \tag{9}$$

As an alternative, wc may compute the information matrix (A), the square root of the covariance (S), or the square root of the information matrix (R). The equations that define these matrices arc given by

$$P = \Lambda^{-1} \tag{10}$$

$$P = SS^{T} \tag{11}$$

$$P^{-1} = R^{T}R \tag{12}$$

'l'bus, wc arc interested in obtaining differential equations of the form

$$\dot{P} = \dot{P}_m + \dot{P}_q + \dot{P}_d \tag{13}$$

$$\dot{\Lambda} = \dot{\Lambda}_m + \dot{\Lambda}_q + \dot{\Lambda}_d \tag{14}$$

$$\dot{S} = \dot{S}_m + \dot{S}_q + \dot{S}_d \tag{15}$$

$$\dot{R} = \dot{R}_m + \dot{R}_q + \dot{R}_d \tag{16}$$

where the subscript m refers to the mapping terms, the subscript g refers to process noise terms, and the subscript d refers to the data update terms.

Mapping Term

The evolution of the covariance as a function of time [I] may be obtained by mapping the state covariance obtained at some epoch (t_0) to some time in the future (t) with the state transition matrix.

$$P(t) = \Phi(t, t_0)\, P(t_0)\, \Phi(t, t_0)^{T} \tag{17}$$

Taking the derivative with respect to time wc obtain

$$\dot{P}(t) = \dot{\Phi}(t, t_0)\ P(t_0)\ \Phi(t, t_0)^{T} + \Phi(t, t_0)\ P(t_0)\ \dot{\Phi}(t, t_0)^{T} \tag{18}$$

Since the state transition matrix is obtained by integrating

$$\dot{\Phi}(t, t_0) = F(t)\, \Phi(t, t_0) \tag{19}$$

wc obtain after substitution

$$\dot{P}_m = FP + 1'1^{\prime\prime} \tag{20}$$

Process Noise Term

In the covariance matrix differential equation, process noise enters as a simple addition to the covariance. Thus wc have

$$P(t + \Delta t) = P(t) + G\Delta Q G^{T} \tag{21}$$

where AQ is the covariance of the process noise admitted over the time interval Δt and

$$\Delta Q = Q\,\Delta t$$

where Q is the rate of accumulation of process noise. Thus, in the continuum we have

$$\dot{P}_q = \lim_{\Delta t \to 0} \left\{ \frac{P(t+\Delta t) - P(t)}{\Delta t} \right\} = G\,Q\,G^{T} \tag{22}$$

Data Update Term

The discrete covariance update may be obtained assuming an additional measurement H_{n+1} is added to a previously determined estimate based on measurements H_n with covariance P_n. The derivation is given in many references [2,3] that are available,

$$P_{n+1} = \left[H_n^{T} \Delta W_n H_n + H_{n+1}^{T} \Delta W_{n+1} H_{n+1}\right]^{-1} \tag{23}$$

in the notation used here, H_n is a matrix with n rows corresponding to the measurements and m columns corresponding to the state parameters. H_{n+1} is a row matrix of dimension m. We also have for the covariance update,

$$P_{n+1}^{-1} = P_n^{-1} + H_{n+1}^{T} \Delta W_{n+1} H_{n+1} \tag{24}$$

and since

$$\Lambda = P^{-1}$$

$$\Lambda_{n+1} = \Lambda_n + H_{n+1}^{T} \Delta W_{n+1} H_{n+1} \tag{25}$$

over the time interval At between measurements, information accumulates at a rate W and

$$\Delta W_{n+1} = W\ At \tag{26}$$

$$\Lambda_{n+1} - \Lambda_n = H_{n+1}^{T}\ WAt\ H_{n+1} \tag{27}$$

Dividing by At and taking the limit as Δt approaches zero,

$$\dot{\Lambda}_m = H^{T}\ w\ 11 \tag{28}$$

we obtain a differential equation for the evolution of the information matrix due to addition of data.

Least Square Data Update

In order to complete the filter equations, we need an algorithm for processing the measurements to obtain a best estimate of the state. The discrete form of the Kalman update algorithm is given by [2]

$$AA' = \tilde{P} H^{T} (\Delta W^{-1} + H \tilde{P} H^{T})^{-1} \tag{29}$$

An equivalent expression is obtained by use of the matrix inversion lemma.

$$\Delta K = \left[\tilde{P}^{-1} + H^{T} \Delta W\, H\right]^{-1} H^{T} \Delta W \tag{30}$$

If we admit the data at a rate W over a time interval At we have

$$\Delta K = \left[\tilde{P}^{-1} + H^{T} W \Delta t\, H\right]^{-1} H^{T} W \Delta t \tag{31}$$

where $\tilde{P}$ is the covariance at the beginning of the interval prior to processing the data. Dividing through by Δt and taking the limit as Δt approaches zero we obtain

$$K = \lim_{\Delta t \to 0} \left\{ \frac{\Delta K}{\Delta t} \right\} = P\, H^{T} W \tag{32}$$

a differential equation for the Kalman update.

FILTER> DIFFERENTIAL EQUATIONS

Collecting the terms derived above, we have the following matrix differential equation or Ricatti equation for the covariance filter,

$$\dot{P} = FP + PF^{\mathrm{T}} + GQG^{\mathrm{T}} + \dot{P}_d \tag{33}$$

$$K = P H^{\mathrm{T}} w \tag{34}$$

and for the information filter,

$$\dot{\Lambda} = \dot{\Lambda}_m + \dot{\Lambda}_q + H^{\mathrm{T}} W H \tag{35}$$

$$K = \Lambda^{-1} H^{\mathrm{T}} W \tag{36}$$

The data update term $(\dot{P}_d)$ is missing from the covariance equation and the mapping $(\dot{\Lambda}_m)$ and process noise $(\dot{\Lambda}_q)$ terms are missing from the information filter equation and these may be obtained by transformation using matrix identities. For the covariance and information equations, we need the following matrix identities.

$$P\Lambda = 1$$

$$\dot{P}\Lambda + P\dot{\Lambda} = 0$$

$$\dot{P} = -P\dot{\Lambda}\Lambda^{-1} = -P\,\dot{\Lambda}\,P \tag{37}$$

$$\dot{\Lambda} = -P^{-1}\,\dot{P}\,\Lambda = -\Lambda\,\dot{P}\,\Lambda \tag{38}$$

Applying these identities to the above matrix differential equations, we have

$$\dot{P} = FP + PF^{\mathrm{T}} + GQG^{\mathrm{T}} - PH^{\mathrm{T}} W H P \tag{39}$$

$$K = P H^{\mathrm{T}} w \tag{40}$$

The covariance filter in this form is called the continuous form of the Kalman-Bucy filter. For the information filter, we have

$$\dot{\Lambda} = -\Lambda F - F^{\mathrm{T}}\Lambda - \Lambda\, GQG^{\mathrm{T}}\, \Lambda + H^{\mathrm{T}} W H \tag{41}$$

$$K = \Lambda^{-1} H^{\mathrm{T}} W \tag{42}$$

A similar set of matrix identities may be developed for the square root covariance filter (SRCF) and the square root information filter (SRIF) that may be used to transform the covariance time derivative. These identities are derived by Scheeres in Reference 4 and the derivation is repeated here.

$$P = SS^{\mathrm{T}}$$

$$\dot{P} = \dot{S}S^{\mathrm{T}} + S\dot{S}^{\mathrm{T}}$$

$$\left[S\dot{S}^{\mathrm{T}} - \frac{1}{2}\dot{P}\right] + \left[\dot{S}S^{\mathrm{T}} - \frac{1}{2}\dot{P}\right] = 0$$

Because of symmetry associated with the above terms in the brackets, both terms in the brackets must be zero and

$$\dot{S} = \frac{1}{2}\dot{P}\,S^{-\mathrm{T}} \tag{43}$$

A similar derivation for the SRIF matrix gives the identity

$$\dot{R} = -\frac{1}{2} R\dot{P} R^{\mathrm{T}} R \tag{44}$$

Applying these identities to the covariance and information filter equations gives the following matrix differential equations for the SRCF and SRIF matrices.

$$\dot{S} = \frac{1}{2}\left[FS + SS^{\mathrm{T}}F^{\mathrm{T}}S^{-\mathrm{T}}\right] + \frac{1}{2}G\,Q\,G^{\mathrm{T}}S^{-\mathrm{T}} - \frac{1}{2}SS^{\mathrm{T}}H^{\mathrm{T}}WHS \tag{45}$$

$$\dot{R} = -\frac{1}{2}\left[RF + R^{-1}F^{\mathrm{T}}R^{\mathrm{T}}R\right] - \frac{1}{2}RGQG^{\mathrm{T}}R^{\mathrm{T}}R + \frac{1}{2}R^{-\mathrm{T}}H^{\mathrm{T}}\,W\,H \tag{46}$$

The mapping terms for both the SRCF and SRIF contain matrix inverses. These may be eliminated by introducing a different factorization of the square roots. Consider the mapping of the square root covariance from an initial epoch t_0 to the epoch t.

$$P(t) = \Phi(t,t_0)\,S(t_0)S(t_0)^{\mathrm{T}}\;\Phi(t,t_0)^{\mathrm{T}} \tag{47}$$

The mapped square root is simply

$$S(t) = \Phi(t,t_0)\,S(t_0) \tag{48}$$

Taking the derivative with respect to time,

$$\dot{S}(t) = \dot{\Phi}(t,t_0)\,S(t_0)$$

$\dot{S}(t) =$ i)(t,io)o(t,to)-"is(t)

$$\dot{S}(t) = F(t)S(t) \tag{49}$$

For the SRIF matrix we have

$$S(t)R(t) = 1$$

$$\dot{S}(t)R(t) + S(t)\dot{R}(t) = 0$$

$$\dot{R}(t) = -R(t)\dot{S}(t)S(t)^{-1}$$

$$\dot{R}(t) = -R(t)\,F(t) \tag{50}$$

Making the above substitutions for the mapping terms, the matrix differential equations and Kalman gain for the covariance, information, square root covariance and square root information filters are summarized below. Equations for the discrete formulation of these filters are given in Reference [5].

Covariance (Kalman-Bucy) Filter

$$\dot{P} = FP + PF^{\mathrm{T}} + G\,Q\,G^{\mathrm{T}} - PH^{\mathrm{T}}\,W\,H\,P \tag{51}$$

$$K = P\,H^{\mathrm{T}}\,W \tag{52}$$

Information Filter

$$h = -\Lambda F - F^{\mathrm{T}}\Lambda - A\;GQG^{\mathrm{T}}\;A + H^{\mathrm{T}}\;W\;H \tag{53}$$

$$K = \Lambda^{-1}\,H^{\mathrm{T}}\,W \tag{54}$$

Square Root Covariance Filter (SRCF)

$$\dot{S} = FS + \frac{1}{2}G\,Q\,G^{\mathrm{T}}S^{-\mathrm{T}} - \frac{1}{2}SS^{\mathrm{T}}H^{\mathrm{T}}WHS \tag{55}$$

$$K = SS^{\mathrm{T}}H^{\mathrm{T}}\,W \tag{56}$$

Square Root information Filter (SRIF)

$$\dot{R} = -RF - \frac{1}{2}RGQG^{\mathrm{T}}R^{\mathrm{T}}R + \frac{1}{2}R^{-\mathrm{T}}H^{\mathrm{T}}\,W\,H \tag{57}$$

$$K = R^{-1}R^{-\mathrm{T}}\,H^{\mathrm{T}}\,W \tag{58}$$

The data update and process noise terms of the above filter equations exhibit a symmetry or duality when the information filters are compared with the covariance filters. For example the data update term of the information filter may be obtained by replacing Q with W and G with H^T in the process noise term of the covariance filter. Also, the process noise update term of the information filter may be obtained by making similar replacements in the data update term of the covariance filter. These same dual relationships exist for the filters in their square root form. The existence of duality enables algorithms designed for data updating to be used for process noise updating and vice versa. For example, the Potter square root covariance data update algorithm may be used to update process noise in the SRIF.

CONTINUOUS SRIF WITH DISCRETE DATA UPDATE

The selection of a filter algorithm depends on many competing criteria related to accuracy, computational efficiency, memory utilization and simplicity of design. Consideration of accuracy seems to favor factorized or square root filters and computational efficiency seems to favor discrete filters. With the proliferation of personal computers, computational efficiency has become less important since computer processing time is relatively cheap. Simplicity of design and memory utilization favor a continuous approach to filtering. The system dynamics and data partial derivatives enter directly into the filter and the need to compute a state transition matrix is completely eliminated. However, data is generally in the form of discrete data points and may not be easily transformed to the continuous form. This suggests a hybrid approach which allows system dynamics and process noise to be treated continuously and data to be treated as a discrete update.

The continuous SRIF, with discrete data update is selected for development of a filter algorithm. information filters have the advantage that apriori on the constant parameters dots not have to be placed on the filter until after all the data is processed. During filtering, the information arrays may be sparse resulting in less computation. The SRIF algorithm described below also includes provision for smoothing and discrete process noise update.

Discrete Data Update

The SRIF discrete data update algorithm follows directly from the least square data update. The least square solution is given by [3]

$$x = \left[H_n^T \Delta W_n H_n\right]^{-1} H_n^T \Delta W_n \hat{Z}_n \tag{59}$$

The measurements can be normalized by factoring ΔW_n into

$$\Delta W_n = \sqrt{\Delta W_n}^T \sqrt{\Delta W_n}$$

and

$$\hat{X} = \left[H_n^T \sqrt{\Delta W_n}^T \sqrt{\Delta W_n} H_n\right]^{-1} H_n^T \sqrt{\Delta W_n}^T \sqrt{\Delta W_n}\, \hat{Z}_n \tag{60}$$

By inspection we can see that

$$R_n = \sqrt{\Delta W_n}\, H_n$$

so after substitution we have

$$\hat{X} = (R_n^T R_n)^{-1} R_n^T \sqrt{\Delta W_n}\, Z_n$$

For the first m measurements, the number of estimated parameters (m) is equal to the number of measurements (n) and R_n is square.

$$\hat{X} = R_n^{-1} \sqrt{\Delta W_n} \hat{Z}_n \tag{61}$$

Multiplying through by R_n gives the data equation

$$R_n \hat{X} = \sqrt{\Delta W_n} \hat{Z}_n = \hat{\eta}_n \tag{62}$$

where $\hat{\eta}_n$ is the normalized measurement. A new measurement can be appended to the data equation resulting in

$$\begin{bmatrix} R_n \\ \\ \sqrt{\Delta W_{n+1}}\, H_{n+1} \end{bmatrix} \hat{X} = \begin{bmatrix} \hat{\eta}_n \\ \\ \hat{\eta}_{n+1} \end{bmatrix} \tag{63}$$

Adding additional measurements results in the row dimension of R exceeding the column dimension. The information matrix would then be given by

$$\Lambda_m = R_{nm}^{\mathrm{T}} R_{nm} \tag{64}$$

where the row dimension n exceeds the column dimension m. Since R_{nm} is not unique, it can be replaced by an upper triangular R_m of dimension m by m,

$$\Lambda_m = R_m^{\mathrm{T}} R_m \tag{65}$$

The Householder algorithm enables one to obtain the matrix R_m without explicitly computing Λ_m. If T is an orthogonal matrix which has the proper-ty

$$T^{\mathrm{T}} T = I \tag{66}$$

then we h ave

$$\Lambda_m = \text{li:\_7'1'7'I\&v}, \tag{67}$$

The Householder algorithm finds a T that gives R_m when multiplied times R_{nm}. The right side of the data equation $(\hat{\eta})$ is is also multiplied by T to obtain a new data equation in upper triangular form. The Householder algorithm thus serves the same purpose in updating the SRIF matrix and right side as the Kalman update algorithm serves to update the covariance and state estimate. An updated state estimate can be obtained from the data equation by simply inverting the SRIF matrix and multiplying times the right side.

Continuous Process Noise Update

The continuous process noise update enables one to introduce process noise directly as a differential equation to the filter, This form is convenient for describing process noise and enables the investigation of a wide variety of process noise models without explicitly solving the differential equation, The continuous process noise update term in the information filter has the same form as the data update term in the covariance filter. The Potter square root covariance data update algorithm [6] provides a means of performing a scalar data update to the square root covariance filter. Because of duality, the discrete Potter data update algorithm can be adapted to the SRIF for a discrete scalar process noise update. Taking the limit as Δt approaches zero enables one to convert the discrete process noise update to a continuous process noise update.

Starting with the process noise update term in the information filter we have

$$\dot{\Lambda}_q = -\Lambda\, GQG^{\mathrm{T}}\, \Lambda = \lim_{\Delta t \to 0} \frac{\Lambda_{n+1} - \Lambda_n}{\Delta t} \tag{68}$$

and in the discrete form,

$$\Lambda = \tilde{\Lambda} - \tilde{\Lambda} G \Delta Q G^{\mathrm{T}} \tilde{\Lambda} \tag{69}$$

where the notation for Λ_n, the information matrix before the update or apriori, is replaced by $\tilde{\Lambda}$ and Λ_{n+1} is replaced by Λ. Since

$$\Lambda = R^{\mathrm{T}} R$$

we have

$$R^{\mathrm{T}} R = \tilde{R}^{\mathrm{T}} \left[I - v\, \Delta Q\, v^{\mathrm{T}} \right] \tilde{R} \tag{70}$$

where

$$v = \tilde{R} G$$

If ΔQ and G arc assumed to bc diagonal (i.e. uncorrelated process noise parameters) then cacb diagonal element of ΔQ is given by a scalar Δq_i. Dropping the i subscript, wc have for the i'th row of R and diagonal element of ΔQ,

$$I - \Delta q v v^T = (I - \Delta\alpha v v^T)^2$$

$$I - \Delta q v v^T = I - 2\Delta\alpha v v^T + \Delta\alpha^2 v v^T v v^T \tag{71}$$

Since $v^T v$ is a scalar, the solution of the above quadratic equation is given by

$$\Delta\alpha = \frac{1 - \sqrt{1 - v^T v \Delta q}}{v^T v} \tag{72}$$

and

$$R^T R = \tilde{R}^T (I - \Delta\alpha\, v v^T)^T (I - \Delta\alpha\, v v^T) \tilde{R}$$

$$R = (I - \Delta\alpha\, v v^T) \tilde{R}$$

$$R = \tilde{R} - \Delta\alpha\, \tilde{R} G G^T \tilde{R}^T \tilde{R} \tag{73}$$

in the continuum wc have

$$\dot{R} = -\Delta\dot{\alpha} \tilde{R} G G^T \tilde{R}^T \tilde{R} \tag{74}$$

and

$$\Delta\dot{\alpha} = \frac{1}{2}(1 - v^T v \Delta q)^{-\frac{1}{2}} \Delta\dot{q} \tag{75}$$

In the limit as Aq and Δt go to zero wc have

$$\Delta\dot{\alpha} = \frac{1}{2}\Delta\dot{q} = \frac{1}{2} q \tag{76}$$

and

$$\dot{R} = -; \; q \; RGG7'1?'1(\tag{77}$$

If wc have more than onc stochastic parameter, the q's can bc assembled into a diagonal matrix Q and wc have

$$\dot{R} = -\frac{1}{2}\sum q_i R G G^T R^T R = -\frac{1}{2} R G Q G^T R^T R \tag{78}$$

'l'his is the same equation as derived above for the continuous SRIF process noise update only wc have assumed diagonal Q and G.

Discrete Process Noise Update

A problem with integrating the process noise differential equation is the computer time required to integrate the SRIF matrix. An alternative to introducing continuous process noise to the filter is the discrete process noise update [7]. The frequency of the discrete update is selected to approximate the accuracy of the continuous update. Over the time interval between discrete process noise updates, the stochastic parameters arc assumed to bc constant and enter into the mapping the same as other constant parameters. The discrete process noise update consists of imposing an analytic solution for the process noise variance over a fixed time interval as an impulsive delta function. This method is equivalent to integrating the process noise differential equation by trapezoidal integration.

For the simple case of exponentiall y time correlated process noise, an analytic solution [8,9] for cacb process noise parameter as a function of time is obtained from the following differential equation.

$$\frac{dp}{dt} = \left(\frac{-1}{\tau}\right) p + \omega(t) \tag{79}$$

The solution is

$$p(t) = e^{\frac{-(t-t_0)}{\tau}} p(t_0) + \int_{t_o}^{t} e^{\frac{-(t-\zeta)}{\tau}} \omega(\zeta) d\zeta \tag{80}$$

The variance of $p(t)$ is given by

$$\sigma_p^2(t) = e^{\frac{-2(t-t_0)}{\tau}} \sigma_p^2(t_0) + \int_{t_o}^{t} e^{\frac{-2(t-\zeta)}{\tau}} \sigma_\omega^2(\zeta) d\zeta \tag{81}$$

The process noise variance may also be obtained by solution of the following differential equation,

$$\frac{d\sigma_p^2(t)}{dt} = \left(\frac{-2}{\tau}\right) \sigma_p^2(t) + \dot{q}(t) \tag{82}$$

where

$$\dot{q} = \frac{2\sigma_s^2}{\tau}$$

and σ_s^2 is the steady state noise variance. In difference equation form these give

$$p_{j+1} = M p_j + \omega_j \tag{83}$$

$$\sigma_{p_{j+1}}^2 = M^2 \sigma_{p_j}^2 + Aq \tag{84}$$

where

$$Aq = (1 - M^2)\sigma_s^2 \left(\approx \frac{2\Delta t}{\tau}\right) \sigma_s^2$$

$$M = e^{\frac{-\Delta t}{\tau}}$$

$$\Delta t = t_{j+1} - t_j$$

The data equation obtained as a result of integrating the square root information matrix from t_j to t_{j+1} is given by

$$\begin{bmatrix} R_p & R_{px} & R_{py} \\ R_{xp} & R_x & R_{xy} \\ 0 & 0 & R_y \end{bmatrix} \begin{bmatrix} p_j \\ x_{j+1} \\ y \end{bmatrix} = \hat{\eta}_j \tag{85}$$

For the discrete process noise data update, the value of the stochastic parameters (p_j) are held constant over the interval t_j to t_{j+1} while the SRIF matrix is mapped by numerical integration. At the time t_{j+1}, the process noise variance accumulated over this same time interval, is introduced via the following data equation as a discrete impulse.

$$R_\omega \hat{\omega}_j = \hat{\eta}_\omega \tag{86}$$

where

$$R_\omega = \frac{1}{\sigma_\omega}$$

Replacing $\hat{\omega}_j$ by the equation in terms of p_j and p_{j+1} we have

$$R_\omega \hat{p}_{j+1} - R_\omega M \hat{p}_j = \hat{\eta}_\omega = 0 \tag{87}$$

The updated data equation is obtained by partitioning and combining with the above noise data equation.

$$\begin{bmatrix} -R_\omega M & R_\omega & O & 0 \\ R_p & o & R_{px} & R_{py} \\ R_{xp} & o & R_x & R_{xy} \\ o & 0 & 0 & R_y \end{bmatrix} \begin{bmatrix} p_j \\ p_{j+1} \\ x_{j+1} \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{\eta}_j \end{bmatrix} \tag{88}$$

The data equation is partially triangularized over the first columns corresponding to the process noise terms to obtain

$$\begin{bmatrix} R^*_{pj} & R^*_{ppj} & R^*_{pxj} & R^*_{pyj} \\ 0 & R^+_p & R^+_{px} & R^+_{py} \\ 0 & R^+_{xp} & R^+_x & R^+_{xy} \\ 0 & 0 & 0 & R^+_y \end{bmatrix} \begin{bmatrix} p_j \\ p_{j+1} \\ x_{j+1} \\ y \end{bmatrix} = \begin{bmatrix} \hat{\eta}^*_j \\ \\ \hat{\eta}_{j+1} \\ \\ \end{bmatrix} \tag{89}$$

where the plus superscript is introduced to indicate a change in the numerical values after the process noise update. The stochastic parameter update is completed by stripping off the top rows corresponding to p_j, those containing the asterisk, and saving them along with the right side $(\hat{\eta}^*_j)$ for smoothing.

Solution Algorithm

The filter is first initialized with apriori information on the state and stochastic parameters, Apriori information on the constant parameters is saved and combined after the data is processed.

$$R(t_0) = \tilde{R}(t_0) \tag{90}$$

where

$$\tilde{R}(t_0) = \begin{bmatrix} R_{p_0} & R_{p_0 x_0} & R_{p_0 y} \\ 0 & R_{x_0} & R_{x_0 y} \\ 0 & 0 & 0 \end{bmatrix} \tag{91}$$

The reference state $(\bar{X})$ is set equal to the apriori estimate of the state $(\tilde{X})$ and the nominal state as a function of time, computed measurements, and partial derivatives arc computed for the reference state. Thus wc have for the initial data equation,

$$R(t_0)\,\hat{X}(t_0) = 0$$

Data is processed and the state and R are mapped via the current state SRIF to the final epoch t_f.

$$R(t_f)\,\hat{X}(t_f) = \hat{\eta}(t_f) \tag{92}$$

Next, the apriori on the constant parameters $(\tilde{R}_y)$ is combined with $R(t_f)$ to obtain $R_s(t_f)$, the SRIF matrix for the solution at t_f The solution is given by

$$\hat{X}_s(t_f) = R_s^{-1}(t_f)\,\hat{\eta}_s(t_f) \tag{93}$$

$$X_s(t_f) = \bar{X}(t_f) + \hat{X}_s(t_f) \tag{94}$$

The solution at epoch is obtained by smoothing, or in the absence of stochastic parameters may be obtained by integrating the solution at t_f back to t_0. The procedure for obtaining smoothed estimates is described below. If the mapping of the state is linear, the solution obtained by a single iteration is the correct solution and no further processing is required, The linearity _of the mapping can be checked by passing the solution through the data using the best estimate X_s (to) as a ncw reference. If a subsequent solution is attempted and results in zero update, convergence has been achieved. Otherwise, the process is repeated until the update is acceptably small. The purpose of iteration is to base the final solution on a reference that is as close to the true state as possible. If the best estimate of the state is used as a reference, the measurement residuals are minimized and the affect of nonlinearity is also minimized.

The procedure for iterating the solution is Newton-Raphson and tbc following algorithm is used. At the conclusion of the k th iteration, the correction computed from the data equation is given by

$$\hat{X}_k(t_f) = R_k^{-1}(t_f)\,\hat{\eta}_k(t_f) \tag{95}$$

'1'dissolution is mapped back to epoch and we have for the new reference state

$$\tilde{X}_{k+1}(t_0) = \tilde{X}(t_0) + \hat{X}_{k+1}(t_0) \tag{96}$$

$$\hat{X}_{k+1}(t_0) = \sum_{i=1}^{k} \hat{X}_i(t_0)$$

The data equation at the beginning of the $k+1$ iteration is given by

$$\tilde{R}\hat{X}_{k+1}(t_0) = \hat{\eta}_{k+1}(t_0) \tag{97}$$

$$\hat{\eta}_{k+1}(t_0) = -\tilde{R}^{-1}\hat{X}_{k+1}(t_0)$$

Using the same procedure described above, the data is processed again and the state and R are mapped via the current state SRIF to the final epoch t_f. A new solution is computed and this procedure is repeated until convergence is obtained.

CONTINUOUS SRIF SMOOTHING ALGORITHM

Often, the objective of a data filter is to obtain a filtered best estimate of the state at the end of the data arc or time of the last data point. Sometimes solutions are needed at other times, such as at the beginning of the data arc or at some time interior to the data arc. These solutions are obtained by smoothing, If there is no process noise, these interior solutions may be obtained by deterministic mapping of the filtered solution. For an epoch state filter, deterministic mapping is performed by simply integrating forward in time; but, for a current state filter the integration would be backward in time.

An epoch state solution is needed for iterating solutions to minimize the effect of nonlinearity and for checking the solution by examination of post-fit data residuals. An epoch state smoothed best estimate, or for that matter a smoothed solution at any epoch, can be obtained by appending constant parameters to the state that represent the values of the dynamic state and stochastic parameters at the desired epoch [10]. These may be conveniently included at the top of the list of y parameters and ordered the same as the stochastic and dynamic state parameters they represent. Since these parameters are constant and do not affect the data, entries in the F matrix and H matrix are normally zero. The one exception is at the smoothing epoch. At this time, a constraint is placed on the SRIF matrix to force the constant smoothing parameters to equal the dynamic parameters that are being smoothed. This may be accomplished by introducing a dummy data point for each smoothed parameter that forces equality with the corresponding dynamic parameter. These dummy measurements are processed thus forcing unity correlation between the smoothed and dynamic parameters at the smoothing epoch. This type of smoother is referred to as a fixed point smoother.

The smoothed epoch state solution for the state and stochastic parameters head the list of constant y parameters and we have

$$y = \begin{bmatrix} p_0 \\ x_0 \\ y_c \end{bmatrix} \tag{98}$$

$$p_0 = p(t_0)$$

$$x_0 = x(t_0)$$

where p_0 and x_0 are the smoothed epoch state stochastic parameters and state parameters respectively and y_c represents all the other constant parameters.

A properly constrained apriori SRIF matrix is obtained by first triangularizing and inverting the portion of the apriori covariance corresponding to the state and stochastic parameters. The remainder of the SRIF matrix corresponding to all the constant parameters is filled in with zeros.

The initial apriori covariance on the state, stochastic parameters and constant estimated parameters is thus given try

$$R(t_0) = \begin{bmatrix} \tilde{R}_p & \tilde{R}_{px} & 0 \\ 0 & \tilde{R}_x & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{99}$$

For each state and stochastic parameter, a dummy measurement is processed of the form

$$\hat{z}_x = x - x_0 \tag{100}$$

$$\hat{z}_p = p - p_0 \tag{101}$$

'J'bus, the data partials for each of these dummy measurements has 1 corresponding to the current state parameter and -1 corresponding to the smoothed epoch state parameter. The data weight is set equal to a large number forcing the apriori correlation between the actual and smoothed parameters to be as c1osc to unity as possible. The apriori covariance on the remaining constant parameters may be applied when the solution is generated after all the measurements have been processed. The initial apriori covariance after the dummy measurements have been processed is given by

$$R(t_0) = \begin{bmatrix} R_{p_0} & R_{p_0 x_0} & R_{p_0 y} \\ 0 & R_{x_0} & R_{x_0 y} \\ 0 & 0 & 0 \end{bmatrix} \tag{102}$$

All the measurements are then processed and the SRIF matrix propagated to the terminal epoch (t_f). The terminal SRIF matrix is combined with the apriori on the constant parameters and a solution generated as described above. At this time, we have a filtered best estimate of the final state and stochastic parameters, a smoothed best estimate of the initial epoch state and stochastic parameters, and a best estimate of the constant parameters, For many applications this is all wc need because the direct dynamic effect of the stochastic parameters on the state is insignificant. However, for some applications, wc need the complete time history of the state and stochastic parameters and this is obtained by smoothing. In the conventional discrete SRI F formulation, smoothed best estimates of state and the stochastic parameters arc obtained by smoothing backwards from the final filtered solution. Since wc have available solutions at both ends of the data arc, wc have the option of smoothing forward or backwards, A forward smoothing algorithm is highly desirable since wc would not need to propagate the state backwards in time.

Wc start with the terminal SRIF matrix and extract the sub-matrix associated with the constant y parameters recalling that these arc headed by smoothed epoch state solutions of the state and stochastic parameters. Thus wc have for the initial apriori smoothing data equation

$$R_{sm}(t_0) \begin{bmatrix} p_0 \\ x_{\sigma 0} \\ y_c \end{bmatrix} = \hat{\eta}_0 \tag{103}$$

where

$$R_{sm}(t_0) = R_y(t_f) \tag{104}$$
$$\hat{\eta}_0 = \hat{\eta}(t_f)$$

and the dimension of the SRIF matrix has been reduced by the number of state and stochastic parameters. Next, wc integrate the square root SRIF matrix from t_0 to the time of the first filter interrupt (t_1). For the continuous process noise update, the filter interrupt times arc coincident with data point processing times. For the discrete process noise update, the filter interrupt times arc coincident with the process noise update times.

$$R_{sm}(t_1) = \int_{t_0}^{t_1} \dot{R}\, dt + R_s m(t_0) \tag{105}$$

We thus have for the case of a discrete process noise update,

$$R_{sm}(t_1)\begin{bmatrix} p_0 \\ x_1 \\ y_c \end{bmatrix} = \hat{\eta}_0 \tag{106}$$

and this data equation may be triangularized and partitioned to make room for the stochastic parameters defined at t_1.

$$\begin{bmatrix} R_{p_1}^- & 0 & R_{px_1}^- & R_{py_1}^- \\ 0 & 0 & 0 & 0 \\ 0 & 0 & R_{x_1}^- & R_{xy_1}^- \\ 0 & 0 & 0 & R_{y_c} \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ x_1 \\ y_c \end{bmatrix} = \hat{\eta}_1^- \tag{107}$$

In the notation used here, the SRIF matrix elements change as information or noise is added to the system or the matrix is retriangularized. Integer subscripts and superscripts denote the epoch of the SRIF matrix element and a minus superscript indicates values before the smoothing update. The asterisk identifies elements that are introduced for smoothing. The above data equation is combined with the data equation saved previously during filtering,

$$\begin{bmatrix} R_{p_1}^* & R_{pp_1}^* & R_{px_1}^* & R_{py_1}^* \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ x_1 \\ y_c \end{bmatrix} = \hat{\eta}_1^* \tag{108}$$

to give the SRIF matrix after the smoothing update is performed at t_1

$$\begin{bmatrix} R_p^{*0} & R_{p1}^{*0} & R_{px}^{*0} & R_{py}^{*0} \\ 0 & R_{p_1} & R_{px_1} & R_{py} \\ 0 & R_{xp_1} & R_{x_1} & R_{xy} \\ 0 & 0 & 0 & R_{y_c} \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ x_1 \\ y_c \end{bmatrix} = \hat{\eta}_1 \tag{109}$$

The SRIF matrix at t_1 is integrated to the time of the next filter interrupt (t_2). After expanding to make room for the stochastic parameters at t_2 we have

$$\begin{bmatrix} R_p^{*0-} & R_{pp}^{*0-} & 0 & R_{px}^{*0-} & R_{py}^{*0-} \\ 0 & R_{p_2}^- & 0 & R_{px_2}^- & R_{py_2}^- \\ 0 & 0 & 0 & 0 & 0 \\ 0 & R_{xp_2}^- & 0 & R_{x_2}^- & R_{xy_2}^- \\ 0 & 0 & 0 & 0 & R_{y_c} \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ x_2 \\ y_c \end{bmatrix} = \hat{\eta}_2^- \tag{110}$$

This data equation is combined with the data equation saved during filtering for smoothing at t_2 and given by

$$\begin{bmatrix} 0 & R_{p_2}^* & R_{pp_2}^* & R_{px_2}^* & R_{py_2}^* \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ x_2 \\ y_c \end{bmatrix} = \hat{\eta}_2^* \tag{111}$$

and after triangularizing we have

$$\begin{bmatrix} R_p^{*0} & R_{p1}^{*0} & R_{p2}^{*0} & R_{px}^{*0} & R_{py}^{*0} \\ 0 & R_p^{*1} & R_{p2}^{*1} & R_{px}^{*1} & R_{py}^{*1} \\ 0 & 0 & R_{p_2} & R_{px_2} & R_{py_2} \\ 0 & 0 & R_{xp_2} & R_{x_2} & R_{xy_2} \\ 0 & 0 & 0 & 0 & R_{y_c} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ x_2 \\ y_c \end{bmatrix} = \hat{\eta}_2 \tag{112}$$

Finally, since we now have all the data equations that involve p_1 combined into one SRIF matrix, we may solve for p_1. Solving for p_1 involves inverting the SRIF matrix and multiplying times the right side. The values of the stochastic parameters (p_i) are written to a file for use in subsequent filter iterations. The top rows corresponding to p_0 are discarded and we continue on to the next data point continuing in this fashion until we reach the final data point. The smoothing algorithm for the case of continuous process noise is similar to that described above only the SRIF matrices must be adjusted to reflect the mapping of stochastic parameters by numerical integration.

NUMERICAL INTEGRATION OF SRIF MATRIX

The continuous SRIF data processing algorithm involves mapping the SRIF matrix from the time of a discrete data or process noise update to the time of the next data point or process noise update, The mapping is accomplished by numerical integration of the SRIF matrix differential equation. The numerical integration is performed with a suitable algorithm. The fifth order Runge-Kutta-Fehlberg method with error control has been successfully employed. Recall the matrix differential equation derived above for the SRIF and discard the data update term.

$$\dot{R} = -RF - \frac{1}{2} RGQG^{\mathrm{T}} R^{\mathrm{T}} R \tag{113}$$

Consider the following partition.

$$\dot{R} = -\begin{bmatrix} R_d & R_{dy} \\ 0 & R_y \end{bmatrix} \begin{bmatrix} F_d \\ 0 \end{bmatrix} - \begin{bmatrix} R_p & R_{px} & R_{py} \\ 0 & R_x & R_y \\ 0 & 0 & R_y \end{bmatrix} \begin{bmatrix} \frac{1}{2} GQG^{\mathrm{T}} R_p^{\mathrm{T}} R_p \\ 0 \\ 0 \end{bmatrix} \tag{114}$$

where R_d corresponds to the dynamic parameters and the matrix F_d contains only the rows of F corresponding to the dynamic parameters. This equation simplifies to

$$\dot{R} = -R_d F_d - \frac{1}{2} R_p GQG^{\mathrm{T}} R_p^{\mathrm{T}} R_p \tag{115}$$

We only have to integrate the top rows of the SRIF matrix corresponding to the dynamic parameters and the derivative is a function of only the R_d partition of the SRIF matrix. For the simple case of exponentially correlated process noise we have,

$$F_d = \begin{bmatrix} \frac{\partial \dot{p}}{\partial p} & 0 & 0 \\ \frac{\partial \dot{x}}{\partial p} & \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} \end{bmatrix}$$

$$\frac{\partial \dot{p}}{\partial p} = \begin{bmatrix} -\frac{1}{\tau_1} & & & \\ & -\frac{1}{\tau_2} & & \\ & & \cdots & \\ & & & -\frac{1}{\tau_i} \end{bmatrix}$$

$$Q = \begin{bmatrix} \frac{2\sigma_{s1}^2}{\tau_1} & & & \\ & \frac{2\sigma_{s2}^2}{\tau_2} & & \\ & & \cdots & \\ & & & \frac{2\sigma_{si}^2}{\tau_i} \end{bmatrix}$$

and G is the identity matrix.

CONCLUSION

A continuous matrix differential equation for the SRIF has been derived and a computer algorithm developed to implement this filter. The filter algorithm performs the mapping of state and process noise by numerical integration of the SRIF matrix and admits data via a discrete least square update.

Accuracy, computational efficiency, memory requirements and simplicity of design are compared with other filter algorithms. This comparison consisted of comparing the filter algorithms at the equation level. A more rigorous comparison of filter algorithms is being pursued that involves parallel implementation. A preliminary evaluation of these competing criteria reveals no significant differences. However, the continuous SRIF is expected to out perform discrete filters with regard to accuracy and memory requirements because there is no need to compute a state transition matrix and error control may be placed directly on the elements of the SRIF matrix. Simplicity of design seems to favor the continuous SRIF particularly when the same numerical integration algorithm used for the state propagation is used to integrate the SRIF matrix.

The integration of process noise makes the continuous SRIF somewhat computationally inefficient. However, an exact result is obtained and discrete filters only approximate the integration of process noise. When process noise is admitted to the continuous filter as discrete updates, the computational efficiency is comparable to the discrete filter implementation.

ACKNOWLEDGEMENT

The work described in this paper was carried out by the Jet Propulsion Laboratory, California institute of Technology, under a contract with the National Aeronautics and Space Administration,

The author thanks V. Alwar, W. C. Masters, R. J. Muellerschoen, D. J. Scheeres, F. M. Stienon, B. G. Williams, and P. J. Wolff for many useful discussions and help with the algorithms presented here.

REFERENCES

1. J. K. Miller, "An Integrated Current State Square Root information Filter-Smoother ", Internal document, 27 November 1991.
2. M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1993.
3. G. J. Bierman, 1977, *Factorization Methods for Discrete Sequential Estimation*, Academic Press,Inc., New York.
4. D. J. Scheeres, "Derivation of a Generalized Form of the SRIF Differential Equation," Internal document, 13 July 1993.
5. P. G. Kaminski, A. E. Bryson, Jr., and S. F. Smith, "Discrete Square Root Filtering:A Survey of Current Techniques," *IEEE Transactions on Automatic Control*, Vol. *AC-16*, pp.727-736,1971.
6. J. E. Potter and R. G. Stern, "Statistical Filtering of Space Navigation Measurements"", Proc. 1963 AIAA Guidance and Control Conference, 1963.
7. P. Dyer and S. R. McReynolds, "Extension of Square Root Filtering to Include Process Noise)" *Journal of Optimization Theory and Applications*, Vol.3, pp.444-458, *1969*.
8. W. C. Masters, P. J. Breckheimer, "Updating Colored Noise and Spacecraft State from One Batch to the Next", Internal document, 19 September 1991

9. W. C. Masters, "Design and Implementation of Estimation Methods in 01)1'", Internal document, 2 December 1991

10. J, K. Miller, "A Single Point Smoothing Algorithm for Athena with Application to Mariner Temple 2", IOM 314.4-543, 22 September 1986.