Defense Technical Information Center Compilation Part Notice

ADP011192

TITLE: Comparison of Batch and Kalman Filtering for Radar Tracking

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: The Annual AIAA/BMDO Technology Conference [10th] Held in Williamsburg, Virginia on July 23-26, 2001. Volume 1. Unclassified Proceedings

To order the complete compilation report, use: ADB273195

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP011183 thru ADP011193 ADP204784 thru ADP204818

Comparison of Batch and Kalman Filtering for Radar Tracking

Haywood Satz, Technical Staff
Raytheon Company
Bedford, MA 01730

Thomas H. Kerr, Ph.D., Consultant
TeK Associates
Lexington, MA 02420

ABSTRACT

Radar tracking performance was compared among two choices of statistical filtering algorithms for the noisy measurements of exo-atmospheric objects in ballistic motion. Such motion is characteristic of satellites and missiles. Object position and velocity were governed by the nonlinear dynamics of body motion in a central force field, and measurements were modeled as nonlinear observations of those object motions in Cartesian coordinates.

The two choices of statistical filtering algorithms were distinguished by their method of handling a sequence of noisy measurements. The first processed measurements, one-at-a-time, in a sequential recursive estimation using the Extended Kalman Filter (EKF), and the second processed that same sequence of measurements, simultaneously, in a batch-least-squares (BLS) estimation algorithm.

Both algorithms used first-variation approximations of the nonlinear observations and error dynamics of object motion. Taylor series expansions were centered about the current best estimates of the state vector. The EKF used those approximations to implement the often referenced linear-minimum-variance (Kalman) estimation formulas. The BLS processed those same measurements simultaneously in a least-squares search over object trajectories spanning the tracking interval, and initial state estimation was based on convergence to the best object path.

Results were obtained for both algorithms performing in a desktop program with a reasonable degree of radar systems modeling, and in a comprehensive mission simulator where radar system errors were represented in greater detail. Those included radar-cross-section fluctuations, scan angle loss, antenna gain patterns, radar signal-to-noise sensitivity, monopulse measurement errors, and front-end noise. The BLS algorithm was seen to converge faster, and predict more accurate 1-sigma values, than the EKF in all comparisons.

INTRODUCTION

Batch processing, as an alternative to minimum-variance statistical filtering, was described in Reference [1] (Chang) where it was applied to estimation of ballistic trajectories with Angle-Only tracking. Least-squares iterations were used to converge to an estimate of the trajectory which brought the performance measure to a local extreme point within the space of candidate object paths. Computer trials demonstrated better performance than with EKF tracking, and estimation-error predictions approached the Cramer-Rao bound.

Reference [2] (Hough) analyzed EKF object tracking, initialized by a single least-squares estimate of the object state. Analysis of a simplified model (of object tracking) was used to support the case for batch initialization. EKF performance was also improved, independently, by modeling the distribution of orbit parameter variations as they were transformed by nonlinear object motion dynamics.

The algorithms of both Chang and Hough computed measurement residuals by including only zero-order derivatives of the Taylor series expansion of the observation model,

$$\hat{\mathbf{h}}(\mathbf{x}) \approx \mathbf{h}(\hat{\mathbf{x}})$$

The consequence of that model approximation was to estimate the state without the benefit of first-order corrections which lessen the influence of nonlinear truncation errors.

Reference [3] (Bancroft) used an approach similar to that of Chang above to track space objects, offline, with a phased array early warning radar. Bancroft's algorithm differs from Chang's in that it includes first-variations of the observation model expansion:

$$\hat{\mathbf{h}}(\mathbf{x}) \approx \mathbf{h}(\hat{\mathbf{x}}) + \{\partial\mathbf{h}(\hat{\mathbf{x}})/\partial\mathbf{x}\}\delta\mathbf{x}$$

and thus estimates the state with smaller truncation errors. It also differs from Hough's algorithm in that it does an iterative least-squares search among candidate trajectories to converge to that path which brings the performance measure to a local extreme point of the function space. Those differences appear to be responsible for the algorithm's improved estimation accuracy in the environment of nonlinear models of motion dynamics and observations.

In this analysis, the Batch algorithm calculated an estimate of the current state, and its associated error covariance, in synchronism with EKF estimates, so as to compare with EKF convergence. The Batch algorithm's purpose was to find that estimated object path, $\hat{\mathbf{x}}(t)$, among many candidate choices over the tracking interval, such that the measurement residual, $[\mathbf{z} - \hat{\mathbf{z}}]$, approached a minimum. Exo-atmospheric free-fall motion, as a unique solution of the initial-value problem, allowed the identification of $\hat{\mathbf{x}}(t)$ with $\hat{\mathbf{x}}(t_o)$ at the initial time, t_o. Initial state error, $\delta\mathbf{x}(t_o)$, was obtained by multiple iterations, where the prior estimate of initial state, $\hat{\mathbf{x}}(t_o)$, was updated as:

$$[\hat{\mathbf{x}}(t_o)]^+ = \hat{\mathbf{x}}(t_o) + \delta\hat{\mathbf{x}}(t_o)$$

after each iteration to improve the least-squares estimation of the initial state error.

Error Dynamics

Object motion was constrained by the dynamic model of exo-atmospheric free-fall, in the Earth-centered-rotating (ECR) frame, where Ω and $\mathbf{g}$ were Earth's rotation rate and gravity field, respectively:

$$d\mathbf{p}/dt = \mathbf{v}$$
$$d\mathbf{v}/dt = \mathbf{g} - \Omega x(\Omega x\mathbf{p}) - 2\Omega x\mathbf{v}$$

Filter states, their estimates conditioned on measurements, $\mathbf{Z}^m$, and their estimation errors were defined as:

$$\mathbf{x} = [\mathbf{p}^T \mid \mathbf{v}^T]^T$$
$$\hat{\mathbf{x}} = E\{\mathbf{x} \mid \mathbf{Z}^m\}$$
$$\delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$$

The state estimation error was constrained, within both algorithms, by the dynamics of a first-variation approximation:

$$\frac{d}{dt}\delta\mathbf{x} = \mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}}) \approx \frac{\partial\mathbf{f}(\hat{\mathbf{x}})}{\partial\mathbf{x}}\delta\mathbf{x}$$

Solutions, $\delta\mathbf{x}(t) = \Phi(t, t_o)\delta\mathbf{x}(t_o)$, were obtained by the transition matrix, $\Phi(t, t_o)$, implied by those linearized dynamics, and satisfying:

$$\frac{d\Phi}{dt} = \frac{\partial f(\hat{\mathbf{x}})}{\partial \mathbf{x}}\Phi, \quad \Phi(t_o, t_o) = \mathbf{1}_n$$

Time histories of $\hat{\mathbf{x}}(t)$ and $\Phi(t, t_o)$ were obtained by simultaneous solution of their coupled differential equations over the tracking interval, $[t_o, t_m]$. Those were used in Batch least-squares processing to estimate the initial state error, $\delta\hat{\mathbf{x}}(t_o)$.

Process noise, representing the random environment of EKF error dynamics, was chosen at a level to partially compensate for truncation errors resulting from nonlinear approximations. It was assigned, as noise in the velocity dynamics equation shown above, to be a small percentage of the nominal gravity field vector magnitude. The Batch algorithm modeled observations without dynamic extrapolation between them, and thus did not include process noise compensation.

The gravity field model included the second spherical harmonic of the Earth's mass distribution non-homogeneity. That term added only small percentages of variation in the otherwise constant orbit momentum over object tracking intervals. It seems not to be required in a self contained desktop simulation, however, it would be essential to include it when making comparisons with other simulations, or in processing data containing those effects.

Observation Model

Measurements of range, z_r, and its unit-vector projections into the plane of the phased array radar, z_u and z_v, were modeled as nonlinear observations of states, in terms of the object range unit vector, $\mathbf{u} = \mathbf{r}/r$, rows, $[\mathbf{c}_i]^T$, of the ECR -to-phased-array-plane coordinate rotation, $[C_e^f]$, and the range/Doppler coupling coefficient, τ, to be:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} z_r \\ z_u \\ z_v \end{bmatrix} = \begin{bmatrix} r + \tau\mathbf{v}^T\mathbf{u} \\ [\mathbf{c}_1]^T\mathbf{u} \\ [\mathbf{c}_2]^T\mathbf{u} \end{bmatrix}$$

Measurement sensitivities to state variations were found to be:

$$\partial \mathbf{z}/\partial \mathbf{x} = \begin{bmatrix} \mathbf{u}^T + \tau\mathbf{v}^T(\mathbf{1}_3 - \mathbf{u}\mathbf{u}^T)/r & \tau\mathbf{u}^T \\ ([\mathbf{c}_1]^T - z_u\mathbf{u}^T)/r & \mathbf{0}^T \\ ([\mathbf{c}_2]^T - z_v\mathbf{u}^T)/r & \mathbf{0}^T \end{bmatrix}$$

and measurement residuals were determined by truncating the measurement model beyond first variations:

$$\hat{\mathbf{z}} = \hat{\mathbf{h}}(\mathbf{x}) \approx \mathbf{h}(\hat{\mathbf{x}}) + \{\partial \mathbf{h}(\hat{\mathbf{x}})/\partial \mathbf{x}\}\delta \mathbf{x}$$

Measurement errors are a function of tracking waveform parameters, and signal-to-noise power ratio (SNR). SNR, in turn, is dependent on object range, r, radar cross-section, σ, pulse-width, τ, and radar sensitivity, S, resulting from phased array radar power-aperture data. SNR and range and angle observation variances were computed for inclusion in both algorithms:

$$\mathrm{SNR} = S\sigma\tau/r^4$$
$$\sigma_r^2 = (c/2BK_r)^2/2\mathrm{SNR}$$
$$\sigma_u^2 = \sigma_v^2 = (\theta_{3dB}/K_m)^2/2\mathrm{SNR}$$

in terms of radar waveform parameters of band-width, beam-width, and range-detection and monopulse slope coefficients, B, θ_{3dB}, K_r, and K_m, respectively.

Batch vs. EKF

The EKF propagates the filter state between measurements, and incorporates measurements sequentially, with the error dynamics and observation models, respectively. It is an implementation of the often referenced linear-minimum-variance (Kalman) formulas, adapted by first-variation approximations of the nonlinear models, centered at the current state estimate, and described in Reference [4] (Jazwinski). The Batch algorithm, in comparison, processes those same measurements simultaneously via iterative least-squares estimation.

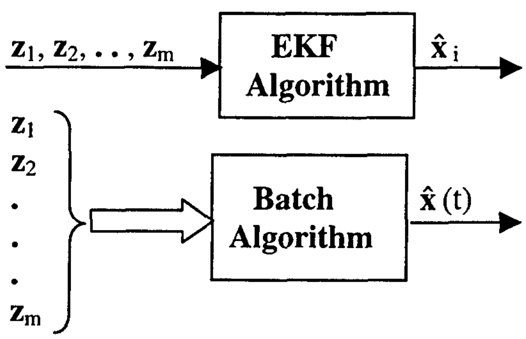

Figure-1 Comparison of Measurement Processing for Batch and EKF Tracking

An overview of the comparison of measurement processing among the two algorithms is shown in Figure-1. Batch algorithm estimations were made in parallel with those of the EKF for comparison of estimation error time histories. Formation of the Batch algorithm's least squares array was based on the observation model:

$$\hat{\mathbf{z}} = \hat{\mathbf{h}}(\mathbf{x}) \approx \mathbf{h}(\hat{\mathbf{x}}) + \{\partial \mathbf{h}(\hat{\mathbf{x}})/\partial \mathbf{x}\}\delta \mathbf{x}$$
$$= \mathbf{h}(\hat{\mathbf{x}}) + \{\partial \mathbf{h}(\hat{\mathbf{x}})/\partial \mathbf{x}\}\Phi(t, t_o)\delta \mathbf{x}(t_o)$$

The Batch algorithm's goal of seeking that object path, $\hat{\mathbf{x}}$, which drives the residual toward zero, $\mathbf{z} - \hat{\mathbf{h}}(\mathbf{x}) \Rightarrow \mathbf{0}$, suggested a regrouping of terms:

$$\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}) = \{\partial \mathbf{h}(\hat{\mathbf{x}})/\partial \mathbf{x}\}\Phi(t, t_o)\delta \mathbf{x}(t_o)$$

Redefining the terms above, and using a diagonal matrix of reciprocals of measurement noise standard deviations, $[\sigma^{-1}]$, to normalize the least-squares model:

$$\delta \mathbf{z}(t_i) = [\sigma^{-1}][\mathbf{z}(t_i) - \mathbf{h}(\hat{\mathbf{x}}(t_i))]$$
$$A(t_i) = [\sigma^{-1}]\,[\partial \mathbf{h}(\hat{\mathbf{x}}(t_i))/\partial \mathbf{x}]\Phi(t_i, t_o)$$

results in:

$$\delta \mathbf{z}(t_i) = A(t_i)\delta \mathbf{x}(t_o)$$

and adjoining the matrix rows generated by each measurement difference, $\delta \mathbf{z}(t_i)$, and its associated time tag, t_i, produces the composite array of equations indicated:

$$\left.\begin{matrix} \delta \mathbf{z}(t_1) & & A(t_1)\delta \mathbf{x}(t_o) \\ \delta \mathbf{z}(t_2) & & A(t_2)\delta \mathbf{x}(t_o) \\ \cdot & = & \\ \cdot & & \\ \cdot & & \\ \delta \mathbf{z}(t_m) & & A(t_m)\delta \mathbf{x}(t_o) \end{matrix}\right\} \mathbf{Z} = A\delta \mathbf{x}(t_o)$$

That model was used in multiple least-squares iterations, minimizing each $|\mathbf{Z} - \mathbf{A}\delta\hat{\mathbf{x}}(t_o)|^2$, and looking for convergence of $|\mathbf{Z}|$ to a minimum as $\delta \mathbf{x}(t_o)$ approached zero.

Each least-squares iteration sequence sought that estimate of initial state error which satisfied the Normal equations:

$$\mathbf{A}^T\mathbf{Z} = \mathbf{A}^T\mathbf{A}\delta\hat{\mathbf{x}}(t_o)$$

Rather than implement the Normal equations, it was preferable to use an effective numerical procedure to form the decomposition of the system matrix into its orthogonal and upper triangular factors, **H** and **U**, respectively, and solve for $\delta\hat{\mathbf{x}}(t_o)$ by back substitution:

$$\mathbf{Z} = \mathbf{A}\delta\hat{\mathbf{x}}(t_o) = \mathbf{HU}\delta\hat{\mathbf{x}}(t_o)$$
$$\mathbf{H}^T\mathbf{H} = \mathbf{1}_n, \qquad \mathbf{U}\text{, upper triangular}$$
$$\mathbf{H}^T\mathbf{Z} = \mathbf{U}\delta\hat{\mathbf{x}}(t_o)$$

The Householder method, Reference [5] (Golub and van Loan), was used to factor the least-squares array.

Comparison of Batch and Kalman Filtering for Radar Tracking

Desktop Analysis

Motion of a single object was simulated with orbit dynamics. Measurement errors were modeled with a constant RCS and pulse-width range equation. The EKF was initialized with a monopulse pair of range and angle measurements, estimating range rate as the quotient of range differences and their time-tag difference. Angle rates were declared to be zero. Although the Batch initialization design included the EKF first estimate, tests have shown that Batch could start as well in some instances with that same EKF monopulse-pair initialization.

Figure-2 is a position estimation error time history of an object in exo-atmospheric

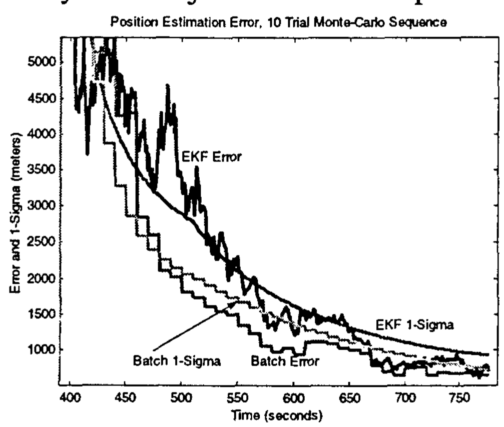

Figure-2 Desktop Simulation of 10-Trial Monte-Carlo Averaged Position Estimation Errors for Batch and EKF

free-fall, tracked at ranges of 2300-3300 km. The tracking signal-to-noise ratio ranged from 10 - 15 dB. EKF updates were at a frequency of once per second while the Batch algorithm was invoked once per 10 seconds. The error curves were the magnitudes of differences of filter state estimations with true states, and the 1-sigma curves were the square root of the sum of corresponding diagonal elements of the filter covariance matrix. Batch covariances were calculated from the least-squares model coefficient matrix, **A**, as:

$$\mathbf{C}(t)_{\text{Batch}} = \Phi(t, t_o)\,[\mathbf{A}^T\mathbf{A}]^{-1}\Phi^T(t, t_o)$$

EKF position errors were seen to exceed their statistically predicted 1-sigma values more often, and converge more slowly than those of the Batch algorithm.

Figure-3 is the velocity estimation error time history corresponding to the same

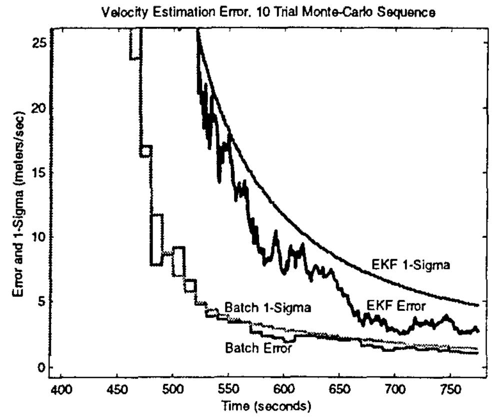

Figure-3 Desktop Simulation of 10-Trial Monte-Carlo Averaged Velocity Estimation Errors for Batch and EKF

sequence of computer trials described in Figure-2 above. Batch estimates, in comparison with those of the EKF, were again seen to converge faster and remain closer to their predicted 1-sigma values. EKF predictions of position error were optimistic (smaller) in comparison with true error paths. After 200 seconds of tracking, EKF estimation errors were at a level of about 10 meters/sec. At that same point, Batch errors were only 2.5 meters/sec.

Comparison of Batch and Kalman Filtering for Radar Tracking

Optimistic predictions of EKF estimation accuracy, such as were seen in the first 100 seconds of position estimation, have ominous implications for mission operations where it is essential to accurately assess the quality of object states. In those instances the Batch algorithm's covariance data will be more reliable than that of the EKF.

Large magnitudes of error (on the order of km/sec, not seen on the graphs) were present in the initial tracking intervals of both desktop and mission simulator time histories. Those initial errors were quite large due to the conspiracy of nonlinear approximation and cross-range errors. The desktop EKF design, in an attempt to mitigate the influence of those errors, included decoupling of range and angle errors within the update algorithm.

Reference [6] (Daum and Fitzgerald) discuss the magnification of angle errors by object range. A milli-radian of angle error multiplied by 1000 kilometers of range to the object, for example, results in a kilometer of cross range error. That magnification error also affected velocity estimation, and thus presented a challenge to estimation convergence, within both algorithms.

Mission Simulator Performance

The mission simulator program included most radar system errors, and motion dynamics such as Earth's non-uniform mass distribution, Earth's rotation, fluctuating target RCS, antenna gain patterns, front-end RF noise, radar-resource allocation and scheduling, and multiple target detections-association-with-tracks.

Figure-4 shows a velocity estimation error time history for an exo-atmospheric free-fall object generated in the mission simulator. Tracking signal-to-noise ratio was a minimum of 10 dB and the average object range was about 2500 km. EKF updates were at a frequency of one per second while the Batch algorithm was invoked once per 10 seconds. The graphs were a composite of 100 Monte-Carlo trials, showing averages of errors, 1-sigma filter predictions, and sampled-standard-deviations.

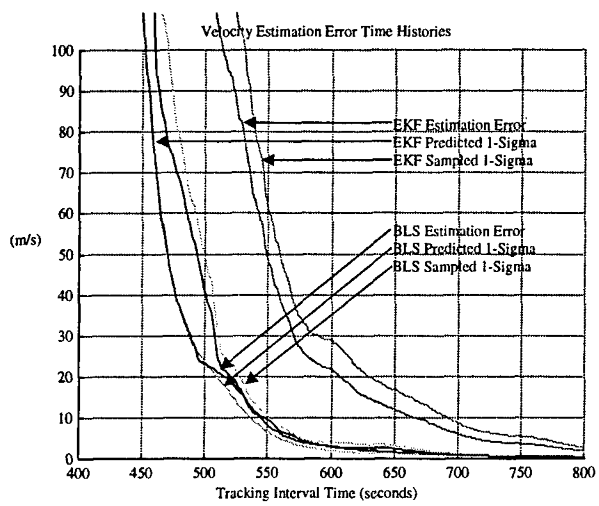

Figure-4 Mission Simulation of 100 Monte-Carlo Trials of Velocity Estimation Error

Batch estimates, the lower grouping of curves, were again seen to converge faster and remain closer to their statistically predicted 1-sigma values when compared with those of the EKF. The EKF 1-sigma predictions, expected to be near the upper grouping of curves, were instead within the lower grouping of Batch curves, indicating an error convergence, after two minutes of tracking, of about 20 meters/sec less than was actually taking place. After about 200 seconds, the differences between predicted and actual errors persisted at about 10 meters/sec. That was a further demonstration of the EKF algorithm's occasional overly-optimistic assessment of its own estimation convergence.

Comparison of Batch and Kalman Filtering for Radar Tracking

The mission simulator EKF did not include the benefit of range-angle covariance decoupling as did the desktop EKF algorithm. Its initial convergence time history could have been improved with some decoupling introduced, however, it is likely that convergence near the end of tracking would not have changed. That was seen in desktop EKF trials with varying degrees of decoupling.

CONCLUSIONS

Batch algorithm estimation of radar object tracking was compared with EKF tracking of the same object. Comparisons were made in a desktop simulation of an object in exo-atmospheric free-fall. The radar was modeled by the range equation with constant RCS and pulse-width. Observations were represented by models of monopulse range and angle measurement errors in terms of SNR.

Comparisons were also made in a more comprehensive mission simulator which included Earth's mass distribution and rotation, fluctuating targets, antenna gain patterns, RF noise, resource allocation and scheduling, and multiple-target detections-to-tracks association. The mission simulator generated a 100 Monte-Carlo sequence of tracking interval time histories to lend further support to the ten-trial desktop results.

Both the desktop and the mission simulators showed that, in comparison with EKF, the Batch algorithm converged faster, more accurately, and closer to its own self-assessed 1-sigma value.

ACKNOWLEDGEMENTS

Dr. Peter Bancroft, Raytheon Technical Staff member, developed the Batch algorithm, implemented it in a BMEWS radar, and demonstrated its utility in the offline tracking of space objects. Thong Pham, Raytheon, Technical Staff member, did a complete modeling and simulation of the algorithm, and provided the initial software implementation for the mission simulator. Mission simulator computer trials were accomplished with much help from Dan Pulido of General Dynamics, and Dmitri Tsaparas, and Dan Lawrence of XonTech, Inc.

REFERENCES

[1] Chang, C. B., "Ballistic Trajectory Estimation with Angle-Only Measurements", IEEE Automatic Control Vol. 25, No. 3, June, 1980, P.474-480

[2] Hough, M. E. "Improved Performance of Recursive Tracking Filters using Batch Initialization and Process Noise Adaptation", AIAA Journal of Guidance, Control, and Dynamics, Vol. 22, No. 5 September/October 1999, p. 676-682

[3] Bancroft, P., "Improved Prediction Accuracy Algorithm for UEWR IPT", Raytheon Final Report for XonTech, Inc., Contract 4-029394-115-8, October 1998

[4] Jazwinski, A., "Stochastic Processes and Filtering Theory", Academic Press, 1970

[5] Golub G. H., and van Loan, C. F. "Matrix Computations", Johns Hopkins University Press, 1983

[6] Daum, F. and Fitzgerald, R. J., "Decoupled Kalman Filters for Phased Array Radar Tracking", IEEE AC Vol. 28 No.3, March, 1983