Comparative Results for a Special Class of Robust Nonlinear Tracking Algorithms

Frank D. Gorecki and Michael J. Piehler
Boeing Aerospace Company, Seattle, Washington

This work presents the extension of two robust, adaptive estimators, both based on linear system theory, to the nonlinear case of an accelerating spacecraft in its ascent phase. This scenario is complicated by nonlinear system dynamics (due to the presence of the gravitational field and vehicle staging) and a nonlinear measurement model (consisting of line-of-sight range, azimuth, and elevation measurements). The first algorithm in our presentation is an adaptive batch estimator (ABE) that utilizes the innovations process to estimate a piecewise constant acceleration approximation to the real-system dynamics. The second algorithm is an iterative least-squares estimator (ILSE) that models the system dynamics with the derivative of acceleration held constant. To account for gross modeling errors, such as discontinuities encountered in the target's acceleration profile due to its staging, a stochastic detector is employed in the later estimator that allows the algorithm to be restarted in the event of divergence. In order to provide a reasonable comparison for these two algorithms, both are implemented in a realistic tracking scenario in which their robustness is demonstrated. Although the simulation results demonstrate the feasibility of using either algorithm, the accuracy of the iterative least-squares algorithm is significantly better than that of the ABE.

Introduction

THE target tracking problem has received considerable attention in the literature, with Refs. 1 and 2 providing excellent summaries. The estimation algorithms presented in this work represent an extension to these developments in that nonlinearities are addressed in both the measurement model and the system dynamics in the absence of a priori information regarding the maneuver of the target.

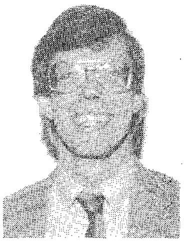
In our earlier papers,[3-5] we presented a combined extended Kalman filter (EKF) and an adaptive batch estimator (ABC) for estimating the states of an accelerating spacecraft. In these works, we extended the two-dimensional linear theory developed by Ref. 6 to the three-dimensional case consisting of nonlinear system dynamics and a nonlinear measurement model. In this paper, we summarize the theoretical developments presented in Refs. 3 and 4 and developed an ILSE for comparison purposes. These two algorithms were chosen for this comparison because they apeared to have the greatest potential for robust behavior with respect to the engagement and tracking parameters. This robustness is a key performance criterion in evaluating tracking algorithms. Performance results between the ABE and ILSE algorithms are demonstrated via Monte Carlo simulation.

The estimation problem that we are considering is illustrated in Fig. 1. A fixed observer measures line-of-sight (LOS) azimuth, elevation, and range to a *constant thrust* spacecraft ascending into orbit. This spacecraft also stages during tracking, resulting in a discontinuity in the acceleration profile as depicted in Fig. 2. Considering this as a parameter estimation problem, we are confronted with three complications:

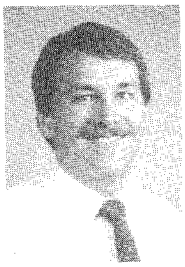
1) The nonlinear system dynamics, due to both the inverse-square law gravitational field and the nominal acceleration profile.

2) The nonlinear measurements, consisting of transcendental functions of Cartesian coordinates.

3) The discontinuities in the acceleration profile due to staging.



Frank D. Gorecki was born in May 1949. He received his Ph.D. degree in electrical engineering from the University of Washington in 1981. From 1979 to the present he has been with the Boeing Company, except for two years (1981–83) at the Shell Development Company. He has developed and applied stochastic control and nonlinear parameter estimation procedures to problems in defense and geophysics. In addition, Dr. Gorecki is an Affiliate Professor in the Electrical Engineering Department at the University of Washington. His research interests are nonlinear filtering theory and nonlinear parameter estimation, and he is a Member of AIAA.



Michael J. Piehler was born in March 1960. He received his M.S. degree in aeronautics and astronautics from the University of Washington in 1983. From 1984 to the present he has been with the Boeing Company and has worked in the areas of nonlinear filtering and parameter estimation.

Received June 1, 1987; revision received Nov. 23, 1987.

Examples of previous efforts in tracking maneuvering targets have demonstrated successful implementations of Kalman filters with acceleration states in the *linear realm* (e.g., Ref. 7). These generally function under the assumption of constant applied acceleration and, in the event that the modeled and true target dynamics are not consistent, include process noise to account for the modeling errors. Unfortunately, this requires a priori statistical information regarding the maneuver of the tracked object.

Algorithms such as the EKF and Gaussian second-order filter (GSOF) address the nonlinearities of the measurement and target dynamics models by utilizing Taylor series expansions of the governing equations. These too, however, often require process noise to compensate for modeling errors, approximations, and system discontinuities, thus increasing the dependence of tracking performance on a priori information.

The estimators presented in this paper do not rely on a priori process noise and therefore do not require "tuning" (adjusting of the process noise) for specific applications. We feel that this robustness lends itself well to applications in which autonomy is required.

In the following sections, we present the model formulation, adaptive batch estimator, iterative least-squares estimator, simulation results, and finally our conclusions.

Model Formulation

Platform Dynamics Model

Following the definitions in Refs. 3–5, we assume that the spacecraft dynamics are given by

$$\mathrm{d}\boldsymbol{x}/\mathrm{d}t = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{u}) \tag{1}$$

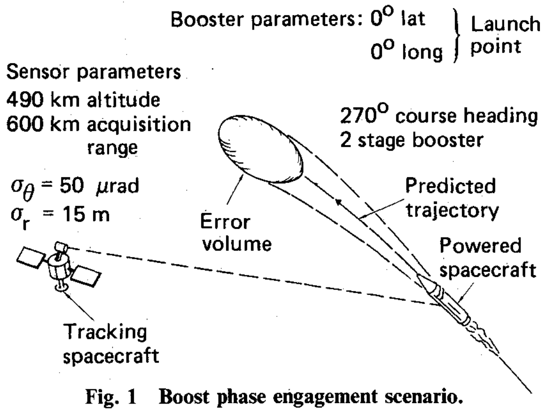


Fig. 1 Boost phase engagement scenario.

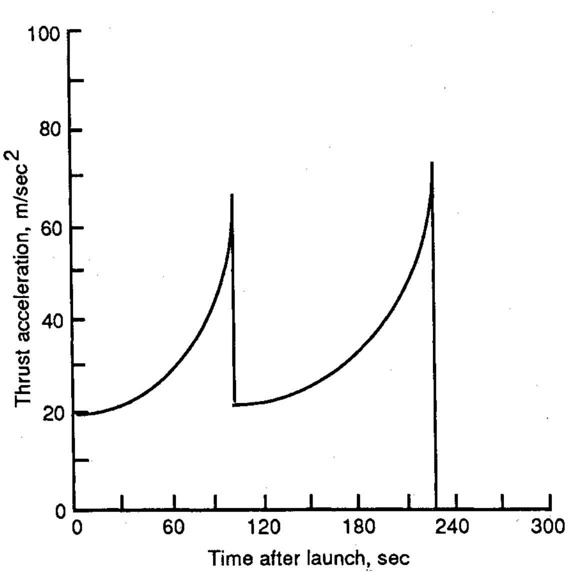


Fig. 2 Nominal booster acceleration profile.

where $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{b}(\boldsymbol{u})$ are nonlinear terms due to gravity and spacecraft thrust, respectively. The system state is defined as

$$\boldsymbol{x} = [\boldsymbol{r}^T \quad \boldsymbol{v}^T]^T \tag{2}$$

which represents a composite position and velocity vector realized in Earth-centered inertial (ECI) coordinates, with $[-]^T$ denoting the vector transpose operation. The *ballistic* dynamics are given by

$$\boldsymbol{f}(\boldsymbol{x}) = [\boldsymbol{v}^T - \mu \boldsymbol{r}^T/r^3]^T, \qquad r = |\boldsymbol{r}| \tag{3}$$

where μ is the gravitational parameter ($\mu = 3.986005 \times 10^{14}$ m^3/s^2).

The target in our example is flying in an exoatmospheric regime on a gravity turn. Physically, this means that the target-state vector is being propagated in the absence of aerodynamic lift and drag and that the trust vector is oriented along the Earth-relative velocity vector. Thus, the unit ECI orientation vector of the applied acceleration is given by

$$\boldsymbol{d} = [\boldsymbol{v} - (\boldsymbol{r} \times \boldsymbol{\omega})]/|\boldsymbol{v} - (\boldsymbol{r} \times \boldsymbol{\omega})| \tag{4}$$

where $\boldsymbol{\omega}$ is the Earth rotation rate vector (oriented along the ECI z axis, where $|\boldsymbol{\omega}| = 7.292115 \times 10^{-05}$ rad/s). The applied acceleration is therefore

$$\boldsymbol{u} = \frac{T}{m_s - T(t - t_s)/(g_0 I_{sp})} \boldsymbol{d} \tag{5}$$

where T is the constant thrust magnitude of the current stage, m_0 is the initial mass of the vehicle at staging (or at launch) corresponding to time t_0, g_0 is the sea-level acceleration due to gravity, and I_{sp} is the specific impulse of the stage. The time history of the magnitude of $\boldsymbol{u}$ for our example is depicted in Fig. 2.

Consistent with intuition, it follows that

$$\boldsymbol{b}(\boldsymbol{u}) = \begin{bmatrix} 0_3 \\ I_3 \end{bmatrix} \boldsymbol{u} \tag{6}$$

where 0_n and I_n are the $n \times n$ zero and identity matrices, respectively.

Measurement Model

The LOS measurements consist of azimuth, elevation, and range based in sensor-centered local level (LL) coordinates. The LL coordinate system is assumed to be a relative coordinate system with the x, y, and z axes oriented in the directions of east, north, and up, respectively. Azimuth is measured in the x-y plane clockwise from north and elevation is measured up from the x-y plane.

In discrete form, the measurement at time t_k is modeled as

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{\rho}_k) + \boldsymbol{w}_k, \qquad R_k = E[\boldsymbol{w}_k \, \boldsymbol{w}_k{}^T] \tag{7}$$

where $\boldsymbol{w}_k$ is an uncorrelated, additive, white, zero-mean, Gaussian noise vector of covariance R_k, $\boldsymbol{h}$ is the measurement function, and $\boldsymbol{\rho}_k$ is the LL relative-position vector (spacecraft to sensor platform). The LL relative-position vector is given by

$$\boldsymbol{\rho}_k = [\rho_1 \quad \rho_2 \quad \rho_3]^T = P(\boldsymbol{r}_k - \boldsymbol{s}_k) \tag{8}$$

where $\boldsymbol{s}_k$ is the ECI sensor-state vector and P the ECI-to-LL transformation matrix. This coordinate transformation is defined as

$$P = \begin{bmatrix} -\sin\phi & \cos\phi & 0 \\ -\cos\phi \sin\lambda & -\sin\phi \sin\lambda & \cos\lambda \\ \cos\phi \cos\lambda & \sin\phi \cos\lambda & \sin\lambda \end{bmatrix} \tag{9}$$

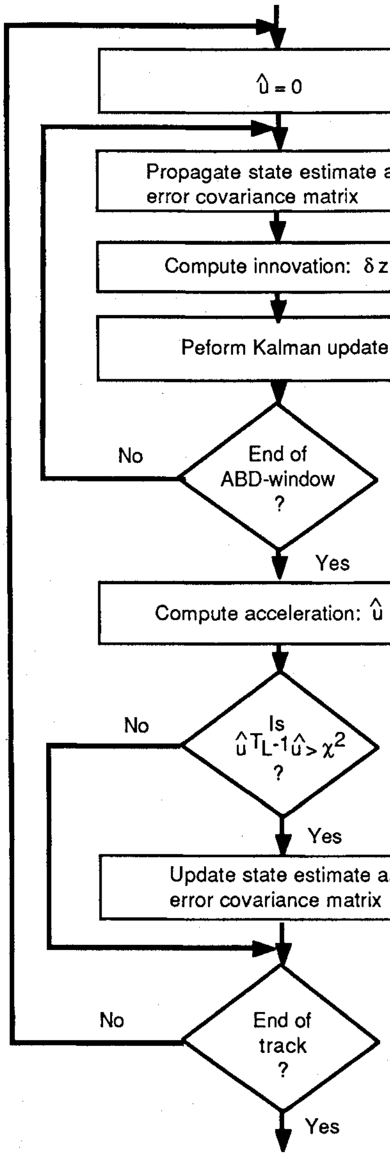

$\hat{u} = 0$
Propagate state estimate and error covariance matrix
Compute innovation: δz
Peform Kalman update
End of ABD-window ?
No
Yes
Compute acceleration: $\hat{u}$
Is $\hat{u}^T L^{-1} \hat{u} > \chi^2$?
No
Yes
Update state estimate and error covariance matrix
End of track ?
No
Yes

Fig. 3 Adaptive batch estimator mechanization.

where λ and ϕ are the latitude and inertial longitude of the sensor platform, respectively. The measurement function $\boldsymbol{h}$ is given by

$$\boldsymbol{h}(\rho_k) = \begin{bmatrix} \alpha \\ \delta \\ \rho \end{bmatrix} = \begin{bmatrix} \tan^{-1}(\rho_2/\rho_1) \\ \sin^{-1}(\rho_3/\rho) \\ |\rho_k| \end{bmatrix} \tag{10}$$

where α is the azimuth, δ the elevation, and ρ the range.

Adaptive Batch Estimator

The scheme employed by the ABE involves propagating the state vector and error-covariance matrix under the assumption of zero applied acceleration across a given time interval consisting of m measurements. This time interval is hereafter referred to as the "ABE window." In the event that the tracked vehicle has an applied acceleration, the innovations sequence produced by EKF updates will be divergent across the ABE window. When this occurs, the sequence of innovations vectors is then placed in a least-squares batch estimator and an update is performed on the state vector and error-covariance matrix. This updated state then becomes the initial state for the next ABE window and the process is repeated until the end of track (see Fig. 3).

In order to understand the assumptions behind the ABE design, we must first consider a simple linear example and then extend it to the nonlinear case. For a linear system defined by the composite position and velocity vector of Eq. (2) that is propagating at a constant velocity from time t_k to t_{k+1}, we note that the position and velocity vectors are given by

$$\boldsymbol{r}(t_{k+1})|_{u=0} = \boldsymbol{r}(t_k) + \boldsymbol{v}(t_k)\tau \tag{11a}$$

$$\boldsymbol{v}(t_{k+1})|_{u=0} = \boldsymbol{v}(t_k) \tag{11b}$$

where the subscripted notation $u = 0$ denotes propagation under the assumption of no applied acceleration, and

$$\tau = t_{k+1} - t_k \tag{12}$$

If a maneuver of constant acceleration $\boldsymbol{u}$ were to begin *at* time t_k, it is obvious that propagating the same state vectors $\boldsymbol{r}(t_k)$ and $\boldsymbol{v}(t_k)$ to time t_{k+1} would be done according to

$$\boldsymbol{r}(t_{k+1})|_{u=\text{const}} = \boldsymbol{r}(t_k) + \boldsymbol{v}(t_k)\tau + \boldsymbol{u}\tau^2/2 \tag{13a}$$

$$\boldsymbol{v}(t_{k+1})|_{u=\text{const}} = \boldsymbol{v}(t_k) + \boldsymbol{u}\tau \tag{13b}$$

where the subscripted notation $u = \text{const}$ denotes propagation in the presence of applied constant acceleration. From this, it is clear that

$$\boldsymbol{x}(t_k)|_{u=\text{const}} = \boldsymbol{x}(t_k)|_{u=0} \tag{14}$$

and thus we are able to conclude that

$$\boldsymbol{x}(t_{k+1})|_{u=\text{const}} = \boldsymbol{x}(t_{k+1})|_{u=0} + \Gamma_k \boldsymbol{u} \tag{15}$$

where

$$\Gamma_k = \begin{bmatrix} \tau^2/2I_3 \\ \hdashline \tau I_3 \end{bmatrix} \tag{16}$$

In order to extend this to the nonlinear case, we begin by directly integrating Eq. (1), under the assumption of no applied acceleration, which yields

$$\boldsymbol{x}(t_{k+1})|_{u=0} = \boldsymbol{x}(t_k) + \int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{x})\,\mathrm{d}t \tag{17}$$

Linearizing Eq. (17) leads to

$$\boldsymbol{x}(t_{k+1})|_{u=0} \approx \Phi_k \boldsymbol{x}(t_k) \tag{18}$$

where Φ_k is the linearized, discrete-time state transition matrix given by the truncated Taylor series

$$\Phi_k = I_6 + F_k \tau \tag{19}$$

with F_k being the Jacobian of the system dynamics [see Eq. (3)]

$$F_k = \partial \boldsymbol{f}/\partial \boldsymbol{x} = \left[\begin{array}{cc} 0_3 & I_3 \\ \hdashline -\mu/r^3[I_3 - 3\boldsymbol{r}\boldsymbol{r}^T/r^2] & 0_3 \end{array}\right]_{\boldsymbol{r}=\boldsymbol{r}_k} \tag{20}$$

If we consider a vehicle undergoing a *constant applied acceleration* (as opposed to a *constant applied thrust*), we can make the approximation

$$\boldsymbol{x}(t_{k+1})|_{u=\text{const}} \approx \Phi_k \boldsymbol{x}(t_k) + \Gamma_k \boldsymbol{u} \tag{21}$$

by noting Eqs. (15) and (16). Using Eq. (21) as the linearized system dynamics equations forms the foundation for the ABE in our application.

Since the ABE has no a priori information regarding the vehicle's acceleration profile, propagation of both the estimate of the system state vector and the error-convariance matrix across the ABE window is performed under the assumption of zero applied acceleration, or

$$\hat{x}_{k+1|k}\big|_{u=0} = \hat{x}_{k|k}\big|_{u=0} + \int_{t_k}^{t_{k+1}} f(x)\,\mathrm{d}t \tag{22a}$$

$$\Sigma_{k+1|k}\big|_{u=0} = \Phi_k \Sigma_{k|k}\big|_{u=0}\Phi_k^T \tag{22b}$$

where it is understood that

$$\hat{x}_{k+1|k} = E[x_{k+1}|z_1, z_2, z_3, \ldots, z_k]$$

(the expectation of x_{k+1} conditioned on the measurement set $\{z_j : j = 1, k\}$).

The Kalman update is performed on the system state vector and error-covariance matrix estimates by

$$\hat{x}_{k+1|k+1}\big|_{u=0} = \hat{x}_{k+1|k}\big|_{u=0} + K_{k+1}[z_{k+1} - h(\hat{\rho}_{k+1})] \tag{23a}$$

$$\Sigma_{k+1|k+1}\big|_{u=0} = [I - K_{k+1}H_{k+1}]\Sigma_{k+1|k}\big|_{u=0} \tag{23b}$$

where

$$\hat{\rho}_{k+1} = [\hat{\rho}_1 \quad \hat{\rho}_2 \quad \hat{\rho}_3] = P[\hat{r}_{k+1|k}\big|_{u=0} - s_{k+1}] \tag{24}$$

the Kalman gains matrix is

$$K_{k+1} = \Sigma_{k+1|k}\big|_{u=0}[H_{k+1}]^T[D_{k+1}]^{-1} \tag{25}$$

and the innovations covariance matrix is

$$D_{k+1} = H_{k+1}\Sigma_{k+1|k}\big|_{u=0}[H_{k+1}]^T + R_{k+1} \tag{26}$$

The measurement Jacobian matrix is determined by linearizing Eq. (7) while noting Eq. (10) and the fact that h is LL-based whereas x is ECI-based. Thus,

$$H_{k+1} = \partial h/\partial x = \left[\begin{bmatrix} \hat{\rho}_2/\hat{\rho}_{12} & -\hat{\rho}_1/\hat{\rho}_{12} & 0 \\ -\hat{\rho}_1\hat{\rho}_3/(\hat{\rho}_{12}\hat{\rho}) & -\hat{\rho}_2\hat{\rho}_3/(\hat{\rho}_{12}\hat{\rho}) & \hat{\rho}_{12}/\hat{\rho} \\ \hat{\rho}_1/\hat{\rho} & \hat{\rho}_2/\hat{\rho} & \hat{\rho}_3/\hat{\rho} \end{bmatrix} P \;\vdots\; 0_3\right] \tag{27}$$

where

$$\hat{\rho}_{12} = \sqrt{(\hat{\rho}_1)^2 + (\hat{\rho}_2)^2} \qquad \text{and} \qquad \hat{\rho} = |\hat{\rho}_{k+1}|$$

If, at the beginning of the ABE window, it is assumed that

$$\hat{x}_{k|k}\big|_{u=0} = \hat{x}_{k|k}\big|_{u=\text{const}} \tag{28}$$

[see Eq. (14)], then a least-squares minimization of

$$J = [\delta y - \Psi\hat{u}]^T\Omega^{-1}[\delta y - \Psi\hat{u}] \tag{29}$$

with respect to $\hat{u}$ leads to an "optimal" solution of $\hat{u}$ (see Ref. 3). Here,

$$\delta y = \begin{bmatrix} z_{k+1} - h(\hat{\rho}_{k+1}) \\ z_{k+2} - h(\hat{\rho}_{k+2}) \\ z_{k+3} - h(\hat{\rho}_{k+3}) \\ \vdots \\ z_{k+m} - h(\hat{\rho}_{k+m}) \end{bmatrix}, \quad \Psi = \begin{bmatrix} H_{k+1}B_{k+1}C_k\Gamma_{k+1} \\ H_{k+2}B_{k+2}C_{k+1}\Gamma_{k+2} \\ H_{k+3}B_{k+3}C_{k+2}\Gamma_{k+3} \\ \vdots \\ H_{k+m}B_{k+m}C_{k+m-1}\Gamma_{k+m} \end{bmatrix} \tag{30}$$

where

$$B_i = I_6 - K_iH_i$$

$$C_i = I_6 + \Phi_iB_iC_{i-1} \quad \text{for} \quad i > k \quad \text{and} \quad C_k = I_6 \tag{31}$$

and

$$\Omega = \begin{bmatrix} D_{k+1} & & & & \bigcirc \\ & D_{k+2} & & & \\ & & D_{k+3} & & \\ & & & \ddots & \\ \bigcirc & & & & D_{k+m} \end{bmatrix} \tag{32}$$

The generalized least-squares estimate of u based on Eq. (29) is

$$\hat{u} = [\Psi^T\Omega^{-1}\Psi]^{-1}\Psi^T\Omega^{-1}\delta y \tag{33}$$

In order to determine if an adaptive update should be performed, a divergence test is necessary. Noting that the covariance of $\hat{u}$ is

$$L = [\Psi^T \quad \Omega^{-1} \quad \Psi]^{-1} \tag{34}$$

we may define a chi-squared quantity in the form

$$\chi_a^2 = \hat{u}^TL^{-1}\hat{u} \tag{35}$$

such that when χ_a^2 is *greater* than some predetermined threshold, the adaptive update is performed. In that event, the adaptive updates for the state estimate and error-covariance matrix are given by

$$\hat{x}_{k+m|k+m}\big|_{u=\text{const}} = \hat{x}_{k+m|k+m}\big|_{u=0} + M\hat{u} \tag{36a}$$

$$\Sigma_{k+m|k+m}\big|_{u=\text{const}} = \Sigma_{k+m|k+m}\big|_{u=0} + MLM^T \tag{36b}$$

where

$$M = B_{k+m}C_{k+m-1} \tag{37}$$

This summarizes the development and implementation of the ABE. Next, we develop a polynomial-time ILSE.

Iterative Least-Squares Estimator

The scheme employed by the ILSE involves a given sequence of m measurements from which we wish to seek a state-vector estimate that corresponds to the mth measurement. A smooth-

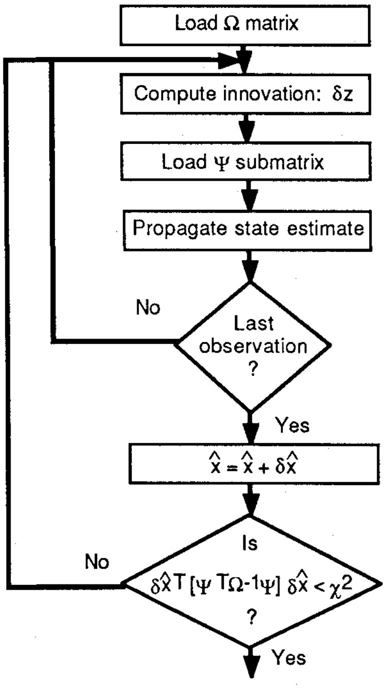


Fig. 4 Iterative least-squares estimator mechanization.

ing process is repeated under some assumption of the system dynamics until a desired degree of convergence is met (see Fig. 4). Because this algorithm is nonrecursive in nature, the process must be repeated each time a new measurement is added to the measurement set.

Presented next is a heuristic development of an ILSE (Refs. 8 and 9 provide excellent discussions of ILS theory). As we begin, let us recall a few concepts from the previous two sections. Defining the estimate of the measurement vector to be

$$\hat{z}_k = h(\hat{p}_k) \tag{38}$$

we may expand this equation in a first-order Taylor series to obtain

$$\hat{z}_k + \delta\hat{z}_k = h(\hat{p}_k) + H_k\delta\hat{x}_k + \varepsilon_k$$

where ε_k represents any higher-order terms in $\delta\hat{x}_k$. This, of course, reduces to

$$\delta\hat{z}_k = H_k\delta\hat{x}_k + \varepsilon_k \tag{39}$$

If we are given a sequence of m measurements, $\{z_k, z_{k-1}, \ldots, z_{k-m+1}\}$, and an initial estimate of $x_k, \hat{x}_k^o$, the problem then is to estimate the "best" $\delta\hat{x}_k$ to improve the estimate

$$\hat{x}_k = x_k^o + \delta\hat{x}_k \tag{40}$$

One possible criterion is to minimize the objective function

$$J = [\delta_y - \Psi\delta\hat{x}_k]^T\Omega^{-1}[\delta y - \Psi\delta\hat{x}_k] \tag{41}$$

with respect to $\delta\hat{x}_k$. Minimizing Eq. (41) with respect to $\delta\hat{x}_k$ yields

$$\hat{x}_k = \hat{x}_k^o + [\Psi^T \quad \Omega^{-1} \quad \Psi]^{-1}\Psi^T\Omega^{-1}\,\delta y \tag{42}$$

Without much difficulty, it can be shown that

$$\Psi = \begin{bmatrix} H_k \\ H_{k-1}\Phi_{k,k-1} \\ H_{k-2}\Phi_{k,k-2} \\ \cdot \\ \cdot \\ \cdot \\ H_{k-m+1}\Phi_{k,k-m+1} \end{bmatrix}, \quad \Omega = \begin{bmatrix} R_k & & & \bigcirc \\ & R_{k-1} & & \\ & & R_{k-2} & \\ & & & \ddots \\ \bigcirc & & & & R_{k-m+1} \end{bmatrix} \tag{43}$$

where $\Phi_{i,j}$ represents the linear-state transition matrix from t_j to t_i. Furthermore,

$$\delta y = \begin{bmatrix} \delta\hat{z}_k \\ \delta\hat{z}_{k-1} \\ \delta\hat{z}_{k-2} \\ \vdots \\ \delta\hat{z}_{k-m+1} \end{bmatrix} = \begin{bmatrix} z_k - h(\hat{p}_k) \\ z_{k-1} - h(\hat{p}_{k-1}) \\ z_{k-2} - h(\hat{p}_{k-2}) \\ \vdots \\ z_{k-m+1} - h(\hat{p}_{k-m+1}) \end{bmatrix} \tag{44}$$

represents the sequence of innovations vectors for $\{\hat{x}_k|z_k, z_{k-1}, \ldots, z_{k-m+1}\}$. Because this innovations sequence requires state-vector estimates at each of the measurements, a knowledge of the system dynamics is necessary for state integration. Since we are interested in the state estimate at time t_k, our smoothing process requires us to integrate *backward* all the way to the time of the first measurement.

Because the ILSE algorithm is a nonrecursive estimator that functions somewhat like an iterative curve-fitting algorithm, it was desired to increase the order of the system dynamics model of the ABE to the constant in jerk (acceleration rate) rather than in acceleration. This increase in order was implemented to boost the fidelity of this iterative algorithm for the same general reason that a higher-order polynomial is used to obtain a better curve fit.

Since this least-squares algorithm allows for a state vector but no control vector, the acceleration and jerk terms are appended to the state vector defined in Eq. (2) to yield

$$x = [r^T \quad v^T \quad u^T \quad j^T]^T \tag{45}$$

Note that with this increase in order, six additional columns of zeros must be appended to the measurement Jacobian matrix defined by Eq. (27) and that the state-transition matrix in this development is dimensioned 12 × 12 rather than 6 × 6.

The assumptions behind ILSE dynamics follows similarly to those of the ABE as demonstrated in Eqs. (11–16), the prime differences being an increase in order and the direction of propagation (i.e., *backward* in time). This development is reiterated for the benefit of the reader.

For a system defined by the composite vector of Eq. (45) that is propagating at a constant velocity from time t_{i+1} to t_i, we note that

$$r(t_i)|_{u=j=0} = r(t_{i+1}) + v(t_{i+1})\tau \tag{46a}$$

$$v(t_i)|_{u=j=0} = v(t_{i+1}) \tag{46b}$$

$$u(t_i)|_{u=j=0} = u(t_{i+1}) = [0 \quad 0 \quad 0]^T \tag{46c}$$

where

$$\tau = t_i - t_{i+1} \tag{47}$$

Now propagating the *same* position, velocity, and acceleration state vectors from time t_{i+1} to t_i with the application of constant jerk j, we have

$$r(t_i)|_{j=\text{const}} = r(t_{i+1}) + v(t_{i+1})\tau + u(t_{i+1})\tau^2/2 + j\tau^3/6 \tag{48a}$$

$$v(t_i)|_{j=\text{const}} = v(t_{i+1}) + u(t_{i+1})\tau + j\tau^2/2 \tag{48b}$$

$$u(t_i)|_{j=\text{const}} = u(t_{i+1}) + j\tau \tag{48c}$$

From this, we are able to conclude that

$$x(t_i)|_{j=\text{const}} = x(t_i)|_{u=j=0} + \Gamma_{i+1}u_{i+1} + \Delta_{i+1}j \tag{49}$$

where

$$\Gamma_{i+1} = \begin{bmatrix} \tau^2/2I_3 \\ \tau I_3 \\ I_3 \\ 0_3 \end{bmatrix}, \qquad \Delta_{i+1} = \begin{bmatrix} \tau^3/6I_3 \\ \tau^2/2I_3 \\ \tau I_3 \\ I_3 \end{bmatrix} \tag{50}$$

If we extend Eq. (49) to the nonlinear case, our integration formula may be stated as

$$\hat{x}_i|_{j=\text{const}} = \hat{x}_{i+1}|_{u=j=0} + \int_{t_{i+1}}^{t_i} f(x)\,\mathrm{d}t + \Gamma_{i+1}\hat{u}_{i+1} + \Delta_{i+1}\hat{j} \tag{51}$$

(noting the direction of propagation). It is noteworthy that $f(x)$ is the same as that given in Eq. (3) with the addition of rows 7–12 (to accommodate the increase in order), each containing zeros.

As in the development of the ABE algorithm, the state-transition matrix used in this development may be approximated with a truncated first-order Taylor series expansion. During the course of our investigation, however, the use of the Jacobian of $f(x)$ in this approximation proved to be numerically burdensome in that many iterations were required for convergence. We then chose the motion described by Eqs. (48) to determine the state-transition matrix. This surprisingly yielded results identical to the method that employed a modified form of Eqs. (19) and (20), but with fewer iterations.

The discrete-time state-transition matrix based on Eq. (48) is

$$\Phi_{i+1,i} = \begin{bmatrix} 1 & 0 & 0 & \tau & 0 & 0 & \tau^2/2 & 0 & 0 & \tau^3/6 & 0 & 0 \\ & 1 & 0 & 0 & \tau & 0 & 0 & \tau^2/2 & 0 & 0 & \tau^3/6 & 0 \\ & & 1 & 0 & 0 & \tau & 0 & 0 & \tau^2/2 & 0 & 0 & \tau^3/6 \\ & & & 1 & 0 & 0 & \tau & 0 & 0 & \tau^2/2 & 0 & 0 \\ & & & & 1 & 0 & 0 & \tau & 0 & 0 & \tau^2/2 & 0 \\ & & & & & 1 & 0 & 0 & \tau & 0 & 0 & \tau^2/2 \\ & & & & & & 1 & 0 & 0 & \tau & 0 & 0 \\ & & & & & & & 1 & 0 & 0 & \tau & 0 \\ & & & & & & & & 1 & 0 & 0 & \tau \\ & & & \bigcirc & & & & & & 1 & 0 & 0 \\ & & & & & & & & & & 1 & 0 \\ & & & & & & & & & & & 1 \end{bmatrix} \tag{52}$$

Because of the linearization and modeling assumptions that were made in this development, iteration on Eq. (42) is necessary to yield the best solution. For convergence, each iteration should yield a successively smaller value for $\delta\hat{x}_k$. In order to determine if convergence has occurred, a chi-squared quantity of the form

$$\chi_b^2 = \delta\hat{x}_k^T[\Psi^T \quad \Omega^{-1} \quad \Psi]\delta\hat{x}_k \tag{53}$$

is compared against a suitable threshold such that when χ_b^2 is *less* than this threshold, iteration is terminated. Although this test is adequate for a convergence test, it does nothing for testing "goodness of fit." To account for any gross modeling errors such as unmodeled staging, a stochastic detector is employed using another chi-squared quantity in the form

$$\chi_c^2 = \sum_{i=1}^{m} [\delta\hat{z}_{k-m+i}]^T[R_{k-m+1}]^{-1}\delta\hat{z}_{k-m+i} \tag{54}$$

such that when χ_c^2 is *greater* than its appropriate threshold, the ILSE is restarted, treating the mth measurement as the first in the new sequence.

Simulation Studies

Simulation Test Case

The test case employed for this sensitivity study is shown in Fig. 1. The observation platform is fixed at the location 490 km altitude, 3.0° latitude, and 0.5° longitude, whereas the accelerating spacecraft is launched from 0° latitude, 0° longitude. Sensor-acquisition is defined at 50 s into the trajectory of the target, at which time the adaptive algorithms begin estimating. Measurements are then obtained at the stated rates until about 180 s after launch.

For the ABE, the covariance matrix Σ was initialized to

$$\Sigma_0 = \mathrm{diag}\{\sigma_p^2,\sigma_p^2,\sigma_p^2,\sigma_v^2,\sigma_v^2,\sigma_v^2\}$$

$$\sigma_p = 1500\ \mathrm{m}, \qquad \sigma_v = 30\ \mathrm{m/s}$$

and the initial state-vector estimate was initialized with the foregoing statistics. The initial position estimate for the ILSE was determined from the initial measurement, whereas the velocity, acceleration, and jerk estimates were initialized to zero. The measurement variances for both algorithms were

$$\sigma_{az}^2 = \sigma_{el}^2 = (50\ \mu\mathrm{rad})^2$$

$$\sigma_r^2 = (15\ \mathrm{m})^2$$

These conditions were used as the baseline for the entire study.

Results for Fifty Monte Carlo Trials

In Ref. 5, we presented results describing the performance of the ABE algorithm. These results, in the form of a measurement-update rate-sensitivity study, are recalled for the purpose of demonstrating the performance characteristics of the ABE algorithm. For a 3-s ABE window, frame times of 0.5, 1.0, and 1.5 s were chosen in an effort to gain a better understanding of the sensitivity of the ABE algorithm to the measurement update rate. Performance results for 50 Monte Carlo trials are shown in Figs. 5–7. Root-mean-square (rms) position and velocity-estimation errors are shown in Figs. 5 and 6, whereas

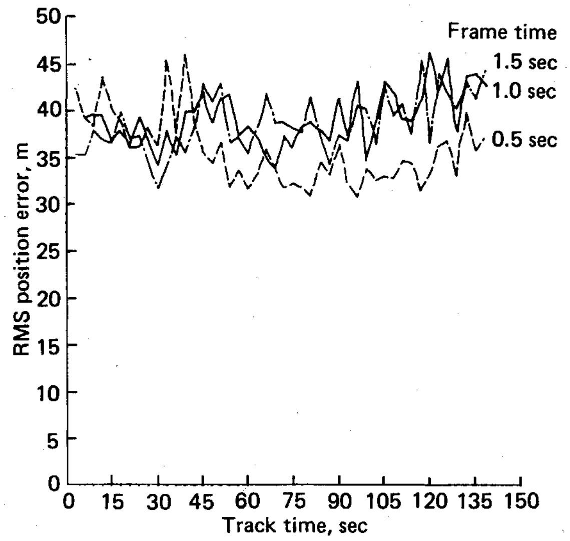


Fig. 5 ABE rms position error.

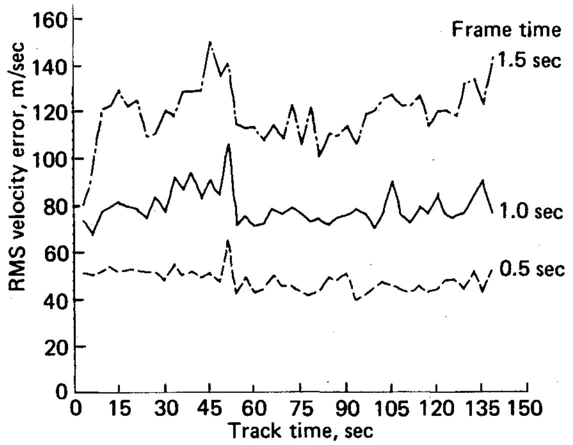


Fig. 6 ABE rms velocity error.

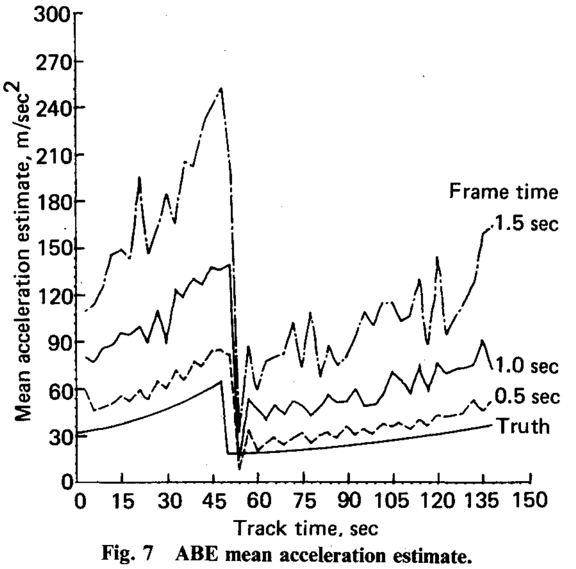


Fig. 7 ABE mean acceleration estimate.

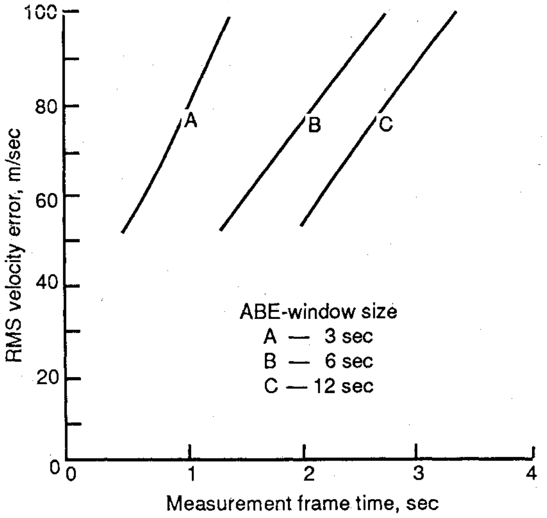


Fig. 8 ABE end-of-track rms velocity error.

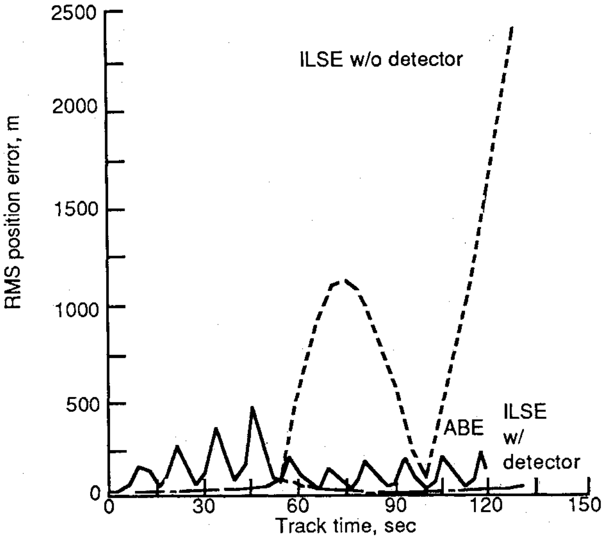


Fig. 9 Comparative rms position error results.

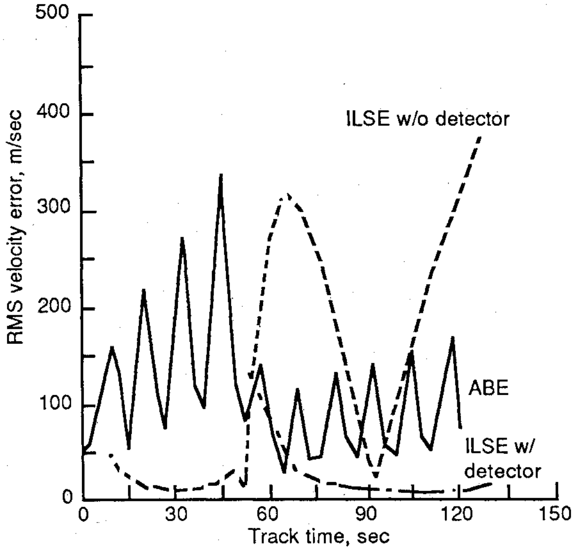


Fig. 10 Comparative rms velocity error results.

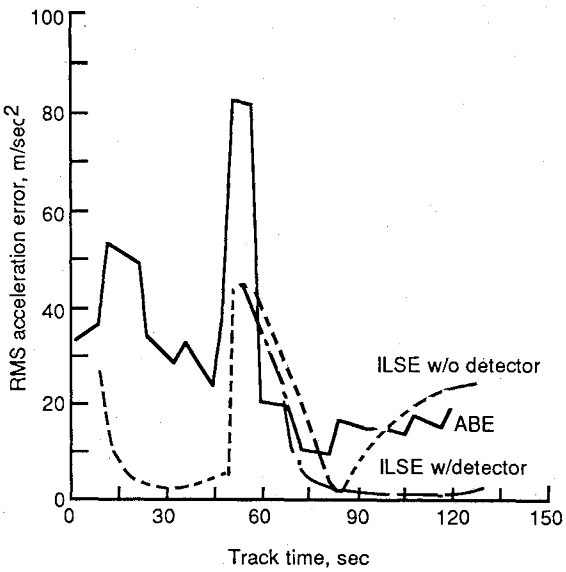


Fig. 11 Comparative rms acceleration error results.

the mean of the acceleration estimate is shown in Fig. 7. Although there is a noticeable "spike" in each of these error histories around the staging time (about 50 s into track), the algorithm demonstrates its robustness in that divergence does not occur. Consistent with intuition, shorter frame times yielded lower estimation errors (due to the increased number of measurements in the ABE window). This influence is less dramatic with the rms position error as depicted in Fig. 5. This is reasonable in light of the fact that the position errors are primarily due to the measurement accuracy of the tracking sensor. Thus, in Fig. 5 the position errors are relatively insensitive to frame time. However, frame time becomes an important parameter as derivatives (velocity and acceleration) of the measured quantities (positions) are formed. These derivatives are estimated as first and second differences with the frame time providing the time scale for the differencing. As expected, Figs. 6 and 7 show that the estimator performance is strongly dependent on frame time.

Often, it is important to predict the estimate of the state after the end of track (EOT). In this event, it is necessary to have sufficiently low-velocity errors at the EOT condition because even small velocity errors here will likely induce large position errors at the prediction point. In evaluating ABE performance at this condition, there are three time scales of interest: the time interval between measurements; the time interval over which batch estimates are averaged; and the total track time (or equivalently the total number of measurements). Figure 8 summarizes the results of a study of the sensitivity of the system to both frame time and three different window sizes (3, 6, and 12 s). As depicted in this diagram, there is an equivalence in performance between shorter frame times with smaller batch windows and longer frame times with larger batch windows. Thus, if one uses smaller batch windows, more measurements (or shorter frame times) are required to obtain a given performance than would be if a larger window size were employed.

In order to demonstrate a comparison of the robustness of the algorithms presented in this paper, 50 Monte Carlo trials were performed and rms statistics are presented. In Figs. 9–11, the ABE performance is compared to that of the ILSE. For the ABE, the size of the ABE window was fixed at 12 s (five observations with a frame time of 3 s). Notice that vehicle staging occurs around 100 s after launch (50 s of track time), resulting in a first-order discontinuity in acceleration. This "jump" can be seen to coincide with the peak ABE error.

To illustrate the robustness of the ILSE algorithm, we chose to perform our comparison tests both with and without the stochastic detector implemented. In both cases, the frame time was the same as that of the ABE: 3 s. The ILSE without the stochastic detector shows an increase in error around 100 s after launch (50 s of track time) and, at about 150 s after launch (100 s of track time), diverges. Additional simulation results confirm the ILSE divergence lagging the acceleration jump by approximately 40–50 s.

It appears as though the ILSE algorithm without the stochastic detector tries to recover after the initial divergence due to the unmodeled staging. This accounts for the large "hump" noticed in the estimation histories. Because of the nonrecursive nature of the ILSE algorithm, however, total divergence eventually occurs. The unusual appearing divergence occurs after the humps in Figs. 9–11 due to the fact that rms error does not reflect direction. The inverted spike appearing in these same graphs just prior to total divergence is due to a change in sign in the position, velocity, and acceleration errors.

With the use of the stochastic detector, however, the ILSE algorithm is able to detect the divergence early enough in the engagement to be able to restart. Upon restarting, the ILSE algorithm converges once again to values even lower than those of the ABE.

In all three performance plots (Figs. 9–11) the ILSE (augmented with the stochastic detector) achieved significantly lower estimation errors than the ABE. Although both demonstrate approximately equal robustness, the differences in estimation performance are due to the following reasons: The higher-order model of the system dynamics in the ILSE added fidelity; the window size of the ILSE grows, whereas that of the ABE remains fixed; and the measurement information in the ABE is used only once in the update process, whereas for the ILSE each measurement is used numerous times, thus compensating for the linearization assumptions in the algorithm design. This increase in performance of the ILSE over the ABE is not without cost, however. As with any comparison between recursive and iterative algorithms, the computational burden of the iterative scheme is significantly greater due to the fact that it begins from scratch with each new measurement.

Conclusions

The simulation results demonstrate the feasibility of using either the adaptive batch estimator (ABE) or the iterative least-squares estimator (ILSE) for the robust tracking of an accelerating spacecraft during its ascent phase. Both algorithms were able to successfully track the vehicle in this inherently nonlinear environment. Even in the event of target staging, both the ABE and ILSE algorithms were able to continue tracking, although the ILSE required a stochastic detector for restarting. Because the ILSE used a higher-order system dynamics model than the ABE and because of the fact that the iteration on the measurements was able to compensate for some of the linearization assumptions, the ILSE achieved lower estimation errors than the ABE.

References

[1] Chang, C. B. and Tabaczynski, J. A., "Application of State Estimation to Target Tracking," *IEEE Transactions on Automatic Control*, Vol AC-29, Feb. 1984, pp. 98–109.

[2] Mehra, R. K., "A Comparison of Several Nonlinear Filters for Re-Entry Vehicle Tracking," *IEEE Transactions on Automatic Control*, Vol. AC-16, Aug. 1971, pp. 307–319.

[3] Gorecki, F. D., Daniels, P. D., and Hardtla, J. W., "Angle-Only State Estimation of Powered Spacecraft," *Proceedings of the 1984 AIAA Guidance and Control Conference*, AIAA, New York, 1984, pp. 116–123.

[4] Gorecki, F. D. and Piehler, M. J., "Exoatmospheric Trajectory Estimation of an Accelerating Spacecraft," American Astronautical Society, Paper AAS 85-407, Aug. 1985.

[5] Gorecki, F. D. and Piehler, M. J., "Adaptive Estimation of an Accelerating Spacecraft," *Proceedings of the 1986 Astrodynamics Conference*, AIAA, New York, 1986, pp. 283–288.

[6] Chan, Y. T., Hu, A. G. C., and Plant, J. B., "A Kalman Filter Based Tracking Scheme with Input Estimation," *IEEE Transactions on Aerospace and Electronic Systems*. Vol. AES-15, March 1979, pp. 237–244.

[7] Bar-Shalom, Y. and Birmiwal, K., "Variable Dimension Filter for Maneuvering Target Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-18, Sept. 1982, pp. 621–629.

[8] Bard, Y., *Nonlinear Parameter Estimation*, Academic Press, New York, 1974.

[9] Lawson, C. L. and Hanson, R. J., *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.