

DDC FILE COPY

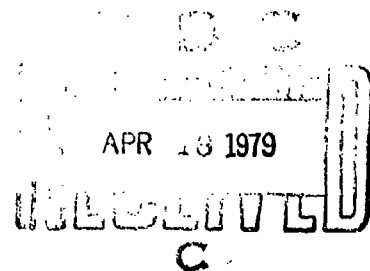
ADA067354

12

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

OPTIMAL STATE ESTIMATION OF BALLISTIC
TRAJECTORIES WITH ANGLE-ONLY MEASUREMENTS

C. B. CHANG
Group 32



TECHNICAL NOTE 1979-1

24 JANUARY 1979

Approved for public release; distribution unlimited.

LEXINGTON

MASSACHUSETTS

ABSTRACT

An iterative least square estimation algorithm is derived and applied to the problem of state estimation of ballistic trajectories with angle only measurements. A filter initiation procedure is suggested. The application of trajectory a priori knowledge for improving the estimate is discussed and solved as a constrained estimation problem. A Monte Carlo simulation study was conducted to evaluate these techniques. It was found that the iterative least square filter achieves the Cramer-Rao bound and it performs better than the extended Kalman filter when the sensor is on a free-falling platform. When the sensor is on a stationary platform however, both estimators asymptotically achieve the Cramer-Rao bound.

[illegible]

CONTENTS

1.	INTRODUCTION	1
2.	TRAJECTORY EQUATION OF MOTION	3
3.	ESTIMATION ALGORITHMS	8
3.1.	An Iterative Least Square Algorithm	8
3.2.	Initial Guess Calculation	15
3.3.	Incorporating A Priori Knowledge	21
3.3.1.	Constrained Estimation	21
3.3.2.	Incorporating the Velocity Constraint	26
3.4.	Overall Estimation Algorithms	28
4.	NUMERICAL RESULTS	33
5.	CONCLUSIONS	42
	ACKNOWLEDGMENTS	44
	REFERENCES	45

1. INTRODUCTION

The state estimation of a ballistic trajectory with angle only measurements is a challenging problem. The problem becomes even more complex when the sensor is placed on a free-falling platform. It is difficult to initiate a filter for such systems. Furthermore, the extended Kalman filter (EKF) is expected to perform poorly since the EKF performance is conditioned on being able to linearize the system about accurate state estimates.

In two previous reports, [1] and [2], the Cramer-Rao lower bound on the covariance of the trajectory estimates was presented for stationary sensor platforms [1] as well as free-falling sensor platforms [2]. Since this bound is calculated using the information matrix (the inverse of the covariance matrix) and it is not tied with any specific estimator, the results can be easily calculated and they are not restrained by the filter initiation problem. Issues remaining to be addressed include: how tight is this bound? can any filter achieve this bound?

In this report, we present an iterative least square (ILS) algorithm for estimating the state of nonlinear deterministic systems with nonlinear noisy measurements. It is known that the exo-atmospheric part of ballistic trajectories can be described by

nonlinear differential equations with zero process noise. This fact enables the above algorithm to be applied to the problem of tracking with angle-only measurements. We have also derived a procedure using a polynomial fit and vehicle-sensor dynamics for calculating filter initial conditions. We apply this procedure to initialize both ILS and EKF filters and evaluate their performances using a Monte Carlo simulation.

Since the angle-only tracking system is only "weakly" observable*, the resulting estimation error is inevitably large. It is suspected that trajectory a priori knowledge (e.g., constraints on velocity, energy, angles, etc.) may be helpful for improving the estimates. This constitutes a constrained estimation problem. We formulate this problem and present a solution procedure.

This report is organized as follows. In the next section, we state the trajectory equation of motion. In the third section, we derive the iterative least square algorithm. Also, presented are initiation and constrained estimation procedures. In the fourth section we present a numerical example illustrating the ILS and EKF performances and compare them with the Cramer-Rao bound.

*Since range is not measured.

2. TRAJECTORY EQUATION OF MOTION

Tracking with angle-only measurements is primarily concerned with exo-atmospheric trajectories. Gravity is therefore the most significant force term on the target. For the case of a free-falling sensor platform, the difference of gravity on the sensor and on the target produces curvature in the relative sensor-target trajectory and is the information enabling the target state to be estimated. For these reasons, we consider the gravity as the only driving force on the trajectory. Furthermore, we use a spherical earth model to simplify the mathematics.

Consider a Cartesian coordinate with origin at the center of the earth. The trajectory differential equations of motion are

$$\begin{aligned}\ddot{x} &= -g_0 \frac{x R_e^2}{(x^2 + y^2 + z^2)^{3/2}} \\ \ddot{y} &= -g_0 \frac{y R_e^2}{(x^2 + y^2 + z^2)^{3/2}} \\ \ddot{z} &= -g_0 \frac{z R_e^2}{(x^2 + y^2 + z^2)^{3/2}}\end{aligned}\tag{2.1}$$

where g_0 is the gravity at sea level and R_e is the radius of the earth.

The sensor measures azimuth (A) and elevation (E) angles of the target relative to the sensor, i.e.,

$$A = \tan^{-1} \frac{x}{y} \quad (2.2)$$

$$E = \tan^{-1} \frac{z}{\sqrt{x^2 + y^2}}$$

where the states used in (2.2) are the difference of the target state $\underline{x}_T$ and the sensor state $\underline{x}_s$, i.e., Eq. (2.2) is evaluated at

$$\underline{x} = \underline{x}_T - \underline{x}_s \quad (2.3)$$

When the sensor is stationary, $\underline{x}_s$ is a fixed point in space. When the sensor is free-falling, $\underline{x}_s$ is described by the same differential equation of motion Eq. (2.1) as that used for targets.

The sensor measurement noise is assumed to be a white noise sequence with covariance

$$R = \text{cov}(E, A) = \sigma^2 \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\cos^2 E} \end{bmatrix} \quad (2.4)$$

where σ is the sensor angle measurement standard deviation.

Equations (2.1) - (2.4) will be used as system and measurement equations in the iterative least square algorithm to be described in the next section. Trajectory equations of motion described in the sensor coordinates will be found useful in computing an initial guess for the iterative least square algorithm. We therefore state them below. If the sensor is free-falling, the target differential equations of motion in the sensor (R,A,E,) coordinates are

$$\ddot{R} = R(\dot{E}^2 + \dot{A}^2 \cos^2 E) - \frac{g_0 R_e^2 \sin E}{R_s^2} \left[\frac{R_s^2 (R + R_s \sin E)}{R_T^3 \sin E} - 1 \right] \quad (2.5a)$$

$$\ddot{A} = -2 \frac{\dot{R}}{R} \dot{A} \cos E + \dot{A} \dot{E} \tan E \quad (2.5b)$$

$$\ddot{E} = -2 \frac{\dot{R}}{R} \dot{E} - \frac{\dot{A}^2}{2} \sin 2E - \frac{g_0 R_e^2 \cos E}{R R_s^2} \left[\left(\frac{R_s}{R_T} \right)^3 - 1 \right] \quad (2.5c)$$

where R is the distance between target and sensor, R_T is the distance between target and the earth center, and R_s is the distance between sensor and the earth center. The R_T is related to R and R_s by

$$R_T = (R^2 + R_s^2 + 2RR_s \sin E)^{1/2} \quad (2.6)$$

Notice that the gravity appears only in R and E. The target-sensor geometry is illustrated in Fig. 2.1.

If the sensor is stationary, the gravity term on range (g_R) and elevation (g_E) are modified as

$$g_R = -g_0 \frac{R_e^2}{R_T^3} (R + R_s \sin E) \quad (2.7)$$

$$g_E = -g_0 \frac{R_e^2}{R R_T^3} R_s \cos E$$

If the sensor is stationary and the target is relatively close to the sensor, a flat earth model may suffice. The gravitational terms for the flat earth are

$$g_R = -g_0 \sin E$$

$$g_E = -\frac{g_0}{R} \cos E$$

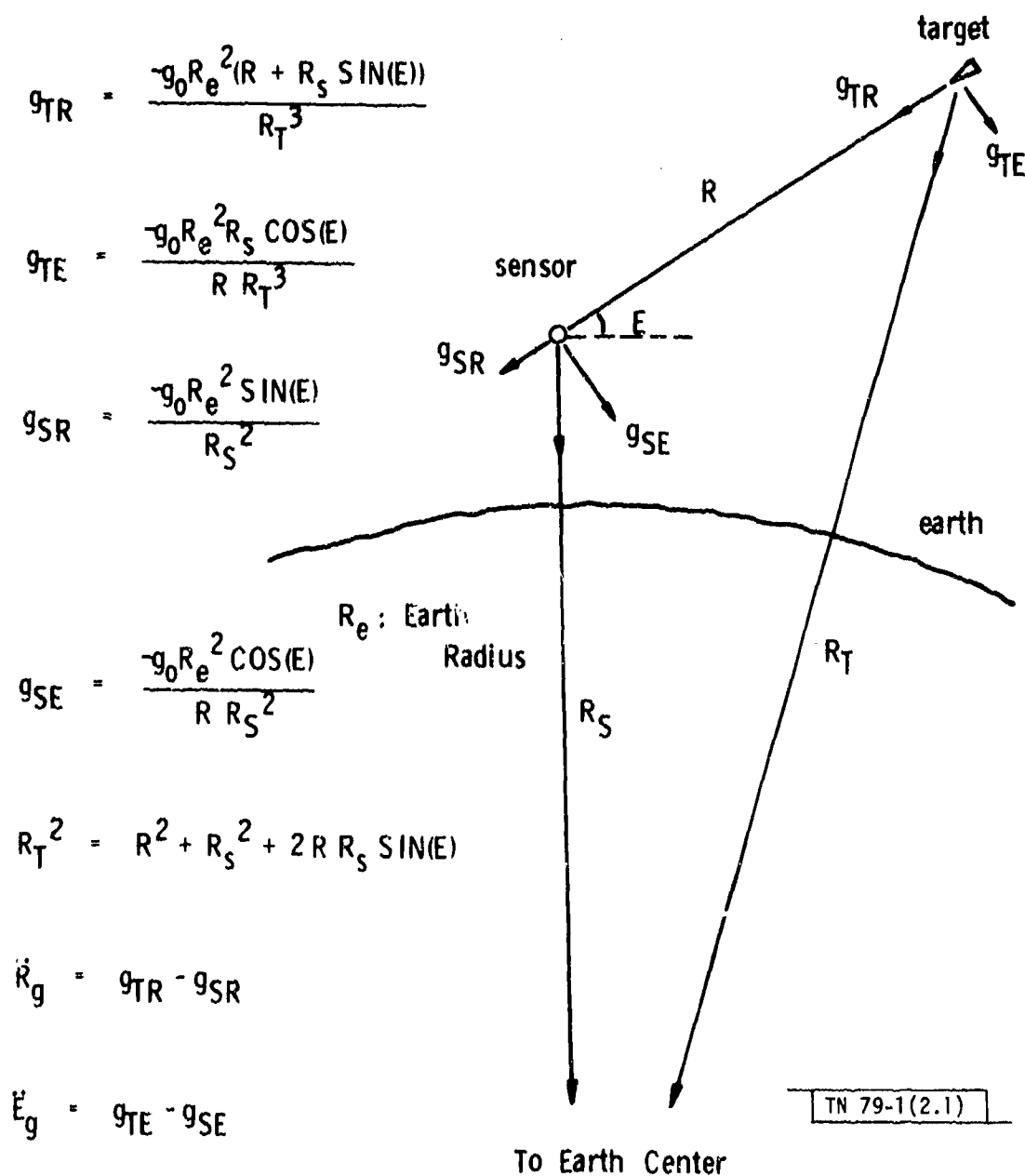


Fig. 2.1. Target-sensor geometry and relative gravity accelerations.

3. ESTIMATION ALGORITHM

In the first subsection, we describe an iterative least square algorithm applicable to state estimation with deterministic nonlinear discrete systems. The convergence of the iterative least square algorithm is dependent upon a particular application and the initial guess (initial state) for the iterative procedure. In the second subsection, we demonstrate a method for computing an initial state by using a batch of angle measurements.

In the ballistic trajectory estimation problem, several variables (e.g., re-entry velocity, re-entry angle, etc.) are known to within a certain range of values. Incorporating this a priori information with measurements to obtain a "combined" estimate constitutes a constrained estimation problem. In the third subsection, we present algorithms for calculating the constrained estimate. In the fourth subsection, we combine these analyses to present two algorithms for ballistic trajectory tracking applications.

3.1. An Iterative Least Square Algorithm

Consider the following nonlinear discrete system and measurement equations:

$$x_{n+1} = f(x_n), \quad n=1, \dots \quad (3.1)$$

$$\tilde{y}_{n+1} = h(x_{n+1}) + v_{n+1} \quad (3.2)$$

where x is the state vector, y is the measurement vector, $\tilde{y}$ is the noise corrupted measurement vector, v is the measurement

noise vector and n is the discrete time index. We assume that v_n is a time-wise uncorrelated random sequence which is Gaussian with zero mean and covariance R_n . Notice that we do not include (1) process noise and (2) unknown time-varying parameters* in Eq. (3.1). These assumptions are valid for exoatmospheric trajectories. If we relax these assumptions, then the simplicity of the ensuing algorithm will be lost.

We emphasize that Eq. (3.1) is a convenient and sufficient way of representing the trajectory estimation problem being considered. When the trajectory is described by a differential equation of motion, i.e.,

$$\dot{\underline{x}} = \underline{g}(\underline{x}); \quad \underline{x}(t_0) \quad (3.1a)$$

then one can always obtain an equivalent discrete system by using numerical integration to evaluate

$$\begin{aligned} \underline{x}(t_{n+1}) &= \underline{x}(t_n) + \int_{t_n}^{t_{n+1}} \underline{g}(\underline{x}) dt \\ &= f(\underline{x}_n) \end{aligned} \quad (3.1b)$$

The least square algorithm to be described works with a batch of measurements. Let $\hat{y}_0, \hat{y}_1, \dots, \hat{y}_N$ denote a batch of N

*The ensuing algorithm is applicable to systems with unknown constant parameter vectors.

measurements. One would like to obtain an estimated state sequence $\hat{\underline{x}}_n$, $n=1, \dots, N$ so that

$$J = \sum_{n=1}^N (\hat{\underline{y}}_n - \underline{y}_n)^T \underline{R}_n^{-1} (\hat{\underline{y}}_n - \underline{y}_n) \quad (3.3)$$

is minimized subject to constraint Eqs. (3.1) and (3.2). Since there is no process noise assumed in (3.1), one is only required to estimate the initial state (corresponding to $n=1$) $\underline{x}_1$. Furthermore, if an inverse function of $f()$, i.e.,

$$\underline{x}_n = f^{-1}(\underline{x}_{n+1}) \quad (3.4)$$

can be found, the optimal estimate of the entire trajectory can be obtained if one minimizes (3.3) with respect to any state vector along the trajectory. In our trajectory estimation application the $f^{-1}()$ function merely corresponds to integration backwards in time. We can therefore minimize (3.3) with respect to any state along the trajectory. For the purpose of convenience, we minimize (3.3) with respect to the initial state $\underline{x}_1$.

Before minimizing (3.3) can be made tractable, we introduce approximations to the system and measurement equations. Let $\hat{\underline{x}}_n^0$ denote an initial guess (estimate) of the true state $\underline{x}_n$,

we approximate the measurement equation by using a first order Taylor series expansion about $\hat{x}_n^0$,

$$y_n = h(x_n) \approx h(\hat{x}_n^0) + H_n^0 (\hat{x}_n - \hat{x}_n^0) \quad (3.5)$$

where H_n^0 is the Jacobian matrix of $h(x_n)$ evaluated at $\hat{x}_n^0$. Using the system Eq. (3.1) and iterating n times yields

$$\underline{x}_n = \underline{f}_n(\underline{x}_1) \quad (3.6a)$$

Note that if (3.1) is a linear system then $\underline{f}_n(\)$ is the product of n transition matrices and if (3.1a) is used then

$$\underline{f}_n(\underline{x}_1) = \underline{x}(t_1) + \int_{t_1}^{t_n} g(\underline{x}) dt \quad (3.6b)$$

Let $\underline{x}_1^0$ denote the initial guess (estimate) of $\underline{x}_1$, then it is related to the $\underline{x}_n^0$ used in (3.5) by

$$\hat{x}_n^0 = \underline{f}_n(\hat{x}_1^0) \quad (3.7)$$

Approximating the system equation by using a first order Taylor series expansion about $\underline{x}_1^0$ yields

$$\underline{x}_n = f_n(\underline{x}_1) \approx f_n(\hat{\underline{x}}_1^0) + F_n^0(\hat{\underline{x}}_1^0 - \underline{x}_1^0) \quad (3.8)$$

where F_n^0 is the Jacobian matrix of $f_n(\)$ evaluated at $\underline{x}_1^0$.

We now minimize (3.3). Substituting (3.5), (3.7), and (3.8) in (3.3) yields

$$J \approx \sum_{n=1}^N (\tilde{y}_n - [h(\hat{\underline{x}}_n^0) + H_n^0 F_n^0(\hat{\underline{x}}_1^0 - \hat{\underline{x}}_1^0)])^T R_n^{-1} \cdot (\tilde{y}_n - [h(\hat{\underline{x}}_n^0) + H_n^0 F_n^0(\hat{\underline{x}}_1^0 - \hat{\underline{x}}_1^0)]) \quad (3.9)$$

Taking the derivative of J with respect to $\underline{x}_1$ and solving for $\underline{x}_1$ yields

$$\hat{\underline{x}}_1 = \hat{\underline{x}}_1^0 + \left[\sum_{n=1}^N F_n^{0T} H_n^{0T} R_n^{-1} H_n^0 F_n^0 \right]^{-1} \left[\sum_{n=1}^N F_n^{0T} H_n^{0T} R_n^{-1} (\tilde{y}_n - h(\hat{\underline{x}}_n^0)) \right] \quad (3.10)$$

Notice that we have used the notation $\hat{x}_1$ to replace x_1 . This algorithm, if it converges, produces an estimate $\hat{x}_1$ which is better than the initial guess $\hat{x}_1^0$. If we use $\hat{x}_1$ in the place of the initial guess and go through the calculation again, we will receive a further-improved estimate (provided that the algorithm converges). Thus we have obtained an iterative algorithm,

$$\hat{x}_1^{k+1} = \hat{x}_1^k + \left[\sum_{n=1}^N F_n^k T H_n^k R_n^{-1} H_n^k F_n^k \right]^{-1} \left[\sum_{n=1}^N F_n^k T H_n^k R_n^{-1} (\hat{y}_n - h(\hat{x}_n^k)) \right] \quad (3.11)$$

and the covariance of $\hat{x}_1^{k+1}$ is

$$\text{cov}(\hat{x}_1^{k+1}) = \left[\sum_{n=1}^N F_n^k T H_n^k R_n^{-1} H_n^k F_n^k \right]^{-1} \quad (3.12)$$

The iteration is terminated when the difference of two successive performance indices (Eq. (3.3)) is below a certain threshold.

We make the following remarks:

1. This iterative algorithm processes a batch (N points) of data. It is iterating over the same batch of data in attempting to minimize the weighted least square error. If it converges and is terminated with a finite number of iterations, it produces near optimum estimates. This algorithm is fundamentally different from the commonly known recursive algorithms such as the extended Kalman filter (EKF).

2. The convergence property of this algorithm is determined by a) the properties of $f(\cdot)$ and $h(\cdot)$ and b) the initial guess $\hat{x}_1^0$. Deriving an analytical convergence criterion for the above algorithm is a rather complicated problem. In our angle-only tracking application, numerical experiences indicate that if the measurement noise is sufficiently low then a properly computed $\hat{x}_1^0$ always converges to the optimal estimate $\hat{x}_1$.

3. The covariance equation Eq. (3.12) has the identical functional form as the Cramer-Rao lower bound on the covariance of trajectory estimation (Refs. [1] and [2]). The difference is that the Cramer-Rao bound is evaluated using the true states while Eq. (3.12) is evaluated using the estimated states. This implies that if the least square estimate converges asymptotically (when the number of measurements is large) then its covariance approaches the Cramer-Rao bound.

4. The covariance equation is apparently related to the observability condition for nonlinear discrete systems. We note that this subject is an area still not very well understood. Notice that the matrix to be inverted in (3.12) may be singular. If it is singular for all N , then this algorithm fails and the system is not observable in the weighted Euclidean norm sense.

5. Once the estimate at the initial time is found, one may calculate state estimates at any arbitrary time by simply applying Eq. (3.1). The associated covariance can be approximated by using

$$\text{cov}(\hat{x}_n) = F_n \text{cov}(\hat{x}_1) F_n^T \quad (3.13)$$

The above procedure is valid because there is no process noise associated with the system dynamics. If one attempts to generalize the above algorithm by introducing process noise, a much more complicated optimization problem results. Fortunately, for the exo-atmospheric trajectory estimation problem, the process noise is negligible.

6. One may use the above procedure to compute an initial estimate for a general recursive tracking filter such as the Extended Kalman filter. In the angle-only measurement trajectory estimation problem, track initiation is often difficult. The above procedure seems to provide a reasonable approach for this application.

7. Due to its simplicity, we suggest the following procedure for tracking application.

7.1 Use the smallest N , so that the matrix in the bracket of (3.12) is nonsingular, to compute an estimate at initial time. The estimate at any arbitrary time can be obtained by applying the trajectory equation.

7.2 When the $(N+1)$ st measurement is available, use the estimate obtained in (6.1) above as an initial guess then apply the algorithm. Because this initial guess is the optimum estimate for N measurements, it converges to the optimal estimate for $N+1$ measurements very quickly.

We will demonstrate an algorithm for computing the initial guess used in (6.1) for the angle-only measurement case in the next subsection.

3.2. Initial Guess Calculation

In the above section, we have illustrated an iterative least square algorithm which can be used to estimate states of ballistic trajectories with angle only measurements. This algorithm, if it converges, gives near optimum estimates. Its convergence is, however, hinged on properly choosing an initial guess. In this section, we suggest a procedure for computing an initial guess using a batch of data.

This procedure is illustrated in Fig. 3.1. A batch of angle measurements, $(\hat{A}_n, \hat{E}_n)$, $n=1, \dots, N$, is first smoothed by a second order polynomial to obtain their derivatives. The polynomial fit procedure can be found in many standard textbooks. A brief but fairly general analysis was presented in an earlier report [1]. We simply state the applicable results below. Let $\hat{\theta}_n$, $n=1, \dots, N$ denote N angle measurements (which can be either $\hat{A}$'s or $\hat{E}$'s), then the $\hat{\theta}$, $\hat{\dot{\theta}}$, and $\hat{\ddot{\theta}}$ corresponding to the center of the data interval can be obtained by

$$\begin{aligned} \hat{\theta} = & \frac{3}{4} \frac{(3N^2-7)}{(N-2)N(N+2)} \sum_{n=1}^N \hat{\theta}_n \\ & - \frac{30}{T^2(N-2)N(N+2)} \sum_{n=1}^N T^2(n-1 - \frac{(N-1)}{2})^2 \frac{\hat{\theta}_n}{2} \end{aligned} \quad (3.14a)$$

$$\hat{\dot{\theta}} = \frac{12}{T^2(N-1)N(N+1)} \sum_{n=1}^N T(n-1 - \frac{(N-1)}{2}) \hat{\theta}_n \quad (3.14b)$$

$$\begin{aligned} \hat{\ddot{\theta}} = & \frac{-30}{T^2(N-2)N(N+2)} \sum_{n=1}^N \hat{\theta}_n \\ & + \frac{720}{T^4(N-2)(N-1)N(N+1)(N+2)} \sum_{n=1}^N T^2(n-1 - \frac{(N-1)}{2})^2 \frac{\hat{\theta}_n}{2} \end{aligned} \quad (3.14c)$$

where T is time between measurements.

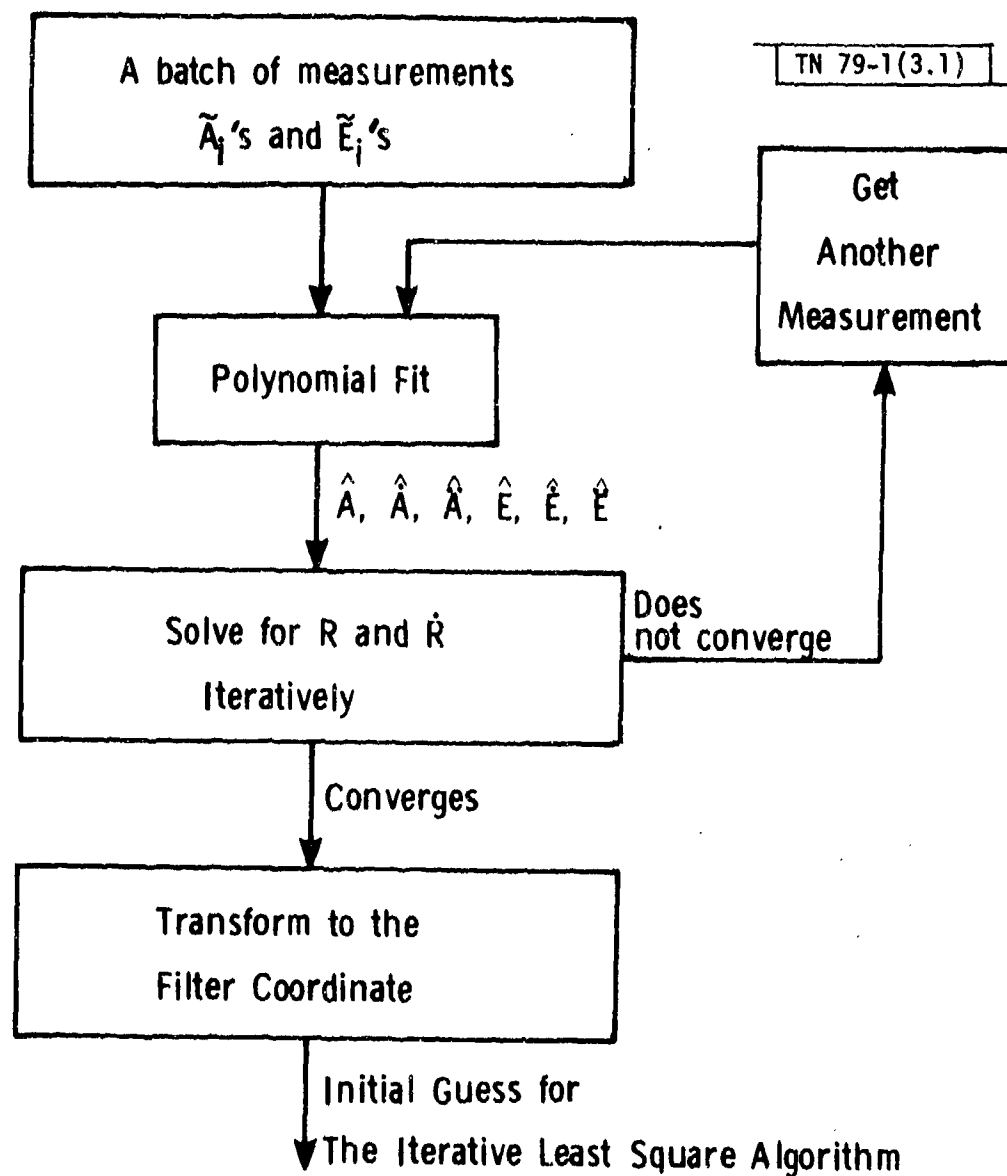


Fig. 3.1. Initial guess calculation procedure.

With the smoothed angles, one may apply eqs. (2.5b) and (2.5c) to compute R and $\dot{R}$. These two equations are restated below.

$$\ddot{A} = -2 \frac{\dot{R}}{R} \dot{A} \cos E + \dot{A} \dot{E} \tan E \quad (3.15a)$$

$$\ddot{E} = -2 \frac{\dot{R}}{R} \dot{E} - \frac{\dot{A}^2}{2} \sin 2E - \frac{g_o R_e^2 \cos E}{R R_s^2} \left[\left(\frac{R_s}{R_T} \right)^3 - 1 \right] \quad (3.15b)$$

Notice that the above are two simultaneous nonlinear equations with two unknowns, R and $\dot{R}$.^{*} They may be solved using iterative algorithms such as Picard's method or Newton's method [4]. The details are omitted here.

There is another method useful for solving R and $\dot{R}$. This second method seems to have better convergence property and it is the algorithm used in the simulation study. This is to use the fact that the total energy of a ballistic object is unchanged throughout its exoatmospheric flight. Let E_t denote the normalized (by mass) average total energy of a intercontinental ballistic missile, it is equal to the sum of its kinetic energy (E_k) and potential energy (E_p).

$$E_t = E_k + E_p = \frac{1}{2} V_T^2 + g_o R_e \left(\frac{R_T - R_e}{R_T} \right) \quad (3.16)$$

^{*}If the term involving g_o is absent, the equations contain only the combination R/R and cannot be solved for R and $\dot{R}$ separately.

where V_T is the target velocity, g_0 the gravitational constant, R_e the radius of the earth and R_T the distance between the target and the center of the earth. Let

$$V_T = \begin{bmatrix} \dot{x}_T \\ \dot{y}_T \\ \dot{z}_T \end{bmatrix} = \text{The target velocity vector in a earth centered Cartesian coordinate}$$

$$V_s = \begin{bmatrix} \dot{x}_s \\ \dot{y}_s \\ \dot{z}_s \end{bmatrix} = \text{The sensor velocity vector in a earth centered Cartesian coordinate}$$

then

$$\underline{V}_T = \begin{bmatrix} \dot{x}_s + aR \\ \dot{y}_s + bR \\ \dot{z}_s + cR \end{bmatrix} \quad (3.17)$$

$$\begin{aligned} \text{where} \quad a &= \frac{\dot{R}}{R} \cos E \sin A - \dot{E} \sin A \sin E + \dot{A} \cos E \cos A \\ b &= \frac{\dot{R}}{R} \cos E \cos A - \dot{E} \cos A \sin E - \dot{A} \sin A \cos E \\ c &= \frac{\dot{R}}{R} \sin E + \dot{E} \cos E \end{aligned} \quad (3.18)$$

Substituting (3.18) and (3.17) in (3.16) and using average total missile energy, one may solve for R and $\dot{R}$ together with Eq. (3.15a).

With $R, \dot{R}, A, \dot{A}, E, \dot{E}$, one may calculate the state vector in a sensor centered Cartesian coordinate using the following equations

$$\begin{aligned}
 x &= R \sin A \cos E \\
 y &= R \cos A \cos E \\
 z &= R \sin E \\
 \dot{x} &= \dot{R} \cos E \sin A - R \dot{E} \sin A \sin E + R \dot{A} \cos E \cos A \\
 \dot{y} &= \dot{R} \cos E \cos A - R \dot{E} \cos A \sin E - R \dot{A} \sin A \cos E \\
 \dot{z} &= \dot{R} \sin E + R \dot{E} \cos E
 \end{aligned} \tag{3.19}$$

where $\underline{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ is the state vector in sensor centered Cartesian coordinates. It is then transformed to earth centered Cartesian coordinate by

$$\underline{x}_T = \underline{x} + \underline{x}_g \tag{3.20}$$

where $\underline{x}_T$ and $\underline{x}_g$ are target and sensor states in earth centered Cartesian coordinates, respectively.

In the next section, we illustrate a technique of incorporating the trajectory a priori knowledge with the state estimate by means of constrained estimation.

3.3. Incorporating A Priori Knowledge

In the ballistic trajectory estimation problem, several trajectory variables such as re-entry velocity, re-entry angle, etc., are known to within a certain range of values. Incorporating this a priori information with measurements to obtain a "combined" estimate constitutes a constrained estimation problem. We will discuss the constrained estimation problem in Sec. 3.3.1. Previous analytical studies [1,2] indicated that the re-entry velocity constraint is the most significant a priori knowledge and it precludes all other constraints. For this reason an algorithm for explicitly incorporating the velocity constraint is discussed in Sec. 3.3.2.

3.3.1. Constrained Estimation

Let Ω denote a known constraint set which the state vector $\underline{x}$ must satisfy. The constrained estimate of $\underline{x}$ based upon both Ω and measurements $\tilde{\underline{y}}_1, \dots, \tilde{\underline{y}}_N$ is the $\underline{x}$ satisfying

$$\min_{\underline{x} \in \Omega} \left[\sum_{n=1}^N (\tilde{\underline{y}}_n - \underline{y}_n)^T \mathbf{R}_n^{-1} (\tilde{\underline{y}}_n - \underline{y}_n) \right] \quad (3.21)$$

subject to system and measurement Eqs. (3.1) and (3.2). Notice that the summation in the bracket is identical to Eq. (3.3).

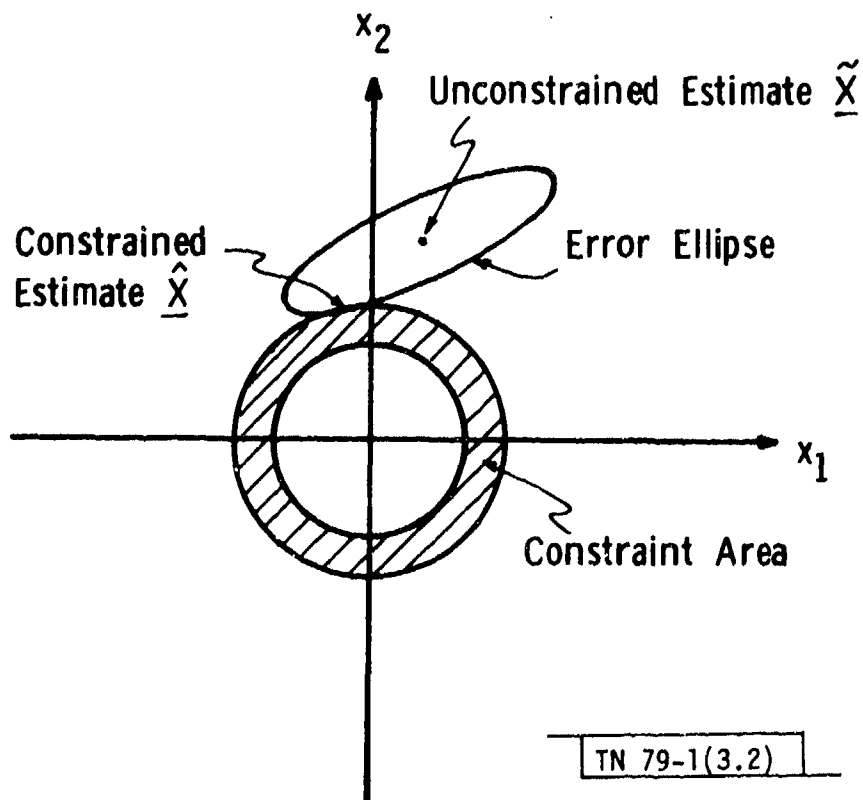
The above minimization can be decomposed into two steps:
 1) obtain the unconstrained estimate and covariance and 2) modify the unconstrained estimate using the constraint set. This decomposition procedure is optimal only for linear systems. We nevertheless adopt it here because the optimal solution for nonlinear systems is very difficult to obtain.

Let $\hat{\underline{x}}$ and P denote the state estimate and covariance obtained by the algorithm described in Sec. 3.1, the constrained estimate $\tilde{\underline{x}}$ is the solution of

$$\min_{\tilde{\underline{x}} \in \Omega} \left[(\tilde{\underline{x}} - \hat{\underline{x}})^T P^{-1} (\tilde{\underline{x}} - \hat{\underline{x}}) \right] \quad (3.22)$$

The above equation indicates that the optimal estimate is a state vector in Ω which has the shortest weighted distance to the unconstrained estimate $\hat{\underline{x}}$. If $\hat{\underline{x}}$ is in Ω , then $\tilde{\underline{x}} = \hat{\underline{x}}$. If $\hat{\underline{x}}$ is outside of Ω , then $\tilde{\underline{x}}$ is on the boundary of Ω . We illustrate a two-dimensional constrained estimation in Fig. 3.2. Notice that the optimal estimate $\tilde{\underline{x}}$ may not be the closest to $\hat{\underline{x}}$ measured by using the conventional distance. This is because that the distance implied in (3.22) is weighted by the covariance of $\hat{\underline{x}}$.

If $\hat{\underline{x}}$ is an unbiased estimate of $\underline{x}$ and is Gaussian, then $\tilde{\underline{x}}$ obtained in using (3.22) is the maximum likelihood estimate of $\underline{x}$ based upon both measurement and constraint set Ω . If a



TN 79-1(3.2)

Fig. 3.2. Constrained estimation.

distribution of $\underline{x}$ in Ω is known, one may also use the minimum mean square error estimator for computing $\hat{\underline{x}}$. Since the minimum mean square error estimation requires a distribution of $\underline{x}$ in Ω , and it is also more difficult to compute, we will not discuss it here. Discussions on the constrained estimation problem can be found in Ref. [6].

In the trajectory estimation application, a constraint set can be specified by bounds on velocity (v), heading angle (θ), and re-entry angle (γ)*. We denote the constraint set by

$$\Omega = \{ \underline{x}: v_1 \leq v \leq v_2; \lambda_1 \leq \gamma \leq \gamma_2; \theta_1 \leq \theta \leq \theta_2 \} \quad (3.23)$$

where these quantities are related to the coordinates in a east (x) - north (y) - up (z) oriented Cartesian coordinate by

$$\begin{aligned} v &= (\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{1/2} \\ \gamma &= \tan^{-1} \frac{\dot{x}}{\dot{y}} \quad (\gamma = 0 \text{ when heading north}) \\ \theta &= \sin^{-1} \frac{-\dot{z}}{v} \end{aligned} \quad (3.24)$$

As we discussed before, when $\tilde{\underline{x}}$ is outside of Ω , then the optimal solution lies on the boundary of Ω . Using (3.23), it is

*One may also include energy bounds Ω . This does not change the generality of the ensuing discussions.

evident that the segment of the boundary which contains $\hat{\underline{x}}$ is a subset of variables in (3.23) achieving equality. Such a subset can be found by searching through all possible combinations of variables achieving equalities. This procedure is tedious but straightforward. Let $\underline{f}(\underline{x}) = \underline{c}$ denote such a subset, then the minimization problem described in (3.22) becomes

$$\min[J = (\hat{\underline{x}} - \underline{x})^T P^{-1}(\hat{\underline{x}} - \underline{x}) + \underline{\lambda}^T (\underline{f}(\hat{\underline{x}}) - \underline{c})] \quad (3.25)$$

where $\underline{\lambda}$ is the vector of Lagrange multipliers. The necessary conditions for minimizing J are

$$-2P^{-1}(\hat{\underline{x}} - \underline{x}) + \frac{af^T(\hat{\underline{x}})}{a\hat{\underline{x}}} \underline{\lambda} = \underline{0} \quad (3.26)$$

$$\underline{f}(\hat{\underline{x}}) = \underline{c} \quad (3.27)$$

An algorithm solving (3.26) and (3.27) by including the constraint set described in (3.23) can be quite involved. Previous analytical studies [1], [2] indicate that constraints on the angles are very loose compared with the velocity constraints. Later

numerical results will also substantiate this fact. For this reason we derive an algorithm by only considering the velocity constraint in the next subsection.

3.3.2. Incorporating the Velocity Constraint

The velocity constraint can be rewritten as

$$\Omega_v = \{ \underline{x}: v_1^2 \leq \underline{x}^T S \underline{x} \leq v_2^2 \} \quad (3.28)$$

where $\underline{x}$ is the state vector in Cartesian coordinates with components $(x, y, z, \dot{x}, \dot{y}, \dot{z})$, S is a matrix defined as

$$S = \begin{bmatrix} 0^{3 \times 3} & \vdots & 0^{3 \times 3} \\ \hline 0^{3 \times 3} & \vdots & I^{3 \times 3} \end{bmatrix}$$

and I is an identity matrix. Assuming $\underline{\hat{x}}$ is not contained in Ω_v and

$$\underline{\hat{x}}^T S \underline{\hat{x}} = v_o^2$$

where v_0 may be equal to either v_1 or v_2 , then Eqs. (3.26) and (3.27) become

$$-P^{-1}(\underline{\hat{x}} - \underline{\hat{x}}) + \lambda S \underline{\hat{x}} = 0 \quad (3.26a)$$

$$\underline{\hat{x}}^T S \underline{\hat{x}} = v_0^2 \quad (3.27a)$$

Solving for $\underline{\hat{x}}$ in (3.26a) yields

$$\underline{\hat{x}} = (P^{-1} + \lambda S)^{-1} P^{-1} \underline{\hat{x}} \quad (3.29)$$

Substituting (3.29) in (3.27a) yields

$$\underline{\hat{x}}^T P^{-1} (P^{-1} + \lambda S)^{-1} S (P^{-1} + \lambda S)^{-1} P^{-1} \underline{\hat{x}} = v_0^2 \quad (3.30)$$

One must solve for λ using (3.30) then substitute it into (3.29) to obtain the final constrained estimate $\underline{\hat{x}}$. We use Newton's method to solve for λ in (3.30). Let

$$f(\lambda) = \tilde{\underline{x}}^T P^{-1} (P^{-1} + \lambda S)^{-1} S (P^{-1} + \lambda S)^{-1} P^{-1} \tilde{\underline{x}} - v_0^2 \quad (3.31)$$

and

$$f'(\lambda) = \frac{df(\lambda)}{d\lambda} \quad (3.32)$$

$$= -2\tilde{\underline{x}}^T P^{-1} (P^{-1} + \lambda S)^{-1} S (P^{-1} + \lambda S)^{-1} P^{-1} \tilde{\underline{x}}$$

The recursive solution for λ is

$$\lambda_{n+1} = \lambda_n - \frac{f(\lambda_n)}{f'(\lambda_n)} \quad (3.33)$$

Results of applying the above algorithm to the angle-only measurement trajectory estimation problem will be shown in the numerical results section.

We emphasize that although the constraint equation is only explicitly applied to the velocity components of the state, it is expected that both position and velocity estimates can be improved due to the fact that they are correlated through the covariance matrix P . This point can be made clear by examining (3.29). Notice that all components of $\tilde{\underline{x}}$ are changed by the constraint if $\lambda \neq 0$. Later numerical results will also substantiate this fact.

3.4. Overall Estimation Algorithms

We now combine the above analyses to present two angle-only tracking algorithms. One of which is strictly using the iterative least square estimator and the other one is using the

extended Kalman filter with the initial conditions provided by the ILS estimator. These two algorithms are illustrated in Figs. 3.3 and 3.4.

Notice that both algorithms use the initial guess calculation method presented in Section 3.2. After the initial guess has been obtained, the ILS algorithm proceeds to recursively apply the ILS estimator. Notice that the initial guess calculation method of Section 3.2 does not suggest ways of calculating the initial covariance matrix. In order to initiate the EKF algorithm, we first apply ILS to obtain both initial state and covariance then proceed to recursively apply the EKF estimator. This makes the ILS and EKF initial estimates identical. Later numerical results will show this effect.

Since there are a number of iterative processes in these algorithms, these following rather arbitrary check points are established to detect divergence:

- (1) For more than 15 iterations, $\Delta I = I_{k+1} - I_k$ is still larger than 10^{-8} where

$$I_k = \hat{x}^k T P^{k-1} \hat{x}^k$$

- (2) Estimated altitude is larger than 10^{10} km.
- (3) It takes more than 1200 seconds to impact

If the estimated state satisfies any one of the above check points, one additional data point is obtained and the initiation procedure is started over again. We emphasize that

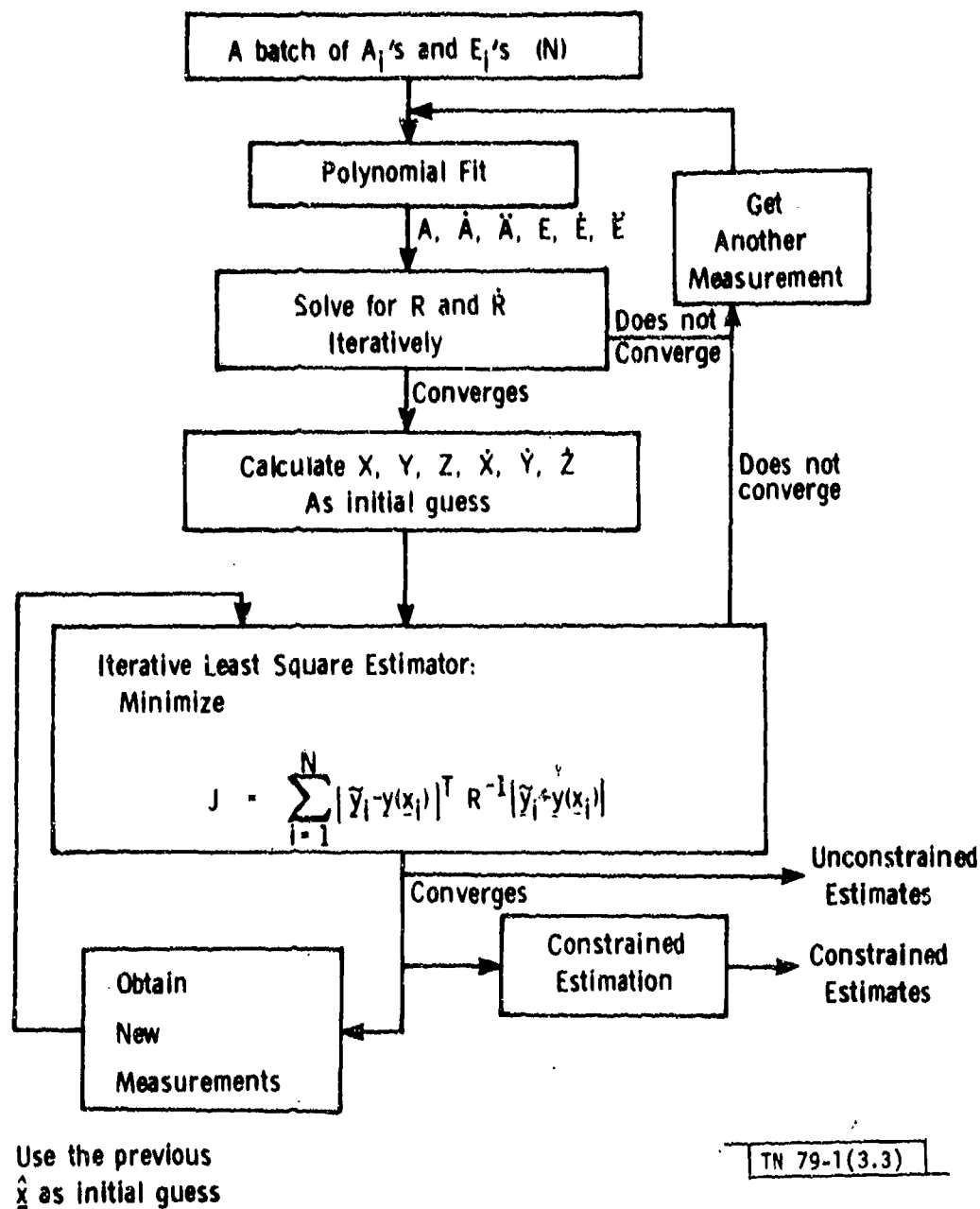


Fig. 3.3. Angle-only track algorithms: The iterative least square estimator.

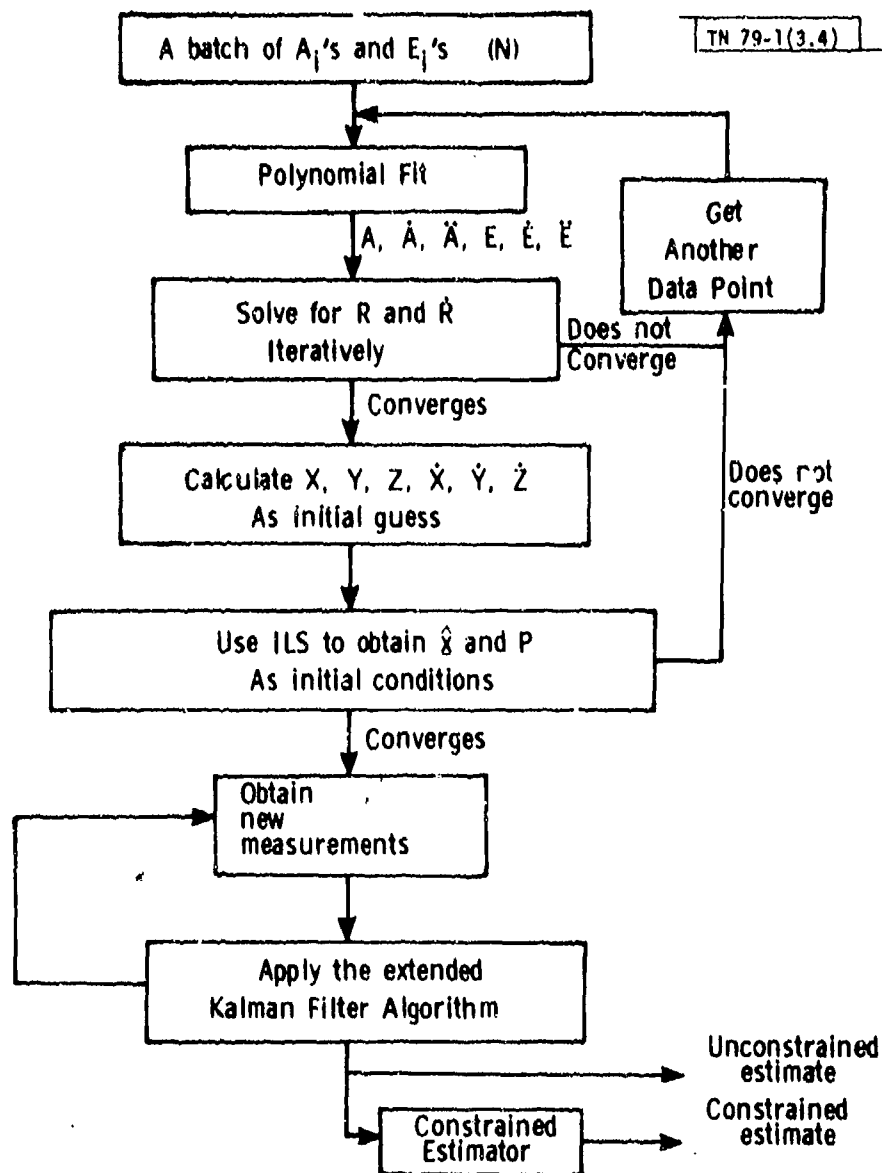


Fig. 3.4. Angle-only track algorithms: The extended Kalman Filter Algorithm.

we are using an expanding data window to include more data points in the initiation process as oppose to using a sliding window. In the numerical example section, we will use an example to illustrate the convergence frequency as a function of measurement standard deviation and number of pulses used in the initiation process.

Notice also that the constrained estimate is not used in the feedback loop. That is, it is applied only at the output of the overall algorithm.

4. NUMERICAL RESULTS

In this section, we present a numerical example to illustrate our results. Both the free-falling and stationary sensor platform cases are considered. The sensor and target initial state vectors in an arbitrarily oriented earth centered Cartesian coordinate are

$$\underline{x}_T = [2000., 3500., 7000., -1.5, -3., -2.]^T \quad \text{in km and km/s}$$

$$\underline{x}_S = [0., 4200., 6500., 0., 2., -1.]^T \quad \text{in km and km/s}$$

For a free-falling sensor platform, the sensor is following the same differential equation of motion (eg. (2.1)) with the above $\underline{x}_S$ as the initial state. For a stationary sensor platform. The sensor position remains constant at the above values. This could be achieved by thrusting the sensor platform to cancel gravity and velocity.

We assume that the sensor measurement standard deviation is 10 μ rad. Notice from Eq. (3.12), the estimation covariance is linearly related to measurement standard deviations. We can therefore scale our results using this fact. Higher measurement standard deviations will however, result in difficulties in convergence. This point will be illustrated later.

We apply velocity constraints at the filter output to improve our estimates. Two levels of velocity bounds are used to examine the sensitivity. They are ± 5 km/s and ± 1.25 km/s at the true velocity.

The simulation program is run in a Monte Carlo fashion to accumulate the error statistics. Root-mean-square (RMS) errors using 20 Monte Carlo runs are computed.

The RMS position errors of ILS and EKF estimates for the moving sensor platform case are presented in Figs. 4.1 and 4.2, respectively. Also shown is the unconstrained Cramer-Rao bounds, Ref. [1], [2]. It is evident that the ILS achieves the Cramer-Rao lower bounds while the EKF does not*. It is also interesting to note that constrained estimates have substantially smaller errors than the unconstrained estimates when the track time is short. When the track time is long, however, i.e., where there are sufficient data to provide velocity estimates with errors better than the bound, the advantage of constrained estimation is lost.

The RMS position errors of ILS and EKF estimates for the stationary sensor platform case are presented in Figs. 4.3 and 4.4 respectively. Notice that both ILS and EKF achieve the Cramer-Rao bound. The estimation error of a stationary sensor is much

*The fact that both EKF and ILS are below the C.R. Bound for short times is artificial and is due to assumptions equivalent to a velocity bound for initialization.

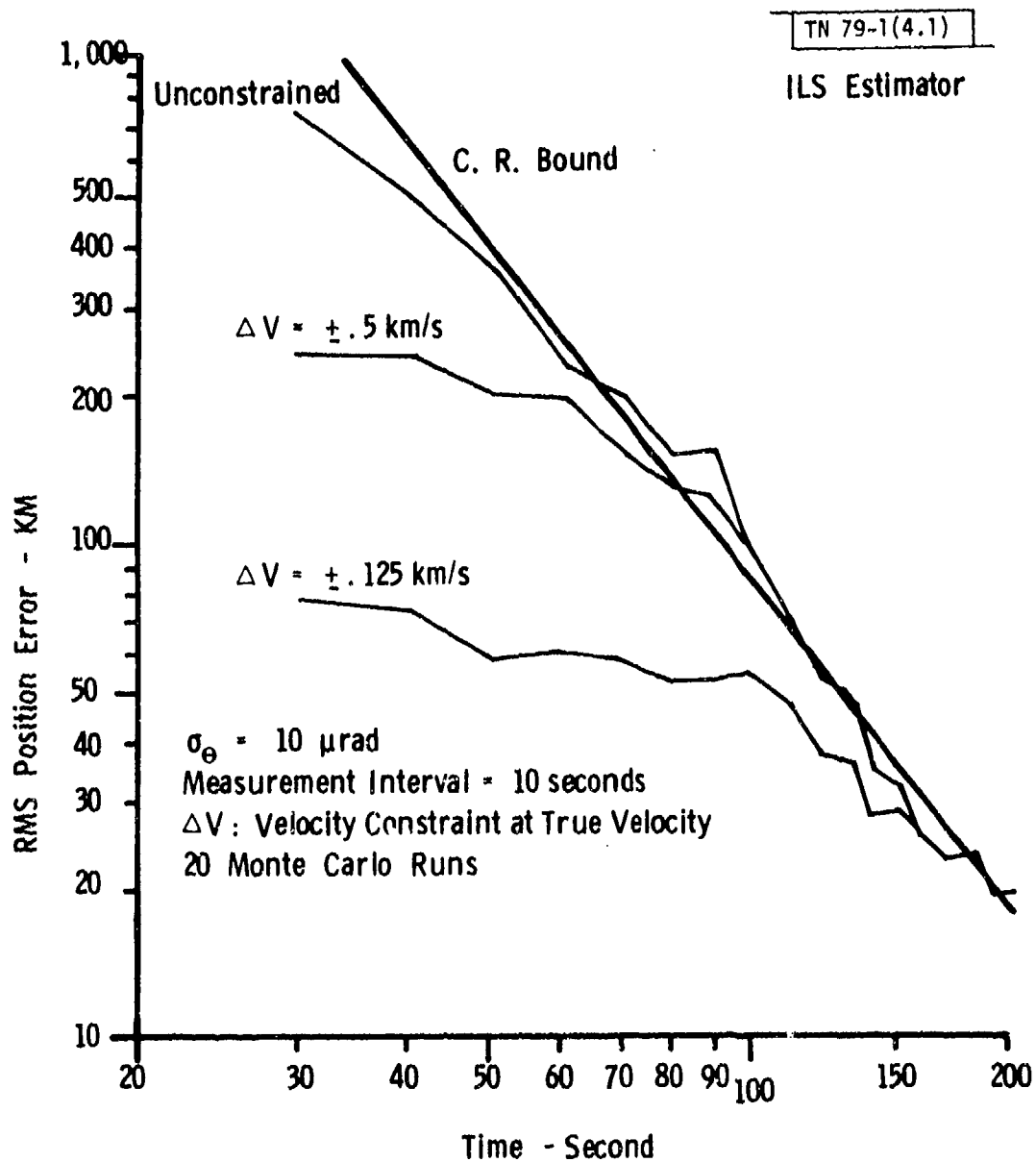


Fig. 4.1. Position estimation error: The ILS estimator with a free-falling sensor.

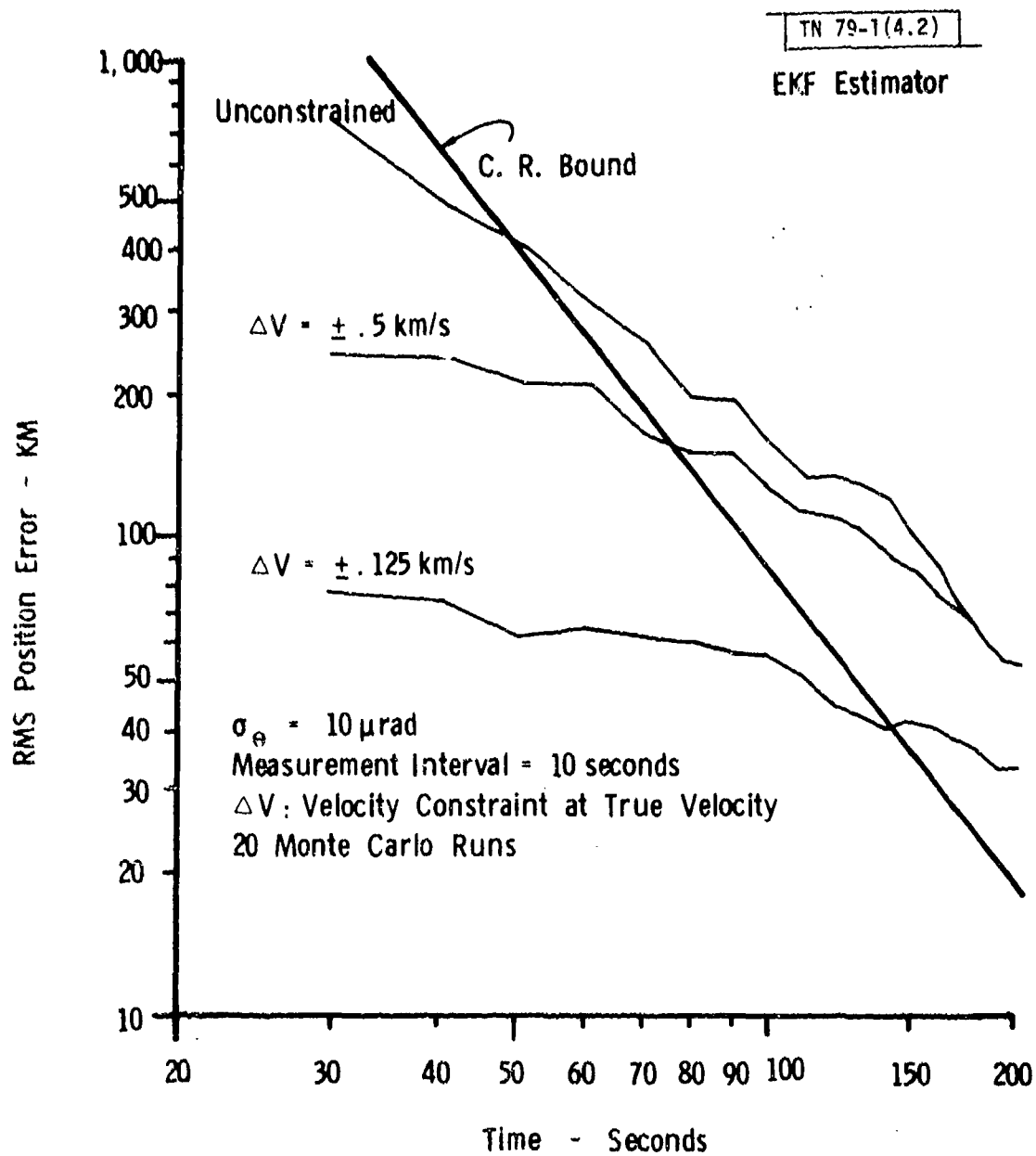


Fig. 4.2. Position estimation error: The EKF algorithm with a free-falling sensor.

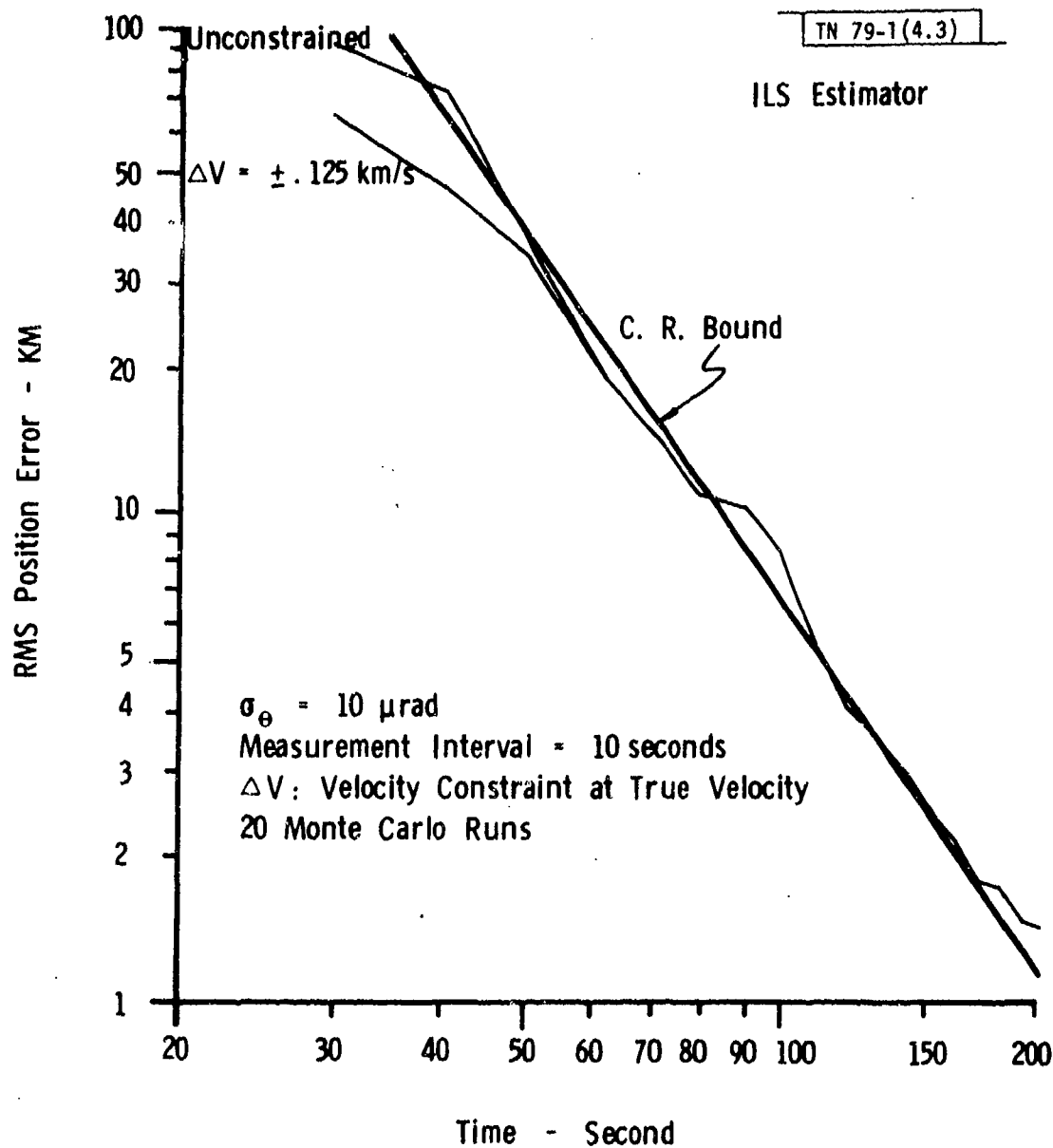


Fig. 4.3. Position estimation error: The ILS estimator with a stationary sensor (notice the vertical scale change with Figs. 4.1 and 4.2).

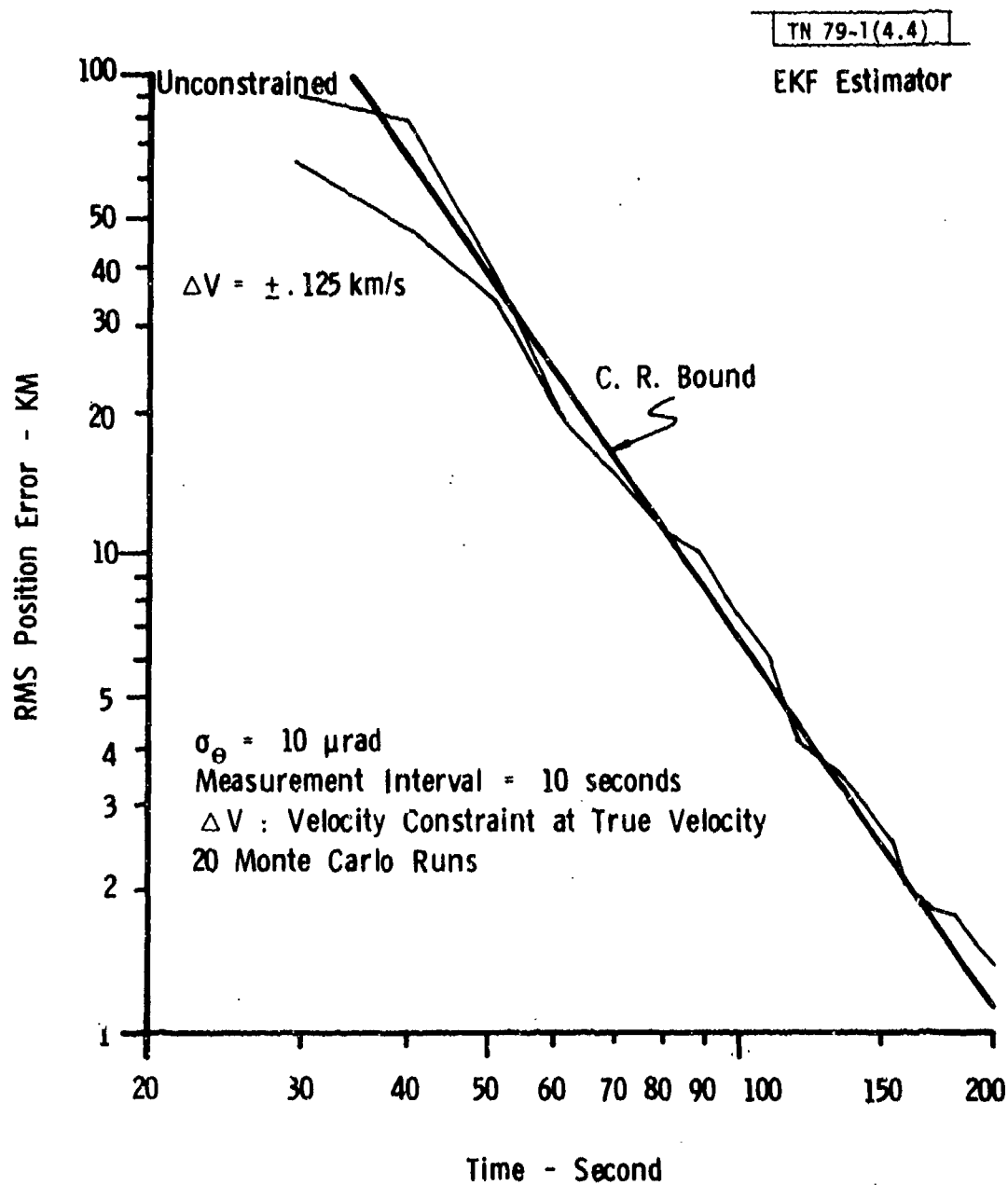


Fig. 4.4. Position estimation error: The EKF algorithm with a stationary sensor (notice the vertical scale change with Figs. 4.1 and 4.2).

smaller than that of a free-falling sensor (notice the change of vertical scale). A stationary sensor platform is apparently "enhancing" the system "observability" and this pushes the EKF much closer to the optimum (in the Cramer-Rao bound sense) performance. Furthermore, the velocity constraints seem to make very little improvement over the unconstrained estimates.

Figs. 4.5 and 4.6 present track initiation success frequencies for free-falling and stationary platforms, respectively. The success frequency for the stationary sensor platform case is much higher than that for the free-falling sensor platform case under the same conditions. Notice that the success frequency drops quickly as σ_θ increases. For a given σ_θ , the success frequency increases for increasing number of pulses used in the initiation process. The expanding window algorithm suggested in Section 3.4 therefore insures a near 100% success frequency for a given σ_θ .

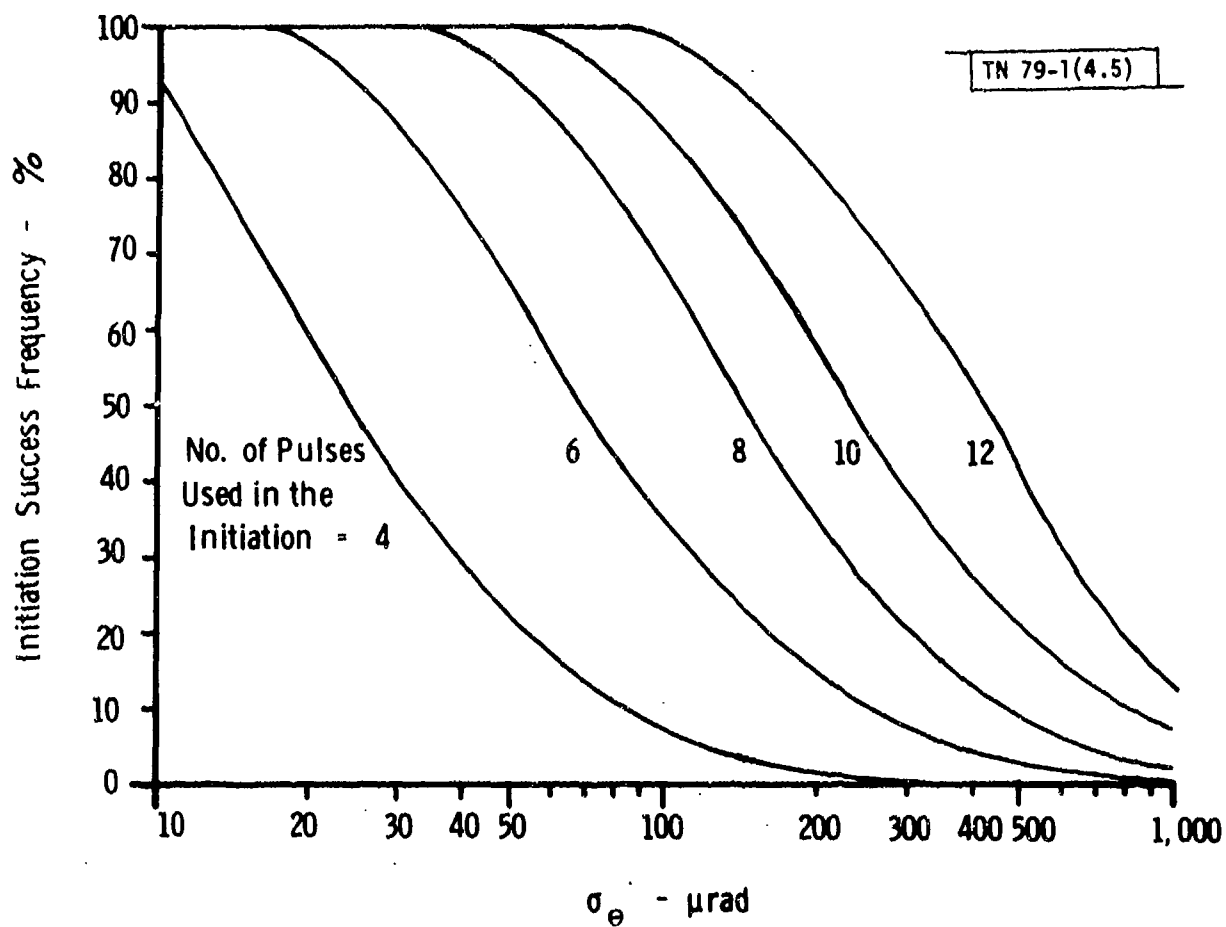


Fig. 4.5. Track initiation success frequency: Free-falling sensor platform.

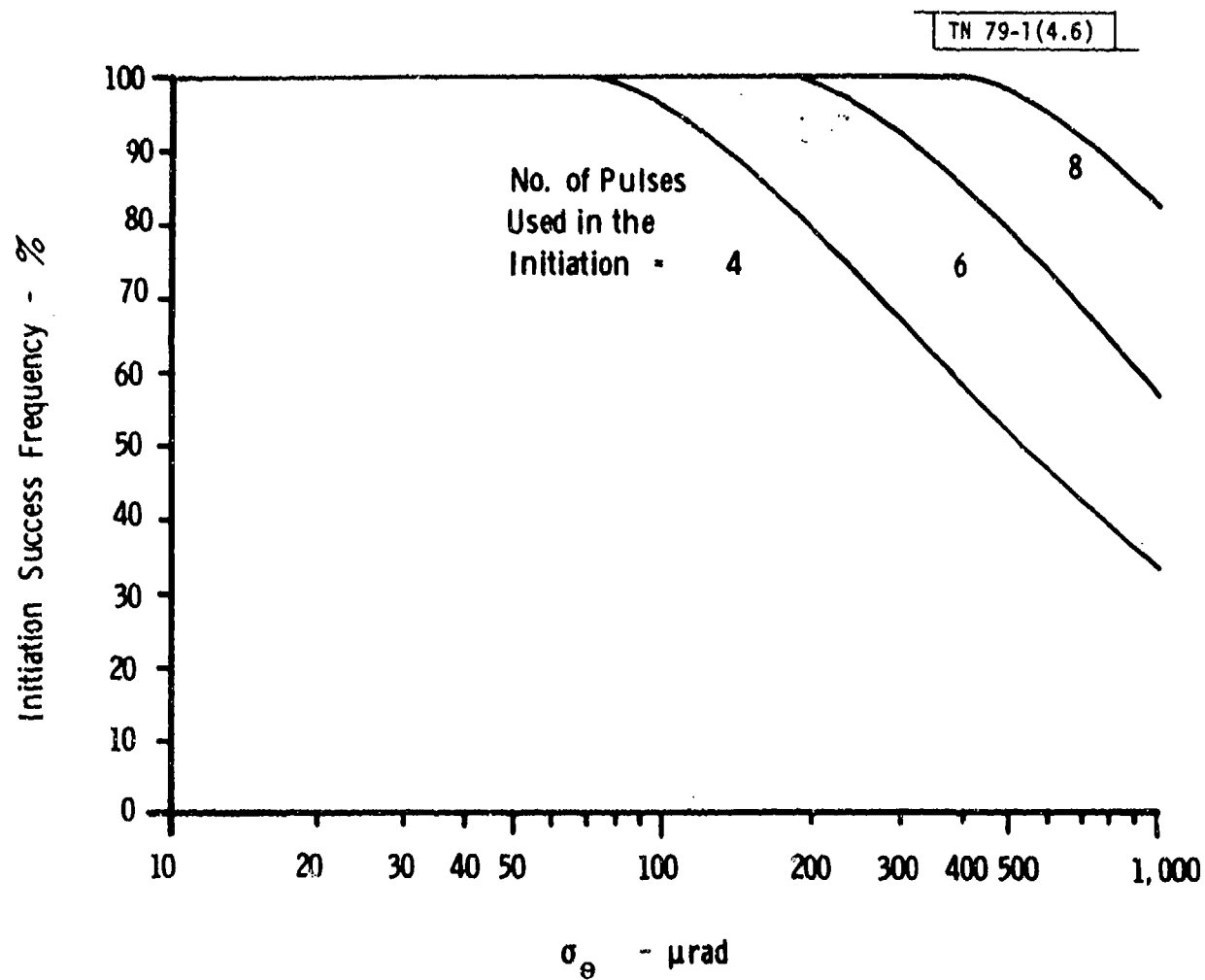


Fig. 4.6. Track initiation success frequency: Stationary sensor platform.

5. CONCLUSIONS

We have derived an iterative least square (ILS) estimation algorithm and applied it to the problem of ballistic trajectory estimation with angle-only measurements. We have also suggested an initiation scheme which is applicable to initiating either the ILS or the extended Kalman filter (EKF) algorithm. Methods for constrained estimation are presented to incorporate a priori trajectory constraints with state estimates.

There are still basic open issues in this area. For example, theories of the conditions for which the ILS will converge, the relationships among observability of nonlinear systems, the Cramer-Rao bound, and the iterative least square algorithm, among others, have not been explored.

It is found that a stationary sensor platform achieves substantially smaller estimation errors than the free-falling sensor. This suggests that there may exist an optimum control strategy for the sensor platform such that the estimation error is minimized. This is also an open area for further research.

In addition to the above discussions, we draw the following conclusions.

(1) For the free-falling sensor platform case, the ILS estimator achieves the Cramer-Rao lower bound on the covariance of trajectory estimates while the EKF does not.

(2) For the stationary sensor platform case, both ILS and EKF achieve the Cramer-Rao lower bound and the estimation error is much smaller than that for the free-falling sensor platform case.

(3) For sufficiently small measurement standard deviations, the suggested initiation procedure seems to have fine convergence properties.

(4) An extremely good velocity bound (i.e., the ± 125 km/s case) only gives marginal improvement over unconstrained estimates after tracking for 150 seconds, for the free-falling sensor platform case. For the stationary platform case, the velocity constraints make very little effect over the unconstrained estimates.

ACKNOWLEDGMENTS

The author would like to express his sincere appreciation of programming support by Jeff Tolleson. Jeff is not only an intelligent programmer, but also a careful mathematician who has contributed much to the successful implementation of these algorithms. In addition, the procedure of initial guess calculation using average missile energy was derived by him. The author would also like to thank Dr. R. W. Miller for many stimulating discussions and to Dr. K. P. Dunn, who has offered help in many respects including technical discussions and supervising simulation efforts.

REFERENCES

1. R. W. Miller and C. B. Chang, "Some Analytical Methods for Tracking, Prediction, and Interception Performance Evaluations," Project Report RMP-129, Lincoln Laboratory M.I.T. (9 August 1977).
2. R. W. Miller, "A Lower Bound on Angle-only Tracking Accuracy," Project Report RMP-149, Lincoln Laboratory, M.I.T. (9 May 1978).
3. C. B. Chang and J. W. Sherman, "The combination of a Statistical Measurement with Deterministic A Priori Constraint," Proceedings of Fourth Symposium on Non-linear Estimation Theory and Its Applications, San Diego, CA (September 1973).
4. A. A. Goldstein, Constructive Real Analysis, (Harper & Row, New York, 1967).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER (18) ESD-TR-79-5	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) (6) Optimal State Estimation of Ballistic Trajectories with Angle-Only Measurements		5. TYPE OF REPORT & PERIOD COVERED (9) Technical Note (14) 77N-	
6. AUTHOR(s) (10) Chaw-Bing Chang		7. PERFORMING ORG. REPORT NUMBER Technical Note 1979-1	
8. CONTRACT OR GRANT NUMBER(s) (15) F19628-78-C-0002		9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) Project No. 8X36334D215	
10. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173		11. REPORT DATE (11) 24 January 1979	
12. CONTROLLING OFFICE NAME AND ADDRESS Ballistic Missile Defense Program Office Department of the Army 5001 Eisenhower Avenue Alexandria, VA 22333		13. NUMBER OF PAGES 53	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB Bedford, MA 01731 (12) 50p.		15. SECURITY CLASS. (of this report) Unclassified	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES None			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) iterative least square filter state estimation ballistic trajectories stationary and free falling sensors angle-only measurements Monte Carlo Cramer-Rao bound constrained estimation			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) An iterative least square estimation algorithm is derived and applied to the problem of state estimation of ballistic trajectories with angle-only measurements. A filter initiation procedure is suggested. The application of trajectory a priori knowledge for improving the estimate is discussed and solved as a constrained estimation problem. A Monte Carlo simulation study was conducted to evaluate these techniques. It was found that the iterative least square filter achieves the Cramer-Rao bound and it performs better than the extended Kalman filter when the sensor is on a free-falling platform. When the sensor is on a stationary platform however, both estimators asymptotically achieve the Cramer-Rao bound.			

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

207650

16