Trajectory and Launch Point Estimation for Ballistic Missiles from Boost Phase LOS Measurements *

Yicong Li†
Comverse Network Systems
100 Quannapowitt Parkway
Wakefield MA 01880, USA.

Thiagalingam Kirubarajan‡ and Yaakov Bar-Shalom§
Electrical and Systems Engineering Department
University of Connecticut
Storrs CT 06269-2157, USA.

Murali Yeddanapudi¶
The MathWorks, Inc.
24 Prime Park Way
Natick MA 01760-1500, USA.

Abstract

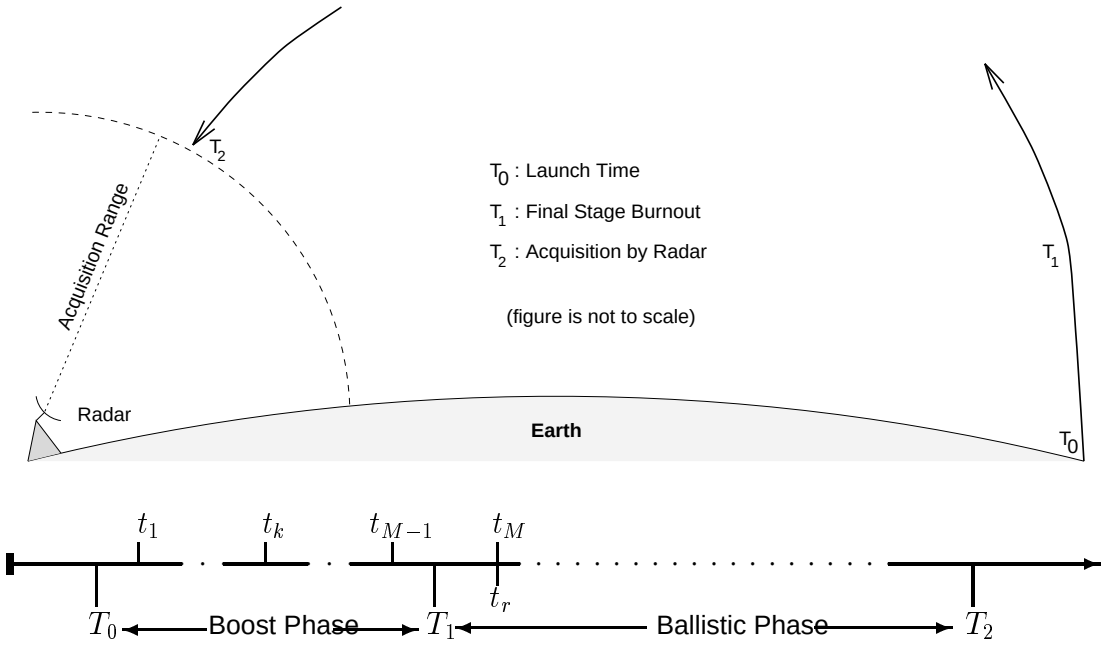
This paper addresses the problem of estimating the trajectory and the launch point of a tactical ballistic missile using line of sight (LOS) measurements from one or more passive sensors (typically satellite-borne). The major difficulties of this problem include the ill-conditioning of the estimation problem due to poor observability of the target motion via LOS measurements, the estimation of the unknown launch time, and the incorporation of inaccurate target thrust profiles to model the target dynamics during the boost phase. We present a maximum likelihood (ML) estimator based on the Levenberg-Marquardt algorithm that provides both the target state estimate and the associated error covariance, taking into consideration the complications mentioned above. One important consideration in the defense against tactical ballistic missiles (TBM) is the determination of the target position and error covariance at the acquisition range of a surveillance radar located in the vicinity of the impact point. We present a systematic procedure to propagate the target state and covariance to a nominal time, when it is within the detection range of a surveillance radar to obtain a cueing region. We also provide an estimate and the error covariance of the (two dimensional) launch position, which can be used to search for the missile launch site. Monte Carlo simulation studies on typical single and multiple sensor scenarios indicate that the proposed algorithms are accurate in terms of the estimates and that the estimator calculated covariances are consistent with the errors.

1 Introduction

The flight of a ballistic missile consists of three phases: boost phase, free flight (mid-course or ballistic phase) and re-entry phase. In this paper we present the trajectory estimation of a tactical ballistic missile (TBM) for the first two phases based on measurements from only the boost stage. During the boost phase the plume generated by a ballistic missile makes it quite visible to most IR sensors mounted on strategically located surveillance satellites. However, after burnout (i.e., after the boost phase) the target

*Research supported by ONR/BMDO-IST Grant N00014-91-J-1950 and AFOSR Grant F49620-97-1-0198.
†Email: *yli@comversens.com.* Research performed while at the University of Connecticut.
‡Email: *kiruba@ee.uconn.edu.*
§Email: *ybs@ee.uconn.edu.*
¶Email: *murali@mathworks.com.* Research performed while at the University of Connecticut.

is very difficult to detect because it no longer burns fuel, i.e., it is very dim in the IR spectrum. In the second stage, the movement of the missile is governed by the two-body movement rules, namely, Kepler's law, which will be used to propagate the state of the missile in the second stage. The time from launch to impact is 5-10 minutes and the missile is visible to the IR sensors on the surveillance satellite only in the first couple of minutes after launch. In the detection and tracking of TBM which have a short range (less than 600 miles), time is a critical factor. Once a missile launch is detected by surveillance satellites, surface-based radar in the vicinity of potential impact points will be alerted to search for the incoming missile. However, this leaves only a very short time (typically a few minutes) to detect, intercept and destroy the missile. The radar search strategy can be vastly improved if one can predict a limited search region for the missile based on the satellite observations. By using this information in the radar acquisition of the missile prior to reentry, the search region can be reduced from the entire radar field of view to the region obtained using the position estimate and the error covariance. In TBM defense, one also wants to locate the launch site of the attacking missile to take action against it. There is only a very short time to search for the launching facilities, which usually move away quickly. This requires an accurate estimate and the error covariance of the launch position. Thus, the estimation of the missile trajectory and its associated error covariance in the boost phase plays an important role in tactical ballistic missile defense.



Note: $t_1, \ldots, t_M$ are the measurement time instants and t_r is the reference time.

Figure 1: Propagation to the the nominal time of acquisition by a surveillance radar.

The models used to describe the motion of a missile during boost can be broadly classified as being either profile-free or profile-based models. In profile-free models the state vector is estimated using a polynomial motion model, i.e., no thrust information is assumed to be available. In profile-based models it is assumed that the magnitude and orientation of the missile thrust vector, thus the acceleration, is known. The profile-free model uses simple state equations for the motion of the TBM, which allows sequential filtering of the missile states (usually the position, velocity, and acceleration in the three dimensional space). For example, (Moré, *et. al.*, 1996) assumes several synchronized satellites and a linear motion. While this approach is computationally inexpensive, it does not accurately model the entire

boost trajectory, especially in the case of multistage missiles, whose thrust changes abruptly. Therefore, accurate estimates cannot be obtained. Hence, in the case of multistage missiles (which is the most common case), profile-based models should be used to accurately describe the missile motion during the boost phase to obtain satisfactory performance in terms of estimation accuracy. The main focus of this paper is the development of a Maximum Likelihood (ML) estimation algorithm with a robust numerical implementation for estimating both the acquisition point and the launch point, incorporating the profile-based modeling of the missile boost phase. In addition, the effects of the partial profile information on the estimation errors are also taken into consideration in evaluating the state estimate and the associated error covariance. The obscuration effects of cloud cover are also investigated. Profile-based models were also mentioned in (Rudd, *et. al.*, 1994) but only as one out of a fixed known family. Our approach includes profile parameter estimation together with missile position/velocity and launch time.

The measurements originate from sensors which are strategically located so as to detect the target within a short period of time after launch. These sensors can be located on airborne early warning systems, or low-altitude earth satellites. The sensors could be either passive, which can measure only the line of sight (LOS) direction, or active, which also measure the distance to the target. In this paper we focus on LOS measurements made from passive sensors located on low-altitude earth satellites. A LOS is a unit vector represented by two angles: azimuth and elevation of the target as detected by the sensor. Assuming that the target state consists of only its position and velocity, we need at least three LOS measurements (at different times) to estimate the six dimensional state of the target. In addition to the problems associated with the profile-based modeling of the boost phase trajectory, another important consideration is the observability of the target through the LOS measurements (Bar-Shalom, *et. al.*, 1993). In general, the sensors are so far away from the target that a significant change in the target position (over a period of time) is reflected only as a slight variation in the LOS measurements. This results in poor observability of the target motion. Consequently, the Maximum Likelihood (ML) trajectory estimation problem is very ill-conditioned for most realistic target-sensor geometries.

The problem is formulated in Section 2. Section 3 deals with the ML trajectory estimation, including the estimation of the launch time and major parameters of the thrust profile. The case of non-unity detection probability is also considered. Results of simulations performed for some typical scenarios are presented in Section 4.

2 Problem Formulation

The main focus of this paper is the estimation of the trajectory of a missile using LOS measurements from satellite-based sensors during the missile boost phase. To be specific, we are interested in the estimation of the state of the missile at two time points. The first is a nominal time at which the missile is within the acquisition range of a surface-based radar located in the vicinity of the impact point. Since we have the LOS measurements only during the boost phase of the missile, we estimate the missile state and covariance close to the final stage burnout time, and propagate this state to obtain the estimate and the associated error covariance of the missile position at the time of acquisition by the radar. As shown in Fig. 1, the boost phase of the missile extends from the launch time T_0 to the final stage burnout time T_1. The missile is in the ballistic phase from time T_1 until reentry (assumed to take place after the nominal time T_2 of acquisition by the radar). During this portion of the trajectory both the thrust and the drag are zero, and the motion of the missile is governed by Kepler's laws. Also shown in Fig. 1 are the measurement times $t_1, t_2, \ldots, t_m$ and the reference time t_r, at which we wish to estimate the missile state. As Fig. 1 shows, the first measurement is made shortly[1] after launch and the last measurement

[1] In the following sections the situation of missing the first several detections is also discussed.

occurs shortly before (or after[2]) the final stage burn-out. We choose the last measurement time t_M as the reference time t_r, when we have the measurement closest to the acquisition range of the ground radar, so that the effect of the inaccuracy of the model on the estimation of that point can be minimized.

The second point of interest is the missile launch point whose position estimate and the associated covariance can provide important information on the launching facilities. In this case we set the launch time T_0 as the reference time t_r and find the launch position estimate and its covariance (this reference time itself also needs to be estimated).

In the following subsections we present the target and measurement models used in this paper.

2.1 Target Motion Model

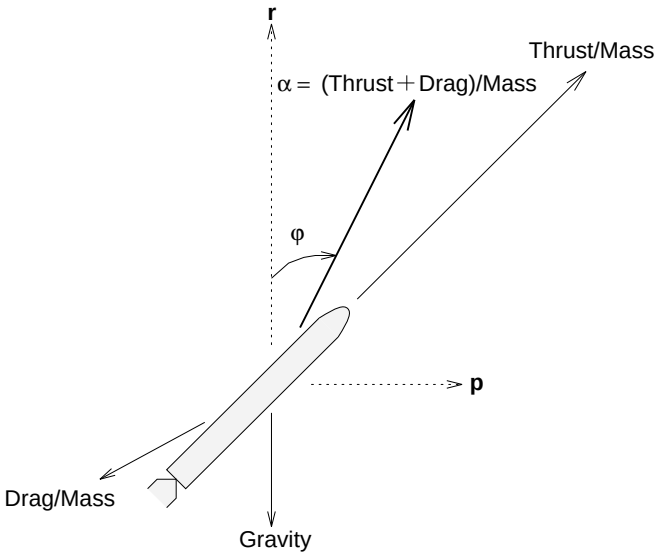


Figure 2: Components of the missile acceleration during the boost phase.

The main forces on the target during the boost phase are shown in Fig. 2. The thrust goes to zero at the end of the boost phase and the drag on the target goes to zero once the target goes out of the earth's atmosphere[3] In modeling the target motion, the thrust and drag can be combined into "the net specific thrust" (i.e., acceleration given by the thrust and drag per unit mass). We shall denote the magnitude of the net specific acceleration[4] by $\alpha(t)$ and the angle between the net thrust and the outward normal vector from the center of the earth-centered earth-fixed (ECEF) frame of reference by $\varphi(t)$ (pitch-over angle — the orientation of the net thrust vector). Both $\alpha(t)$ and $\varphi(t)$, as defined above, are indicated in Fig 2.

Let $\mathbf{r} = [r_x\ r_y\ r_z]'$ and $\dot{\mathbf{r}} = [\dot{r}_x\ \dot{r}_y\ \dot{r}_z]'$, denote the position and velocity, respectively, of the target in the ECEF frame of reference. The dynamic equation describing the target motion can be written as:

$$\ddot{\mathbf{r}} = \left(\alpha\cos\varphi - \frac{\mu}{\|\mathbf{r}\|^2}\right)\ \frac{\mathbf{r}}{\|\mathbf{r}\|} + (\alpha\sin\varphi)\ \mathbf{p} \tag{1}$$

[2]An IR sensor can still detect the missile very shortly after the boost-out before the engine of the missile cools down.

[3]In this paper they are taken as the same time because after the boost phase of the missile the atmosphere is thin enough to be ignored, compared to the other forces.

[4]This acceleration is the vector sum of the thrust plus drag devided by the mass.

where, $\|\cdot\|$ denotes the Euclidean norm, μ is the Earth's gravitational constant[5] and

$$\mathbf{p} = \frac{\dot{\mathbf{r}} - (\dot{\mathbf{r}} \cdot \mathbf{r}/\|\mathbf{r}\|)\ (\mathbf{r}/\|\mathbf{r}\|)}{\|\dot{\mathbf{r}} - (\dot{\mathbf{r}} \cdot \mathbf{r}/\|\mathbf{r}\|)\ (\mathbf{r}/\|\mathbf{r}\|)\|} \tag{2}$$

is a unit vector orthogonal to $\mathbf{r}$, such that the two vectors $\mathbf{r}$ and $\mathbf{p}$ lie the missile's orbital plane[6].

As a note on the coordinate system, the estimation at the reference time in the boost phase is done in the ECEF (i.e., rotating) frame of reference, rather than local coordinates, to evaluate the trajectory propagation over the earth curvature. The free flight under Kepler's law is inertial by nature and the propagation is done in ECI coordinates (earth centered inertial). The earth rotation under the flying object will not influence the estimation accuracy at the radar acquisition time (which yields the radar cueing region, our main interest). The rotation of the earth during the time when the missile undergoes free flight should be used as an adjustment of the nominal radar acquisition point.

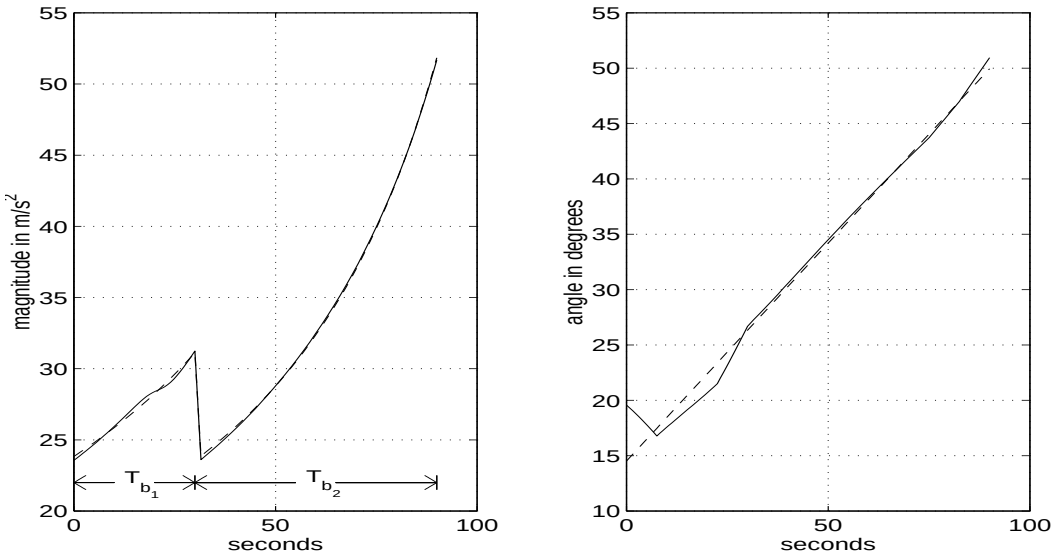


Figure 3: Magnitude $\alpha(t)$ and orientation $\varphi(t)$ of the net thrust on the target during boost phase (solid line: actual; dashed line: model).

Typical curves (Blair, *et. al.*, 1997) for $\alpha(t)$ and $\varphi(t)$ are shown in Fig. 3. We shall define the thrust profile $\mathcal{P}$ as the set consisting of the curves $\alpha(t)$ and $\varphi(t)$, i.e.,

$$\mathcal{P} \stackrel{\triangle}{=} \{\alpha(t),\ \varphi(t),\ \forall\ T_0 \le t \le T_1\} \tag{3}$$

Also shown in the above figures, the dashed lines are approximate models for $\alpha(t)$ and $\varphi(t)$. Using these models (a piecewise hyperbolic model for the magnitude $\alpha(t)$, discussed in detail later, and a linear model for $\varphi(t)$, the orientation angle) in place of the actual curves allows us to use a relatively simple propagation of the target state in calculating the cost function shown later in (12). The acceleration magnitude model is chosen under the assumption that the missile has an approximately constant mass flow rate in each stage. Of all the parameters determining the profile $\mathcal{P}$, the burn-out times T_{b_1} and T_{b_2} shown in Fig. 4, are the most critical ones, and will be estimated in the next section.

[5] $\mu = 398601.2\ \mathrm{km}^3/\mathrm{s}^2$.

[6] The trajectory of the missile is in this plane, which is conventionally defined using $\mathbf{r}$ and $\dot{\mathbf{r}}$. The pair $\{\mathbf{r}/\|\mathbf{r}\|,\ \mathbf{p}\}$ is an orthonormal base to this orbital plane.

Since (1) is a second order nonlinear differential equation in $\mathbf{r}$, and under the assumption that the thrust profile $\mathcal{P}$ is given, the missile state x consists of only the position and velocity components, i.e., $x \triangleq [\mathbf{r}'\ \dot{\mathbf{r}}']'$. Hence, given the thrust profile $\mathcal{P}$, the launch time T_0, the state of the missile $x(t)$ at any time $t \in [T_0, T_2]$ can be determined by propagating deterministically (via numerical integration) the state x_r at some reference time t_r. We can formally write this as

$$x(t) = f(\mathcal{P}, t, t_r, x_r, T_0) \quad \forall\, t \in (T_0, T_2] \tag{4}$$

Consequently, a parameter estimation problem will be solved, for which the Cramer-Rao Lower Bound (CRLB) can be used as the covariance matrix of the corresponding estimation error (Bar-Shalom, *et. al.*, 1995).

2.2 Measurement Model

The measurements consist of LOS from N passive sensors: $S_1, \ldots, S_N$. The measurements originating from the N sensors are assumed to be ordered based on their "time stamps" and the sensors are (realistically) modeled as asynchronous. We shall use $\zeta(k)$ to indicate that measurement k is from sensor $S_{\zeta(k)}$. Note that $\zeta(k) \in \{1, \ldots, N\}\ \ \forall\, k$. The measurement equation is

$$z_k = h_k[x_{\text{true}}(t_k)] + w_k \quad k = 1, 2, \ldots, M \tag{5}$$

where $x_{\text{true}}(\cdot) = [\mathbf{r}_{\text{true}}(\cdot)'\ \dot{\mathbf{r}}_{\text{true}}(\cdot)']'$ is the *true state* of the missile, M is the number of measurements, and w_k is the measurement noise which is modeled as a zero-mean[7], white, Gaussian[8] sequence with known covariance R_k, i.e.,

$$w_k \sim \mathcal{N}(0, R_k) \quad k = 1, 2, \ldots, M \tag{6}$$

The LOS $h_k(\cdot)$ consists of the azimuth $\phi_k(\cdot)$ and the elevation $\theta_k(\cdot)$ of the target as seen from sensor $S_{\zeta(k)}$, i.e.,

$$h_k[x(t_k)] = \begin{bmatrix} \phi_k[x(t_k)] \\ \theta_k[x(t_k)] \end{bmatrix} = \begin{bmatrix} \arctan\left\{ \dfrac{r_y(t_k) - s_{\zeta(k),y}(t_k)}{r_x(t_k) - s_{\zeta(k),x}(t_k)} \right\} \\ \arcsin\left\{ \dfrac{r_z(t_k) - s_{\zeta(k),z}(t_k)}{\left\| \mathbf{s}_{\zeta(k)}(t_k) - \mathbf{r}(t_k) \right\|} \right\} \end{bmatrix} \tag{7}$$

where the vector $\mathbf{s}_{\zeta(k)}(t_k) = \left[s_{\zeta(k),x}(t_k)\ \ s_{\zeta(k),y}(t_k)\ \ s_{\zeta(k),z}(t_k) \right]'$ is the position of sensor $S_{\zeta(k)}$ at time t_k, and is assumed to be known and the second subscript in the above equation denotes the coordinate.

3 Maximum Likelihood Estimation

In this section we first address the specific aspects of the trajectory estimation problem at a general reference time, including the estimation of the unknown launch time and key parameters of the thrust profile with the consideration of missing the first several detections. Then, we will set the reference time to either the last measurement time (and propagate to the nominal time for the acquisition range of the ground radar) or the (to be estimated) launch time, and estimate the missile state at that time.

[7] The problem of sensor biases is beyond the scope of this study.

[8] The Gaussian assumption is a reasonably realistic one and on the conservative side (Bar-Shalom, *et. al.*, 1995) as long as the variances are correct.

Let Z_M denote the set of measurements, i.e., $Z_M \triangleq \{z(k),\ k = 1, \ldots, M\}$. Since the measurements in this set are mutually independent, we can write the conditional probability density of Z_M as the the following product:

$$p\left(Z_M | x_r,\, T_0,\, \mathcal{P}\right) = \prod_{k=1}^{M} p\left(z(k) | x_r,\, T_0,\, \mathcal{P}\right) \tag{8}$$

This is the likelihood function of the parameters to be estimated.

The measurement errors are modeled as Gaussian with known covariance, hence, the conditional density of $z(k)$ can be written as

$$p\left(z(k) | x_r,\, T_0,\, \mathcal{P}\right) = \left|2\pi R_k\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left\|D_k^{-1}\left[\hat{z}_k\left(x_r,\, T_0,\, \mathcal{P}\right) - z(k)\right]\right\|^2\right\} \tag{9}$$

where $\hat{z}_k\left(x_r,\, T_0,\, \mathcal{P}\right) \triangleq h_k\left[\mathbf{f}\left(x_r,\, T_0,\, \mathcal{P}\right)\right]$ is the estimated measurement and D_k is the square root[9] of R_k; $\hat{z}_k\left(x_r,\, T_0,\, \mathcal{P}\right)$ is evaluated by numerical integration using the information of either the solid lines or the dotted lines in Fig. 4. Substituting (9) in (8), and simplifying we get

$$p\left(Z_M | x_r,\, T_0,\, \mathcal{P}\right) = \frac{1}{c}\exp\left\{-\frac{1}{2}\sum_{k=1}^{M}\left\|D_k^{-1}\left[\hat{z}_k\left(x_r,\, T_0,\, \mathcal{P}\right) - z(k)\right]\right\|^2\right\} \tag{10}$$

where c is a normalization constant, which is not relevant for the present discussion. Let $\nu\left(x_r,\, T_0,\, \mathcal{P}\right)$ be the vector of normalized measurement residuals, i.e.,

$$\nu\left(x_r,\, T_0,\, \mathcal{P}\right) \triangleq \begin{bmatrix} D_1^{-1}\left[\hat{z}_1\left(x_r,\, T_0,\, \mathcal{P}\right) - z(1)\right] \\ \vdots \\ D_M^{-1}\left[\hat{z}_M\left(x_r,\, T_0,\, \mathcal{P}\right) - z(M)\right] \end{bmatrix} \tag{11}$$

We take the square of the Euclidean norm of the vector $\nu\left(x_r,\, T_0,\, \mathcal{P}\right)$ as the cost function $J\left(x_r,\, T_0,\, \mathcal{P}\right)$, i.e.,

$$J\left(x_r,\, T_0,\, \mathcal{P}\right) \triangleq \left\|\nu\left(x_r,\, T_0,\, \mathcal{P}\right)\right\|^2 = \sum_{k=1}^{M}\left\|D_k^{-1}\left[\hat{z}_k\left(x_r,\, T_0,\, \mathcal{P}\right) - z(k)\right]\right\|^2 \tag{12}$$

Substituting (12) in (10) we can write

$$p\left(Z_M | x_r,\, T_0,\, \mathcal{P}\right) = \frac{1}{c}\exp\left\{-\frac{1}{2}J\left(x_r,\, T_0,\, \mathcal{P}\right)\right\} \tag{13}$$

Observe that $J\left(x_r,\, T_0,\, \mathcal{P}\right)$ is proportional to the negative log likelihood of $p\left(Z_M | x_r,\, T_0,\, \mathcal{P}\right)$. Hence, maximizing the likelihood function is equivalent to minimizing the cost function

$$\arg\max_{\xi}\ p\left(Z_M | x_r,\, T_0,\, \mathcal{P}\right) \equiv \arg\min_{\xi}\ J\left(x_r,\, T_0,\, \mathcal{P}\right) \tag{14}$$

where the arguments ξ of the minimization are given below.

With the above results at hand, we now state the trajectory estimation problems in the following four cases, in order of increasing complexity. The information available and the parameters to be estimated in each case are shown in Table 1.

Case (a). We shall first consider the simplest case when only the target state at a reference time is to be estimated. In this case, it is assumed that accurate values of both the launch time and the thrust profile are available and are, hence, not estimated. Let T_0^{true} be the launch time and $\mathcal{P}^{\text{true}}$ (the solid line in Fig. 3)

[9] Since these two angle measurements are uncorrelated, D is just the square root of the diagonal matrix R, i.e., $D_{ii} = \sqrt{R_{ii}}$.

	Launch Time	Profile	Measurements
Case (a)	true	true	all
Case (b)	to be estimated	true	all
Case (c)	to be estimated	to be estimated	all
Case (d)	to be estimated	to be estimated	$\geq j$

Table 1: Summary of information available and parameters to be estimated in four cases.

be the thrust profile, both of which are fixed. It only remains to estimate the state x_r of the target at a reference time t_r to completely determine the target trajectory. The maximum likelihood estimate $\hat{x}_r^{(a)}$ of the target state would then be given by:

$$\hat{x}_r^{(a)} = \arg\min_{x_r} J\left(x_r,\, T_0^{\text{true}},\, \mathcal{P}^{\text{true}}\right) \tag{15}$$

Case (b). Next, we relax the assumption of known launch time and estimate it along with the target state. In this case, we still assume that the thrust profile is fixed at the true value $\mathcal{P}^{\text{true}}$. The maximum likelihood estimates of the launch time $\hat{T}_0^{(b)}$ and target state $\hat{x}_r^{(b)}$ are obtained via the global minimization of the cost function over both x_r and T_0, i.e.,

$$\left\{\hat{x}_r^{(b)},\, \hat{T}_0^{(b)}\right\} = \arg\min_{x_r,\, T_0} J\left(x_r,\, T_0,\, \mathcal{P}^{\text{true}}\right) \tag{16}$$

Case (c). Now we address the problem of trajectory estimation in the absence of an accurate thrust profile. Theoretically we can estimate all the parameters in the constant mass flow rate model profile. However, during the short period of boost phase, we do not have enough information (number of measurements) to obtain a good estimate, especially in this LOS measurements based ill-conditioned problem. Out of the parameters defining the thrust profile (the dotted lines in Fig. 3), the stage burn-out times T_{b_1} and T_{b_2} are the most critical. So we add only these two parameters to the set of the parameters to be estimated via the global minimization of the cost function, i.e.,

$$\left\{\hat{x}_r^{(c)},\, \hat{T}_0^{(c)},\, \hat{\mathcal{P}}^{(c)}\right\} = \arg\min_{x_r,\, T_0,\, T_{b_1},\, T_{b_2}} J\left(x_r,\, T_0,\, \mathcal{P}\right) \tag{17}$$

As an important note, the number of parameters to be estimated in the thrust profile is a compromise between the prior knowledge on the profile and the information obtained through sensor detections. To further explain, if we have good prior information on the missile profile but limited detections, we can estimate only a small number of parameters with reasonable accuracy; however, if we have relatively good measurement information, we can estimate more parameters (e.g., the parameters determining the pitch-over angle) accurately. In the present case, we limit our discussion to the burn-out time estimation only.

Case (d). Finally, we go one more step further to relax the unity detection probability assumption by assuming a cloud cover, in which case we miss the first $j-1$ measurements. The maximum likelihood estimates of the target state $\hat{x}_r^{(d)}$, launch time $\hat{T}_0^{(d)}$, and the thrust profile $\mathcal{P}^{(d)}$ are obtained via the global minimization of the cost function over x_r, T_0, T_{b_1} and T_{b_2} i.e.,

$$\left\{\hat{x}_r^{(d)},\, \hat{T}_0^{(d)},\, \hat{\mathcal{P}}^{(d)}\right\} = \arg\min_{x_r,\, T_0,\, T_{b_1},\, T_{b_2}} J\left(x_r,\, T_0,\, \mathcal{P}\right) \tag{18}$$

where the following modified version of (12) is used

$$J\left(x_r,\, T_0,\, \mathcal{P}\right) \triangleq \left\|\nu\left(x_r,\, T_0,\, \mathcal{P}\right)\right\|^2 = \sum_{k=j}^{M} \left\| D_k^{-1}\left[\hat{z}_k\left(x_r,\, T_0,\, \mathcal{P}\right) - z(k)\right]\right\|^2 \tag{19}$$

Note that the cost function $J(\cdot)$ in all the above four cases, is a sum of squares of nonlinear functions of the parameters being estimated (x_r, T_0, T_{b_1} and T_{b_2}), and hence the right hand sides of (15)–(18) involve the solution of a nonlinear least squares (NLS) problem. A robust implementation of the Levenberg-Marquardt algorithm, described in Appendix A, was used to solve for the estimates in the above four NLS problems.

3.1 Covariance of the Estimates

Expressions for the covariance associated with the four estimates are presented in this section. Let the matrix $\Gamma\left(x_r,\, T_0,\, \mathcal{P}\right)$ be the Jacobian[10] of the residual vector with respect to x_r, i.e.,

$$\Gamma\left(x_r,\, T_0,\, \mathcal{P}\right) \triangleq \left[\nabla_{x_r} \cdot \nu\left(x_r,\, T_0,\, \mathcal{P}\right)'\right]' = \begin{bmatrix} D_1^{-1}\left[\nabla_{x_r} \cdot \hat{z}_1\left(x_r,\, T_0,\, \mathcal{P}\right)'\right]' \\ \vdots \\ D_M^{-1}\left[\nabla_{x_r} \cdot \hat{z}_M\left(x_r,\, T_0,\, \mathcal{P}\right)'\right]' \end{bmatrix} \tag{20}$$

Similarly, let the vector $\omega_{T_0}\left(x_r,\, T_0,\, \mathcal{P}\right)$ be the derivative of $\nu\left(x_r,\, T_0,\, \mathcal{P}\right)$ with respect to T_0, i.e.,

$$\omega_{T_0}\left(x_r,\, T_0,\, \mathcal{P}\right) \triangleq \frac{\partial}{\partial T_0}\,\nu\left(x_r,\, T_0,\, \mathcal{P}\right) \tag{21}$$

and the vector $\omega_{T_{b_1}}\left(x_r,\, T_0,\, \mathcal{P}\right)$ and $\omega_{T_{b_2}}\left(x_r,\, T_0,\, \mathcal{P}\right)$ be the derivatives of $\nu\left(x_r,\, T_0,\, \mathcal{P}\left(T_{b_1}\, T_{b_2}\right)\right)$ with respect to T_{b_1} and T_{b_2}, respectively, i.e.,

$$\omega_{T_{b_1}}\left(x_r,\, T_0,\, \mathcal{P}\right) \triangleq \frac{\partial}{\partial T_{b_1}}\,\nu\left(x_r,\, T_0,\, \mathcal{P}\left(T_{b_1}\, T_{b_2}\right)\right) \tag{22}$$

$$\omega_{T_{b_2}}\left(x_r,\, T_0,\, \mathcal{P}\right) \triangleq \frac{\partial}{\partial T_{b_2}}\,\nu\left(x_r,\, T_0,\, \mathcal{P}\left(T_{b_1}\, T_{b_2}\right)\right) \tag{23}$$

We shall present expressions for the covariances of the estimates $\hat{x}_r^{(a)}$, $\hat{x}_r^{(b)}$, $\hat{x}_r^{(c)}$ and $\hat{x}_r^{(d)}$ in terms the above Jacobian and derivatives. In addition, $\Gamma(\cdot)$ and $\omega(\cdot)$ are also used in the minimization of the cost function via the Levenberg-Marquardt algorithm. Since the state propagation given in (4) is via numerical integration, we do not have an exact expression for either $\Gamma(\cdot)$ or $\omega(\cdot)$, therefore, these partial derivatives are computed using the method of finite differencing.

Exact expressions for the covariance matrices associated with these four estimates do not exist. However, in all four cases, we can obtain approximate covariance matrices based on the CRLB that have been found to be quite accurate in our simulations.

Case (a). In a completely linear setting it can be shown that the ML estimator is both unbiased and efficient, i.e., the covariance of the estimate is equal to the CRLB, which is the inverse of the Fisher Information evaluated at the true value (Bar-Shalom, *et. al.*, 1993). In the present nonlinear context, we assume that these properties of the ML estimator hold approximately. Hence, the covariance $P_{(a)}$ associated with $\hat{x}_r^{(a)}$ is approximately given by:

$$\mathsf{Cov}\left\{\hat{x}_r^{(a)}\right\} \triangleq P_{(a)} \approx \left[\Gamma'_{(a)}\,\Gamma_{(a)}\right]^{-1} \tag{24}$$

[10] Denoting the gradient of a scalar as a column vector, the gradient of a scalar z with respect to a n dimensional vector x is defined as the $n \times 1$ vector $\nabla_{\mathbf{x}} z$ whose j^{th} component is $\frac{\partial z}{\partial x_j}$. Extending this notation to the derivative of a vector, we define the Jacobian of an m dimensional vector $\mathbf{y}$ with respect to an n dimensional vector x as the $m \times n$ matrix $[\nabla_{\mathbf{x}} \mathbf{y}']'$ whose $(i,j)^{\text{th}}$ term is $\frac{\partial y_i}{\partial x_j}$.

where $\Gamma_{(a)} \triangleq \Gamma\left(\hat{x}_r^{(a)},\, T_0^{\text{true}},\, \mathcal{P}^{\text{true}}\right)$ is the Jacobian of the residual vector $\nu\left(x_r,\, T_0,\, \mathcal{P}\right)$.

Case (b). In this case the minimization is over both x_r and T_0, hence the Jacobian is now $\left[\Gamma_{(b)}(\cdot)\ \ \omega_{(b)}(\cdot)\right]$, i.e., with an additional column. Using this, under the same assumptions as in case (a), the expression for the CRLB becomes

$$\mathsf{Cov}\left\{\begin{bmatrix}\hat{x}_r^{(b)}\\ \hat{T}_0^{(b)}\end{bmatrix}\right\} \approx \left\{\left[\Gamma_{(b)}\ \ \omega_{(b)}\right]'\left[\Gamma_{(b)}\ \ \omega_{(b)}\right]\right\}^{-1} = \begin{bmatrix}\Gamma'_{(b)}\Gamma_{(b)} & \Gamma'_{(b)}\omega_{(b)}\\ \omega'_{(b)}\Gamma_{(b)} & \omega'_{(b)}\omega_{(b)}\end{bmatrix}^{-1} \tag{25}$$

where $\omega_{(b)} \triangleq \omega_{T_0}\left(\hat{x}_r^{(b)},\, \hat{T}_0^{(b)},\, \mathcal{P}^{\text{true}}\right)$ and $\Gamma_{(b)} \triangleq \Gamma\left(\hat{x}_r^{(b)},\, \hat{T}_0^{(b)},\, \mathcal{P}^{\text{true}}\right)$ are, as in case (a), the partial derivatives evaluated at the estimates. Using the equations for the partitioning of the inverse of a partitioned matrix in (25), we can obtain the following expression which approximates the covariance of $\hat{x}_r^{(b)}$:

$$\mathsf{Cov}\left\{\hat{x}_r^{(b)}\right\} \triangleq P_{(b)} \approx \left[\Gamma'_{(b)}\left\{I - \frac{\omega_{(b)}\cdot\omega'_{(b)}}{\|\omega_{(b)}\|^2}\right\}\Gamma_{(b)}\right]^{-1} \tag{26}$$

Similarly, the standard deviation of the error in the estimated launch time $\hat{T}_0^{(b)}$ is approximately given by:

$$\mathsf{Std}\left\{\hat{T}_0^{(b)}\right\} \triangleq \Delta_{(b)} \approx \frac{1}{\|\omega_{(b)}\|}\left[1 + \frac{\omega'_{(b)}\Gamma_{(b)}\, P_{(b)}\, \Gamma'_{(b)}\omega_{(b)}}{\|\omega_{(b)}\|^2}\right]^{\frac{1}{2}} \tag{27}$$

Case (c). In this case, we use the dotted line in Fig. 3 as the thrust profile model. Due to the low observability and the relatively small number of measurements available, we only consider the contribution of the error in the critical parameters T_{b_1} and T_{b_2} of the constant mass flow rate thrust profile model, along with the uncertainty in $\hat{x}_r$ and $\hat{T}_0$.[11] Let us also define $\mathcal{T} \triangleq [T_0\ \ T_{b_1}\ \ T_{b_2}]'$ and $\omega_{(c)}(\cdot) \triangleq \left[\omega_{T_0}(\cdot)\ \ \omega_{T_{b_1}}(\cdot)\ \ \omega_{T_{b_2}}(\cdot)\right]$. The minimization is over x_r and $\mathcal{T}$ and the Jacobian becomes $\left[\Gamma_{(c)}(\cdot)\ \ \omega_{(c)}(\cdot)\right]$. Under the same assumptions as in case (a), using the CRLB, one has

$$\mathsf{Cov}\left\{\begin{bmatrix}\hat{x}_r^{(c)}\\ \hat{\mathcal{T}}^{(c)}\end{bmatrix}\right\} \approx \left\{\left[\Gamma_{(c)}\ \ \omega_{(c)}\right]'\left[\Gamma_{(c)}\ \ \omega_{(c)}\right]\right\}^{-1} = \begin{bmatrix}\Gamma'_{(c)}\Gamma_{(c)} & \Gamma'_{(c)}\omega_{(c)}\\ \omega'_{(c)}\Gamma_{(c)} & \omega'_{(c)}\omega_{(c)}\end{bmatrix}^{-1} \tag{28}$$

where $\omega_{(c)} \triangleq \omega\left(\hat{x}_r^{(c)},\, \hat{T}_0^{(c)},\, \hat{\mathcal{P}}^{(c)}\right)$, $\omega \triangleq \left[\omega_{T_0}\,\omega_{T_{b_1}}\,\omega_{T_{b_2}}\right]$ and $\Gamma_{(c)} \triangleq \Gamma\left(\hat{x}_r^{(c)},\, \hat{T}_0^{(c)},\, \hat{\mathcal{P}}^{(c)}\right)$ are, as in case (a), the partial derivatives.

An extra term $P_{\mathcal{P}}$ needs to be added to the covariance matrix $\mathsf{Cov}\left\{\hat{x}_r^{(c)}\right\}$ obtained above to account for the inaccuracy of the thrust profile model. Assuming a constant acceleration error reflected by $\sigma_{\mathcal{P}}$, this term is

$$P_{\mathcal{P}} = G\left(\hat{T}_{b_1}^{(c)} + \hat{T}_{b_2}^{(c)}\right)\left[\sigma_{\mathcal{P}}^2\, I\right] G\left(\hat{T}_{b_1}^{(c)} + \hat{T}_{b_2}^{(c)}\right)' \tag{29}$$

where

$$G(\Delta) = \begin{bmatrix}\frac{1}{2}\Delta^2\, I\\ \Delta\, I\end{bmatrix} \tag{30}$$

Case (d). In this case, the covariance is the same as above except for the loss of information caused by the missing of first $j-1$ detections, i.e., (19) replaces (12) in evaluating (28).

[11] See the discussion in the second paragraph in Section 3 case (c).

3.2 Position and Error Covariance at Radar Acquisition

In the following we discuss the prediction of the position and its error covariance at acquisition at a certain nominal time by a surface-based radar located in the vicinity of the impact point. The estimated target state $\hat{x}_r$ at the reference time t_r is used to determine the time $\hat{t}_a$ when the target enters the acquisition range of a surface-based radar. The reference time t_r is set to the time when the last detection occurs at the end of the boost phase. The determination of the acquisition time $\hat{t}_a$ from $\hat{x}_r$, t_r, the radar position and its range is given in (Yeddanapudi, *et. al.*, 1996). Once $\hat{t}_a$ is determined, the predicted state $\hat{x}_a = \left[\hat{\mathbf{r}}_a' \ \dot{\hat{\mathbf{r}}}_a'\right]'$ and position error covariance E_a are obtained by propagating the estimated state $\hat{x}_r$, and its associated covariance, from the reference time t_r to $\hat{t}_a$. Since the motion of the missile from t_r to $\hat{t}_a$ is governed by Kepler's laws, there exists a computationally efficient algorithm (instead of numerical integration) for propagating the state and computing the Jacobian $\nabla_{x_r}\mathbf{r}(\hat{t}_a)'$ (Yeddanapudi, *et. al.*, 1995). This algorithm is given in Appendix B. Using this Jacobian, we can write

$$\mathsf{Cov}\left\{\hat{\mathbf{r}}_a\right\} \triangleq E_a \approx \left[\nabla_{x_r}\mathbf{r}(\hat{t}_a)'\right]' \cdot \mathsf{Cov}\left\{\hat{x}_r\right\} \cdot \left[\nabla_{x_r}\mathbf{r}(\hat{t}_a)'\right] \tag{31}$$

The error covariance matrix E_a calculated in the above equation is extremely useful in initiating the search by a surface-based radar located in the vicinity of the impact point. This (algorithm-calculated) matrix yields an estimate of the position error e_a as the square root of the sum of the diagonal elements of E_a, i.e.,

$$e_a \triangleq \sqrt{[E_a]_{1,1} + [E_a]_{2,2} + [E_a]_{3,3}} \tag{32}$$

In section 4.1 we show that the above algorithm calculated values for e_a that agree very well with the actual values determined using Monte Carlo runs.

3.3 Position and Error Covariance at Missile Launch

Now the reference time is set to the (estimated) launch time, rather than the last detection time. At the launch point, we need to estimate the position of the missile only on a two dimensional surface and find the associated covariance. In *launch local* coordinates[12], we force the states other than the first two in the position $\mathbf{r}$ to a fixed value and estimate only two state components, i.e., $x = [\mathbf{r}' \ \dot{\mathbf{r}}']' = [r_x \ r_y \ h_0 \ 0 \ 0 \ 0]$, where h_0 is a fixed altitude, and we estimate only r_x and r_y. This change in the state to be estimated will affect equations (20) to (28). Note that the reference time t_r is not fixed during the optimization algorithm except in case (a) where it is assumed to be known.

The 2×2 error covariance matrix E_l calculated in the above equation is useful in making decisions on searching for the missile launch site. This (algorithm-calculated) matrix yields an estimate of the position error e_l as the square root of the sum of the diagonal elements of E_l, i.e.,

$$e_l \triangleq \sqrt{[E_l]_{1,1} + [E_l]_{2,2}} \tag{33}$$

3.4 Initialization

To initialize the optimization search for the parameters when the reference time is set as the last measurement time, we can use prior knowledge on T_{b_1} and T_{b_2}. As an initial guess for T_0, $t_1 - 0.5\,\delta_{\zeta(1)}$ is good enough, where $\delta_{\zeta(1)}$ is the sampling interval of the sensor from which the first measurement is generated. In case (d) we use $t_M - T_{b_1} - T_{b_2}$ as initial guess for T_0, where t_M is the last measurement time. The initialization of the position/velocity parameters at time t_r is done through a grid search.

[12]We can use an initial guess in the vicinity of the potential launch site as the original point to set up the launch local coordinates. This is accurate because in the vicinity of the launch point, the earth surface can be assumed to be flat.

To start the optimization search for the parameters at the launch point, we still use prior knowledge on T_{b_1} and T_{b_2}. Since the algorithm is relatively insensitive to the choice of the initial guess for r_x and r_y, we can use an initial guess for the two dimensional position and a grid search for the launch time t_0 (which is also the reference time t_r) from the first measurement time backwards. Note that we need a relatively accurate initial t_0 because, obviously, a bad guess of reference time can shift the whole profile too far thus make the optimization fall easily into a local minimum.

4 Results

The experimental setup is as follows. The target launch location at $T_0 = 0\,\mathrm{s}$ is 30°N latitude and 45°E longitude; the launch azimuth (w.r.t. true north) of the target is 225°, i.e., the target is headed south-west at the time of launch. The target enters the ballistic phase (i.e., final stage burnout, see Fig. 1) at $T_1 = 90\,\mathrm{s}$ and then it enters the detection region, which is at a distance of approximately $206\,\mathrm{km}$ from a radar located at the true impact point at time $T_2 = 372\,\mathrm{s}$ and, finally, reenters the atmosphere at time $T_3 = 400\,\mathrm{s}$, at a reentry altitude of $h_3 \approx 38\,\mathrm{km}$.

We consider the following two scenarios. In the single sensor scenario, the sensor is located on a surveillance satellite which is in a circular orbit in the equatorial plane rotating in the westward direction, at an altitude of $600\,\mathrm{km}$. At the time of target launch (i.e., at $t = T_0$), the satellite is located at 0° latitude, $30^\circ E$ longitude. The sensor has a sampling interval of $5\,\mathrm{s}$ and for cases (a)–(c) we assume unity probability of detection, hence, the first measurement occurs within a sampling interval after launch. In case (d) the first measurement occurs after $t = 30\mathrm{s}$, i.e., six measurements are missed.

In the two sensor case, we have $N = 2$ passive sensors located on surveillance satellites which are in circular orbits in the equatorial plane, rotating in the westward direction, at an altitude of $600\,\mathrm{km}$ above mean sea level. At the time of target launch (i.e., at $t = T_0$) the satellite S_1 is located at 0° latitude and 0° longitude and satellite S_2 is located at 0° latitude and 90°E longitude. Both sensors have a sampling interval of $10\,\mathrm{s}$ and generate LOS measurements of the target in an asynchronous manner (the "phase difference" between them is randomized; the sensor sampling intervals are chosen for one sensor and two sensors scenarios so that one has the same number of detections in both scenarios). As in the single sensor case, for cases (a)–(c) both sensors have unity probability of detection, hence, the first measurement from both the sensors occurs within a sampling interval after launch. In case (d) the first measurement from both sensors occurs after $t = 30\mathrm{s}$. The missile becomes invisible to the sensors (in both scenarios) shortly after the last stage burnout, hence, the last measurement (or, last two measurements, in the two satellite case) can occur at most one sampling interval after the final stage burnout. The measurement covariance R_k for the sensor $S_{\zeta(k)}$ is taken as

$$R_k \stackrel{\triangle}{=} \sigma^2_{\zeta(k)}\, I_{2\times 2} \tag{34}$$

where $\sigma_{\zeta(k)}$ is the standard error in the LOS measurements (both azimuth and elevation) from sensor $S_{\zeta(k)}$. We present results for $\sigma_{\zeta(k)} = 15\,\mu$rad and $60\,\mu$rad, and in the two satellite scenario we assume, in our simulations, that both the sensors have the same measurement noise levels (i.e., $\sigma_1 = \sigma_2$, when $N = 2$).

In acquisition position prediction, we obtain the estimate $\hat{x}_r$ of the target state at the reference time $t_r = t_M$, i.e., the reference time is chosen to be the time of the last measurement. The predicted target state $\hat{x}_a$ and the predicted acquisition time $\hat{t}_a$ are obtained by propagating the state estimate $\hat{x}_r$ to a nominal range from a surface-based radar. In our simulations the radar is located at the true point of impact and we use an nominal acquisition range of approximately $206\,\mathrm{km}$. In the launch point estimation, we obtain the estimate $\hat{x}_r$ of the target state at the reference time $t_r = \hat{T}_0$, i.e., the reference time is chosen to be the time of the (estimated) launch time. All experimental data, unless otherwise indicated, presented

in this section are from 100 Monte Carlo runs. Simulation details pertaining to the four estimation problems are described before the major results of the simulations are summarized in the various subsections.

Case (a). In the simulations for this case, we set the launch time $T_0^{(a)} = T_0^{\text{true}}$ (i.e., to the true launch time) and the thrust profile $\mathcal{P}^{(a)} = \mathcal{P}^{\text{true}}$, where $\mathcal{P}^{\text{true}}$ is the true thrust profile, which consists of the solid lines shown in Fig. 3. The results of interest in this case are: the estimates of the position at acquisition $\hat{\mathbf{r}}_a$, the associated error region $\mathsf{Cov}\{\hat{\mathbf{r}}_a\}$, the estimates of the 2-D position at launch time T_0 and the associated error $\mathsf{Cov}\{\hat{\mathbf{r}}_l\}$.

Case (b). In this case we estimate the launch time $\hat{T}_0$, along with the state $\hat{x}_r$. The thrust profile is set to the true profile, as in case (a) above. The results of interest in this case, are those as in case (a), plus the statistics of the launch time estimates.

Case (c). In this case we estimate the target state $\hat{x}_r$, the launch time $\hat{T}_0$, and, in addition, the key parameters T_{b_1} and T_{b_2} in the thrust profile. In the simulations we shall use a piecewise hyperbolic model $\alpha_k(t)$ for the thrust magnitude and a linear model $\varphi_k(t)$ for the orientation, given by the following equations

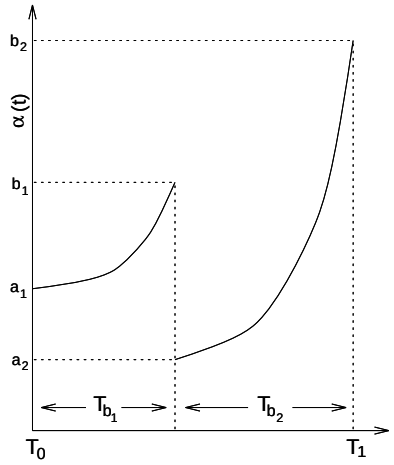


Figure 4: Piecewise hyperbolic model $\alpha_k(t)$ for the magnitude of the net acceleration.

$$\alpha(t) = \begin{cases} \dfrac{a_1\, b_1}{\left(\frac{t-T_0}{T_{b_1}}\right) a_1 + \left(\frac{T_0+T_{b_1}-t}{T_{b_1}}\right) b_1} & T_0 \le t \le T_0 + T_{b_1} \\ \dfrac{a_2\, b_2}{\left(\frac{t-T_0-T_{b_1}}{T_{b_2}}\right) a_2 + \left(\frac{T_0+T_{b_1}+T_{b_2}-t}{T_{b_2}}\right) b_2} & T_0 + T_{b_1} < t \le T_0 + T_{b_1} + T_{b_2} \end{cases} \tag{35}$$

$$\varphi(t) = \varphi_0 + \frac{\varphi_2 - \varphi_0}{T_{b_1} + T_{b_2}}\,(t - T_0) \qquad T_0 \le T_0 + T_{b_1} + T_{b_2} \tag{36}$$

where $a_1,\ b_1,\ a_2,\ b_2$ are shown in Fig. 4 and $\varphi_0,\ \varphi_2$ specify the linear model for the orientation. The values of these parameters are chosen to yield a best fit to the actual curves as shown in Fig. 3. The times T_{b_1} and T_{b_2} are the first and the second stage burn-out times, respectively (also indicated in Fig. 4), used in the thrust profile $\mathcal{P}_k$. The justification behind this model is that most missiles have an approximately constant mass flow rate due to the near constant burning of the fuel in the following simplified form,

$$f = \frac{c_1}{c_2 - t} \tag{37}$$

where f denotes the thrusting force, c_1 and c_2 are constant and t is the time the missile has flown.

Case (d). Due to cloud cover we should not always expect prompt detection of the missile by the IR sensors located on the satellites. In this case, we model the loss of information by eliminating the measurements at $t \leq 30$s. With this limited information we can still get reasonably good estimate of the target state $\hat{x}_0$, the launch time $\hat{T}_0$, and the thrust profile parameters T_{b_1} and T_{b_2}.

4.1 Position Errors at the Nominal Acquisition Range

The RMS errors in the estimated position $\hat{\mathbf{r}}_a$ at time $\hat{t}_a$ are shown in Table 2. Both the actual errors and the values calculated by the algorithm are averaged over 100 Monte Carlo runs. The *algorithm-calculated* position error, in each run, is computed using (32). In Table 2, the RMS value of this algorithm calculated position error is compared with the RMS value of the actual position error, both using the data from 100 Monte Carlo runs. The "nominal" acquisition time is defined such that the estimated position $\hat{\mathbf{r}}_a$ of the missile is at the nominal detection range of a surveillance radar located at the true impact point. In each run, the actual position error is the distance between the estimate $\hat{\mathbf{r}}_a$ and the true position $\mathbf{r}_{true}\left(\hat{t}_a\right)$, while the algorithm-calculated value e_a is obtained from (32). The RMS values of e_a and the actual error from 100 Monte Carlo runs are shown in Table 2.

$\sigma_{\zeta(k)}$ (μrad)	**Acquisition Point RMS Position Error (in km)** Actual *Algorithm Calculated*							
	One Sensor Scenario				Two Sensor Scenario			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
15	1.89 2.39	4.37 2.94	11.10 11.37	25.40 28.75	0.97 0.83	2.22 1.98	9.90 10.60	14.24 14.02
60	10.67 9.58	14.46 12.83	29.57 34.77	60.30 106.61	3.62 3.31	9.02 8.13	20.10 31.88	34.12 47.59

Table 2: RMS position errors at time $\hat{t}_a$, for the single and two satellite scenarios at $\sigma_{\zeta(k)} = 15$ and $60\,\mu$rad. Note: In case (c) and (d), the profile modeling noise is $\sigma_{\mathcal{P}} = 0.1g$.

The results in Table 2 clearly show that the algorithm-calculated values match well with the actual ones, in both the single and two satellite scenarios for both the low and high measurement error levels. Secondly, we observe that the errors increase steadily as one goes from case (a) to case (d), due to the incorporation of unknown launch time, incomplete thrust profile, and delayed detection into the estimation problem. Finally, the substantially smaller position errors in the two satellite scenario over the single satellite scenario show the advantage of improved observability obtained by using two satellites.

The cueing region obtained in a typical run for the most realistic case (d) with two sensor measurements and 60μrad angle measurement noise, is shown in Fig. 5. This illustrates the size of the cueing region, and that the true position of the missile does lie within the indicated 2σ region about the predicted location. The figure is drawn along the radar axes defined below. The radar is at the true impact point with the range axis directed towards the true missile position at time T_2. The "up-down"axis is a unit vector in the missile's orbital plane and is perpendicular to the range axis. The "left-right" axis completes the right-hand coordinate system.

4.2 Position and Launch Time Errors at the Missile Launch

The two dimensional RMS errors in the estimated position $\hat{\mathbf{r}}_l$ at time $\hat{t}_l$ are shown in Table 3. Both the actual errors and the values calculated by the algorithm, averaged over 100 Monte Carlo runs are shown.

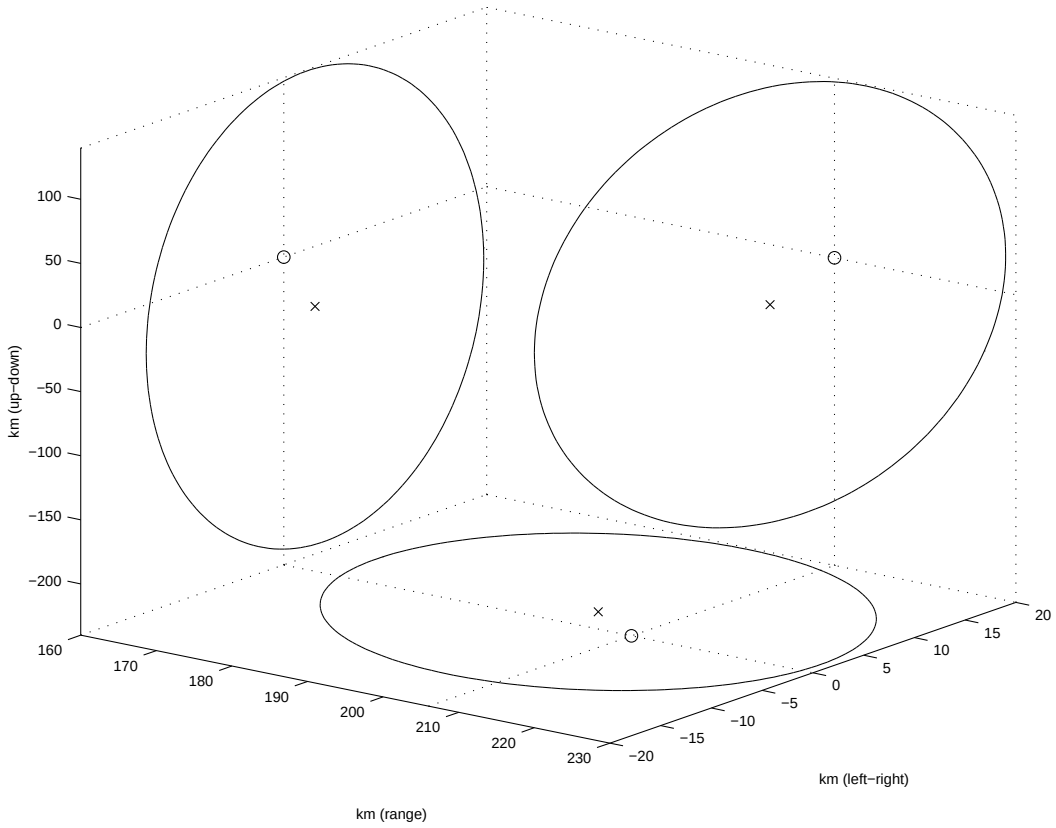


Figure 5: The 2σ cueing region for two sensor scenario with 60μrad measurement noise in case (d) with its projections ($\times$: the predicted position $\hat{r}_a$; o: the true position $r_{\rm true}$).

The *algorithm-calculated* position error, in each run, is computed using (33). In Table 3, the RMS value of this algorithm calculated position error is compared with the RMS value actual position error, both using the data from 100 Monte Carlo runs. In each run, the actual position error is the distance between the estimate $\hat{\mathbf{r}}_l$ and the true position $\mathbf{r}_{true}\left(T_0\right)$. The RMS values of e_l and the actual error of the launch time from 100 Monte Carlo runs are shown in Table 3. Fig. 6 shows the launch point uncertainty region in the launch local coordinates for a typical run.

The launch time estimation is done together with the 2-D position estimation in cases (b) through (d). The RMS values of the algorithm calculated error and the actual error from 100 Monte Carlo runs are shown in Table 4.

4.3 Consistency of the Estimator

The position errors in Table 2 and Table 3 indicate that estimate $\hat{\mathbf{r}}_a$, $\hat{\mathbf{r}}_l$ and the algorithm-calculated error matrix E_a, E_l are consistent with the true values. In this section we present a more rigorous test of the consistency of the algorithm by considering the entire error matrix E_a and E_l. For this purpose we define the normalized estimation error squared (NEES) (Bar-Shalom, *et. al.*, 1993) as

$$c_a \stackrel{\triangle}{=} \left[\hat{\mathbf{r}}_a - \mathbf{r}_{true}\left(\hat{t}_a\right)\right]' \left[E_a\right]^{-1} \left[\hat{\mathbf{r}}_a - \mathbf{r}_{true}\left(\hat{t}_a\right)\right] \tag{38}$$

and

$$c_l \stackrel{\triangle}{=} \left[\hat{\mathbf{r}}_l - \mathbf{r}_{true}\left(T_0\right)\right]' \left[E_l\right]^{-1} \left[\hat{\mathbf{r}}_l - \mathbf{r}_{true}\left(T_0\right)\right] \tag{39}$$

$\sigma_{\zeta(k)}$ (μrad)	**Launch Point RMS Position Error (in km)** Actual *Algorithm Calculated*							
	One Sensor Scenario				Two Sensor Scenario			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
15	0.081 0.136	0.139 0.140	0.204 0.316	0.391 0.653	0.043 0.043	0.045 0.043	0.054 0.051	0.073 0.097
60	0.254 0.543	0.677 0.536	1.034 1.263	1.739 2.501	0.181 0.171	0.182 0.182	0.225 0.205	0.348 0.389

Table 3: RMS launch position errors at time $\hat{t}_l$, for the single and two satellite scenarios at $\sigma_{\zeta(k)} = 15$ and $60\,\mu$rad.

$\sigma_{\zeta(k)}$ (μrad)	**Launch Time RMS Error (in s)** Actual *Algorithm Calculated*							
	One Sensor Scenario				Two Sensor Scenario			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
15	N/A	0.029 0.011	0.051 0.067	0.084 0.158	N/A	0.024 0.011	0.046 0.044	0.064 0.097
60	N/A	0.178 0.090	0.334 0.266	0.452 0.633	N/A	0.115 0.071	0.278 0.175	0.356 0.401

Table 4: RMS values of the actual and algorithm calculated errors in the launch time estimates for both the single and two sensor scenarios at $\sigma_{\zeta(k)} = 15$ and $60\,\mu$rad.

where, $\mathbf{r}_{true}\left(\hat{t}_a\right)$ is the true position of the missile at $\hat{t}_a$, which is obtained by propagating the true position of the missile from the reference time t_r, using the procedure described in Appendix B; $\mathbf{r}_{true}\left(T_0\right)$ is the true position at the launch time. If the estimator is unbiased and if the algorithm-calculated covariance is equal to the true covariance of the estimator, then c_a defined above is chi-squared distributed with 3 degrees of freedom (Bar-Shalom, *et. al.*, 1993) and c_l with 2 degrees of freedom. Taking the average over 100 Monte Carlo runs, the two sided 95% probability bounds[13] on $\bar{c}_a$ are

$$\frac{\chi^2_{300}(0.975)}{100} \leq \bar{c}_a \triangleq \frac{1}{100}\sum_{i=1}^{N} c_{a,i} \leq \frac{\chi^2_{300}(0.025)}{100} \tag{40}$$

Similarly, the two sided 95% probability bounds on $\bar{c}_l$ are

$$\frac{\chi^2_{200}(0.975)}{100} \leq \bar{c}_l \triangleq \frac{1}{100}\sum_{i=1}^{N} c_{l,i} \leq \frac{\chi^2_{100}(0.025)}{100} \tag{41}$$

where $c_{a,i}$ and $c_{l,i}$ are the value of c_a and c_l obtained in the i^{th} Monte Carlo run, respectively. The values of $\bar{c}_a$ and $\bar{c}_l$ for the different scenarios considered are presented in Table 5.

It can be seen from the above tables that for both one and two sensors scenarios, $\bar{c}_a$ and $\bar{c}_l$ are in most cases within their theoretical bounds, which indicates that the algorithm is consistent both in terms of the estimates and the calculated error matrix for these cases. In acquisition position for the single sensor scenario, $\bar{c}_a$ in cases (a) and (b) are out of bounds. This is because of the ill-conditioned

[13]The notation used indicates that the probability mass to the right of the point $\chi^2_n(\alpha)$ is equal to α.

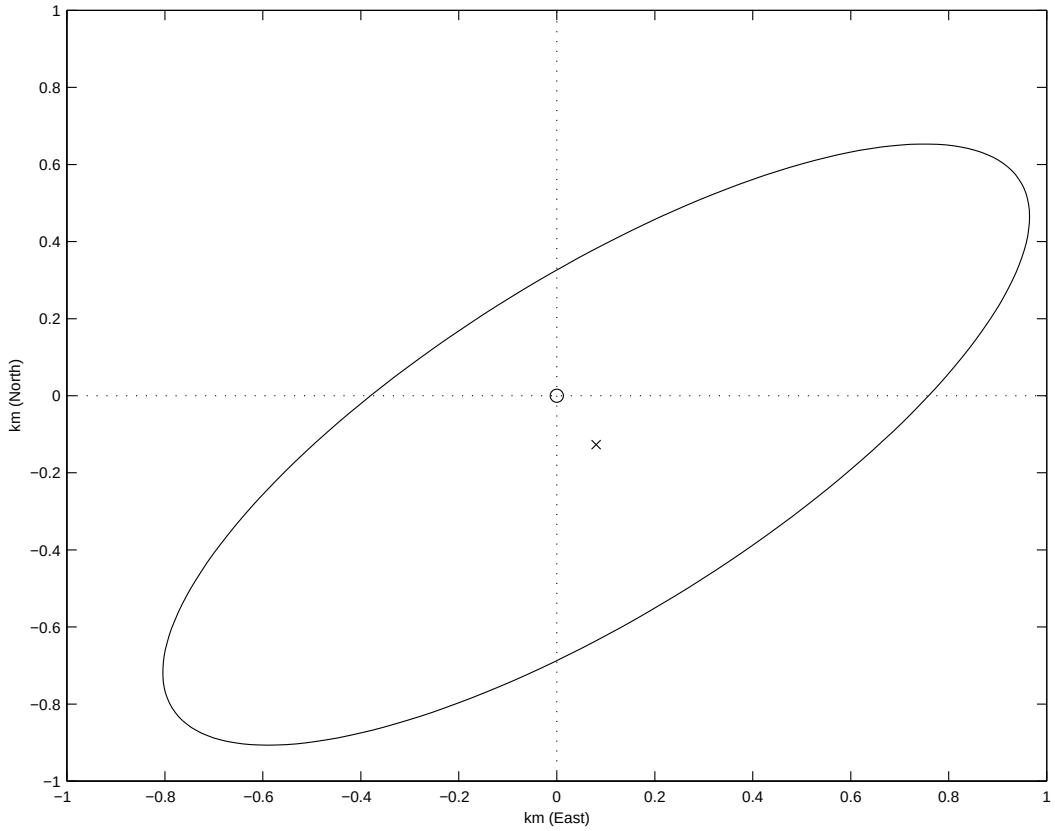


Figure 6: The 2σ launch position uncertainty region for two sensor scenario with 60μrad measurement noise in case (d) ($\times$: the estimated position $\hat{r}_l$ at $\hat{T}_0$; o: the true launch position).

observability from the single sensor, especially in high measurement noise level. In cases (c) and (d), the profile modeling noise redresses this problem. For launch position in case (d), the algorithm is slightly conservative in the error covariance matrix calculation.

5 Conclusions

An algorithm has been developed to estimate the state (position/velocity) of a ballistic missile based on passive observations during its boost phase. It has been shown that even with a single satellite-based IR sensor one can obtain a reliable estimate. With two such sensors the accuracy is substantially better. The algorithm-calculated covariances have been shown to be consistent with the actual errors and therefore, they can be used to set up the cueing region for acquisition by a radar near reentry, and the two dimensional search region for the missile launch site. This has been accomplished by a simple modeling of the thrust magnitude and pitch-over during boost. The algorithm can also estimate the unknown launch time. The major parameters in the thrust profile are also estimated together with the missile position/velocity and launch time. If the detection probability is less than unity, e.g., several measurements are missed due to cloud cover, the algorithm still performs well, although it yields higher errors.

$\sigma_{\zeta(k)}$ (μrad)	$\frac{\chi^2_{300}(0.975)}{100}$	**Normalized Estimation Error Squared for Acquisition Position**								$\frac{\chi^2_{300}(0.025)}{100}$
		One Sensor Scenario				Two Sensor Scenario				
		(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	
15	2.54	2.91	6.65	3.20	2.88	3.40	3.52	2.93	3.48	3.49
60	2.54	4.53	7.83	3.42	3.23	3.25	3.48	3.47	2.62	3.49

Table 5: Acquisition position NEES $\bar{c}_a$, averaged over 100 Monte Carlo runs for both the single and two sensor scenarios at $\sigma_{\zeta(k)} = 15$ and $60\,\mu$rad.

$\sigma_{\zeta(k)}$ (μrad)	$\frac{\chi^2_{300}(0.975)}{100}$	**Normalized Estimation Error Squared for Launch Position**								$\frac{\chi^2_{300}(0.025)}{100}$
		One Sensor Scenario				Two Sensor Scenario				
		(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	
15	1.63	1.77	2.11	1.54	1.31	2.03	1.67	2.11	1.35	2.41
60	1.63	1.85	2.36	2.08	1.19	2.10	1.88	2.31	1.24	2.41

Table 6: Launch position NEES $\bar{c}_a$, averaged over 100 Monte Carlo runs for both the single and two sensor scenarios at $\sigma_{\zeta(k)} = 15$ and $60\,\mu$rad.

A The Levenberg-Marquardt Algorithm

The nonlinear least squares (NLS) minimization problem can be solved using a variety of iterative algorithms. The performance of these algorithms is affected by two major factors. The first factor is the condition number of the Hessian $\nabla_x \nabla_x' J(x)$ of the cost function $J(x)$ at the minimum and in its neighborhood. The second factor is the choice of the initial point used to start the minimization algorithm.

For well conditioned problems[14], the performance of most algorithms is very good, in the sense that they converge rapidly to the minimum and the rate of convergence is not sensitive to the initial point used to start the algorithm. On the other hand, for ill-conditioned problems the performance of almost all the minimization algorithms is critically dependent on the initial starting point. As stated earlier, the minimization of the cost function associated with the ML target state estimation from LOS measurements is very ill-conditioned because of poor target motion observability. Hence, a robust minimization algorithm is required for this problem. The Levenberg-Marquardt algorithm, which is specially designed for the minimization of NLS functions, has been found to perform well for the present minimization problem and is described below.

Consider the cost function $J(x_0)$ which is to be minimized. Denoting the estimated minimum at the n^{th} iteration by $x_{0,n}$, the vector of normalized measurement residuals by $\nu(x_{0,n}) = \nu_n$ and the Jacobian by $\Gamma(x_{0,n}) = \Gamma_n$, we can form an affine approximation $\omega_n(x_0)$ to $\nu(x_0)$ as

$$\omega_n(x_0) = \nu_n + \Gamma_n\,(x_0 - x_{0,n}) \tag{42}$$

Using the above equation, we can form a local quadratic model $q_n(x_0)$ for the cost function as

$$q_n(x_0) = \omega_n(x_0)'\omega_n(x_0) = \nu_n'\nu_n + 2\nu_n'\Gamma_n(x_0 - x_{0,n}) + (x_0 - x_{0,n})'\Gamma_n'\Gamma_n(x_0 - x_{0,n}) \tag{43}$$

Under the assumptions that this quadratic model $q_n(x_0)$ is a good approximation to the actual cost function $J(x_0)$, the Gauss-Newton algorithm would update the estimate to the minimizer of this local

[14]The condition number of a matrix is the ratio of the largest and the smallest singular values. If the condition number of the Hessian of $J(x)$ is less than 10^3, then both the Hessian and the associated cost function $J(x)$ can be said to be well-conditioned.

quadratic approximation, i.e.,

$$x_{0,n}^{\mathsf{GN}} = \arg \min_{x_0 \in \Re^6} q_n(x_0) \tag{44}$$

It can shown that this Gauss-Newton update, $x_{0,n}^{\mathsf{GN}}$, is obtained as the least squares solution of a set of $2M$ linear equations, i.e.,

$$\Gamma_n \left(x_{0,n}^{\mathsf{GN}} - x_{0,n} \right) = -\nu_n \tag{45}$$

The Gauss Newton iterative algorithm, though very straightforward to implement, is not globally convergent. Convergence is assured only if the initial starting point $x_{0,0}$ is quite close to the global minimum. Quite a few modifications to this algorithm have been proposed to make it globally convergent. A class of algorithms that fall into this category are based on a modification to the Gauss-Newton algorithm suggested by (Levenberg, *et. al.*, 1944) and (Marquardt, *et. al.*, 1963). This algorithm (i.e., the Levenberg-Marquardt algorithm) also updates the estimate at the n^{th} iteration by trying to minimize the local quadratic model $q_n(x_0)$ at each iteration, but this updated estimate, $x_{0,n+1}$, is constrained to lie within a hyper-ellipsoid centered around $x_{0,n}$, i.e.,

$$x_{0,n+1} = \arg \min_{x_0 \in \Omega} q_n(x_0) \tag{46}$$

$$\Omega = \{x_0 \ : \ x_0 \in \Re^6 \text{ and } \|D_n \left(x_0 - x_{0,n}\right)\| \leq \delta_n\} \tag{47}$$

where D_n is a 6×6 diagonal scaling matrix and δ_n is the radius of the "scaled model trust region". It is easy to see that if the Gauss-Newton update is feasible, i.e., $\left\| D_n \left(x_{0,n}^{\mathsf{GN}} - x_{0,n} \right) \right\| \leq \delta_n$ then it is accepted as the estimate $x_{0,n+1} = x_{0,n}^{\mathsf{GN}}$. However, in general, $x_{0,n}^{\mathsf{GN}}$ may lie outside the trust region; in such a case, the updated estimate is obtained by solving the following min-max problem:

$$x_{0,n+1} = \arg \min_{x_0 \in \Re^6} \max_{\lambda \in \Re} \left[q_n(x_0) + \lambda \left(\|D_n \left(x_0 - x_{0,n}\right)\|^2 - \delta_n^2 \right) \right] \tag{48}$$

where λ is the Lagrangian multiplier, which in the present context is called the Levenberg parameter. It can be shown that given a value of λ, we can solve for a $x_{0,n}(\lambda)$, (i.e., solve the outer minimization problem) as the least squares solution of the following $(2M + 6)$ set of linear equations:

$$\begin{bmatrix} \Gamma_n \\ \lambda^{\frac{1}{2}} D_n \end{bmatrix} \left(x_{0,n}(\lambda) - x_{0,n} \right) = \begin{bmatrix} -\nu_n \\ 0_{6 \times 1} \end{bmatrix} \tag{49}$$

Note that $x_{0,n}(0) = x_{0,n}^{\mathsf{GN}}$, i.e., the Gauss-Newton update. This solution $x_{0,n}(0)$ may not feasible, but in general, it can be shown that there exists a $\lambda_n \in [0, \infty)$ such that for all $\lambda \geq \lambda_n$, $x_{0,n}(\lambda)$ is feasible. The updated estimate is then $x_{0,n+1} = x_{0,n}(\lambda_n)$. This value, λ_n, of the Levenberg parameter is obtained by solving $\|D_n \left(x_0(\lambda) - x_{0,n}\right)\| = \delta_n$. Some of the important aspects of this algorithm are summarized in the following remarks.

- Note that at each iteration, the Levenberg parameter λ_n has to obtained via an inner iterative loop. In each of these inner iterations for obtaining λ_n, we need to solve (49) for $x_{0,n}(\lambda)$. In the implementation of this algorithm by (Moré, *et. al.*, 1977) and (Moré, *et. al.*, 1980) a two stage QR procedure is used to solve for the Levenberg parameter. In addition, this procedure factors the the Jacobian matrix so that the intermediate parameters required for updating δ_n can be computed very efficiently.

- Another important aspect in the evaluation of λ_n is the choice of the iterative method used in solving the nonlinear equation in λ. Newton's method for solving nonlinear equations can be used, but a much more efficient iteration has been suggested by (Hebden, *et. al.*, 1973) that exploits the special structure of the problem.

- The ii element of the diagonal scaling matrix D_n is updated at each iteration in the following manner:

$$[D_n]_{ii} = \max\left\{\left\|\text{col}_i\left(\Gamma_n\right)\right\|, [D_n]_{ii}\right\}, n > 0 \tag{50}$$

  At the first iteration, we have $[D_0]_{ii} = \left\|\text{col}_i\left(\Gamma_0\right)\right\|$.

- The radius of the model trust region δ_n is updated based on how well the local quadratic approximation predicts the change in the function value. The ratio ϑ_n of the actual to the predicted reduction in the function value is given by

$$\vartheta_n = \frac{f\left(x_{0,n}\right) - f\left(x_{0,n}\left(\lambda_n\right)\right)}{f\left(x_{0,n}\right) - q_n\left(x_{0,n}\left(\lambda_n\right)\right)} \tag{51}$$

  The denominator of the expression on the right hand side of the above equation can be easily shown to be non-negative. Depending on the value of ϑ_n, the following update strategy is adopted :

  - If ϑ_n is negative, it implies that the value of the cost function has increased instead of going down. This happens if the local quadratic model is totally inadequate. In such a case, δ_n is drastically reduced (typically by a factor of 10) and the value of the Levenberg parameter is recomputed. The estimate is updated, i.e., $x_{0,n+1} = x_{0,n}\left(\lambda_n\right)$ only if $\vartheta_n \geq 0.0001$ (a small positive number).
  - If $0 < \vartheta_n \leq 0.25$, the estimate is accepted, but δ_n is reduced for the next iteration, since values of ϑ_n considerably less than one indicate that the local quadratic model does not accurately reflect the cost function behavior. The new trust region radius δ_{n+1} is computed using $\delta_{n+1} = \mu\delta_n$, where a value of $\mu \in [0.1\ \ 0.5]$ is chosen using a quadratic fit in the direction of the updated estimate.
  - If $\vartheta > 0.75$, then we choose $\delta_{n+1} = \delta_n$ if $\lambda_n = 0$, and $\delta_{n+1} \approx 2\delta_n$ if $\lambda_n > 0$. Since, in these cases, the local quadratic model seems to be a reasonably good approximation to the actual cost function, the Gauss-Newton update may be appropriate. To allow this update to be feasible, δ_{n+1} is increased.

A detailed theoretical development of this algorithm can be found in (Dennis, *et. al.*, 1983) and (Moré, *et. al.*, 1977). The algorithm is started using an initial point $x_{0,0}$ obtained using the initialization procedure described in section 3.4. The algorithm has been found to converge to the solution, i.e., $x_{0,N} \approx x_0^{\text{ML}}$ after a reasonably small number of iterations N (approximately 5 to 15 iterations, in most cases).

B Algorithm for State Propagation

Presented below is an algorithm that propagates the state of an object in a ballistic trajectory around the earth. Let $x(t)' = [\mathbf{r}(t)'\ \ \dot{\mathbf{r}}(t)']$ be the unknown state to be computed at time t, given the state $x_0' = \mathbf{x}(t_0)' = [\mathbf{r}_0'\ \ \dot{\mathbf{r}}_0']$ at the time t_0. The underlying theoretical concepts and the derivation of the equations used in this algorithm can be found in (Bate, *et. al.*, 1971). The gravitational parameter $\mu = 3.986012 \times 10^5\ \text{km}^3/\text{sec}^2$ and the convergence check parameter $TOL = 10^{-10}$ are used.

`STEP 1` $r_0 := \|\mathbf{r}_0\|$; $v_0 := \|\dot{\mathbf{r}}_0\|$; $q_0 := \frac{1}{\mu}\,\mathbf{r}_0'\,\dot{\mathbf{r}}_0$; $a_0 := \frac{2}{r_0} - \frac{v_0^2}{\mu}$; $p_0 := \frac{1 - a_0\, r_0}{\sqrt{\mu}}$;

`STEP 2` `if` $(a_0 \geq 0)$

$\alpha := \frac{a_0(t - t_0)}{\sqrt{\mu}}$;

`else`

$$\alpha := \frac{\text{sign}(t-t_0)}{\sqrt{-a_0}} \log\left(\frac{2(\sqrt{-a_0})^3(t-t_0)}{q_0\sqrt{-a_0}+p_0\text{sign}(t-t_0)}\right);$$

$\beta := a_0\alpha^2;$

STEP 3 `if` $(a \geq 0)$

$$c := \frac{1-\cos(\sqrt{\beta})}{\beta}; \qquad s := \frac{\sqrt{\beta}-\sin(\sqrt{\beta})}{\beta\sqrt{\beta}};$$

`else`

$$c := \frac{1-\cosh(\sqrt{-\beta})}{\beta}; \quad s := \frac{\sinh(\sqrt{-\beta})-\sqrt{-\beta}}{\beta\sqrt{-\beta}};$$

STEP 4 $\tau := p_0\alpha^3 s + q_0\alpha^2 c + \frac{r_0}{\sqrt{\mu}}\alpha;$

$\frac{\mathrm{d}\tau}{\mathrm{d}\alpha} := p_0\alpha^2 c + q_0\alpha(1-s\beta) + \frac{r_0}{\sqrt{\mu}};$

$\alpha := \alpha + \left[\frac{\mathrm{d}\tau}{\mathrm{d}\alpha}\right]^{-1}\ [(t-t_0)-\tau];$

STEP 5 `if` $([(t-t_0)-\tau] > TOL)$

`goto STEP 3`

STEP 6 $f := 1 - \frac{\alpha^2 c}{r_0}; \qquad g := (t-t_0) - \frac{\alpha^3 s}{\sqrt{\mu}}; \qquad \mathbf{r}(t) := f\,\mathbf{r}_0 + g\,\dot{\mathbf{r}}_0; \qquad r := \|\mathbf{r}(t)\|;$

STEP 7 $\dot{f} := \left(\frac{\sqrt{mu}}{r_0}\right)(s\beta-1)\left(\frac{\alpha}{r}\right); \qquad \dot{g} := 1 - \frac{\alpha^2 c}{r}; \qquad \dot{\mathbf{r}}(t) := \dot{f}\,\mathbf{r}_0 + \dot{g}\,\dot{\mathbf{r}}_0;$

The above steps yield the required state $x(t)' = [\mathbf{r}(t)'\ \ \dot{\mathbf{r}}(t)']$ at time t. In order to predict the covariance of the position $\mathbf{r}(t)$ by propagating the covariance of $\mathbf{r}(t_0)$ from t_0 to t, we need to compute the 6×3 matrix $\nabla_{x_0}\mathbf{r}(t)$. The computation of this matrix involves the following additional steps.

STEP 8 $\nabla_{x_0} r_0 := \begin{bmatrix} \mathbf{r}_0 \\ 0 \end{bmatrix}\left(\frac{1}{r_0}\right); \qquad \nabla_{x_0} v_0 := \begin{bmatrix} 0 \\ \dot{\mathbf{r}}_0 \end{bmatrix}\left(\frac{1}{v_0}\right);$

$$\nabla_{x_0} q_0 := \begin{bmatrix} \dot{\mathbf{r}}_0 \\ \mathbf{r}_0 \end{bmatrix}\left(\frac{1}{\mu}\right);$$

$$\nabla_{x_0} a_0 := (\nabla_{x_0} r_0)\left(\frac{-2}{r_0^2}\right) + (\nabla_{x_0} v_0)\left(\frac{-2v_0}{\mu}\right);$$

$$\nabla_{x_0} p_0 := (\nabla_{x_0} r_0)\left(\frac{-a_0}{\sqrt{\mu}}\right) + (\nabla_{x_0} a_0)\left(\frac{-r_0}{\sqrt{\mu}}\right);$$

STEP 9 $\frac{\mathrm{d}s}{\mathrm{d}\beta} := \frac{c-3s}{2\beta}; \qquad \frac{\mathrm{d}c}{\mathrm{d}\beta} := \frac{1-s\beta-2c}{2\beta};$

STEP 10 $b_1 := (\nabla_{x_0} q_0)(-\alpha^2 c) + (\nabla_{x_0} p_0)(-\alpha^3 s) + (\nabla_{x_0} r_0)\left(\frac{-\alpha}{\sqrt{\mu}}\right); \qquad b_2 := (\nabla_{x_0} a_0)(-\alpha^2);$

$$A := \begin{bmatrix} 3p_0\alpha^2 s + 2q_0\alpha c + \frac{r_0}{\sqrt{\mu}} & p_0\alpha^3\frac{\mathrm{d}s}{\mathrm{d}\beta} + q_0\alpha^2\frac{\mathrm{d}c}{\mathrm{d}\beta} \\ 2a_0\alpha & -1 \end{bmatrix};$$

$$[(\nabla_{x_0}\alpha)\ \ (\nabla_{x_0}\beta)] := [b_1\ b_2]\,A^{-1};$$

STEP 11 $\nabla_{x_0} f := \left[(\nabla_{x_0} r_0)\left(\frac{\alpha c}{r_0}\right) - (\nabla_{x_0}\alpha)(2c) - (\nabla_{x_0}\beta)\left(\alpha\frac{\mathrm{d}c}{\mathrm{d}\beta}\right)\right]\left[\frac{\alpha}{r_0}\right];$

$$\nabla_{x_0} g := \left[\left(\nabla_{x_0} \alpha \right) (3s) - \left(\nabla_{x_0} \beta \right) \left(\alpha \frac{\mathrm{d}s}{\mathrm{d}\beta} \right) \right] \left[\frac{-\alpha^2}{\sqrt{\mu}} \right];$$

STEP 12 $$\nabla_{x_0} \mathbf{r}(t) = \begin{bmatrix} fI_3 \\ gI_3 \end{bmatrix} + \left(\nabla_{x_0} f \right) \mathbf{r}_0' + \left(\nabla_{x_0} g \right) \dot{\mathbf{r}}_0';$$

References

Bar-Shalom, Y. and Li, X. R. (1993). *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Norwood, MA.

Bar-Shalom, Y. and Li, X. R. (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, Storrs CT.

Bate, R., et al. (1971). *Fundamentals of Astrodynamics.* New York, NY: Dover,.

Blair, W. D., Lawton, J. and Martell, G. (1995). Private Communication from NSWC Dahlgren Division, VA.

Becker, K. (1993). "Simple linear theory approach to TMA observability", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 575-578.

Chang, C. B. (1980). "Ballistic Trajectory Estimation with Angle-Only Measurements", *IEEE Transactions on Automatic Control*, vol. AC-25, no. 3.

Danis, N. J. (1993). "Space-Based Tactical Ballistic Missile Launch Parameter Estimation", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 412-424.

Dennis, J. E. and Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.

Drescher, G. H. and Sitzman, G. L. (1994). "Tactical ballistic missiles trajectory state and error propagation" *Technical Digest*, pp. 44-54, NSWC Dahlgren Division.

Golub, G. H. and Van Loan, C. F. (1989). *Matrix Computations*, Johns Hopkins Press, Baltimore, MD.

Hebden, M. D. (1973). "An algorithm for minimization using exact second derivatives", Rept. TP515, Atomic Energy Research Establishment, Harwell, UK.

Levenberg, K. (1944). "A method for the solution of certain nonlinear problems in least squares", *Quart. Appl. Math.*, vol. 2, pp. 164-168.

Marquardt, D. (1963). "An algorithm for least-squares estimation of nonlinear parameters", *SIAM J. Appl. Math.*, vol. 11, pp. 431-441.

Moré J. J. (1977). "The Levenberg-Marquardt algorithm: implementation and theory", in *Numerical Analysis* G. A. Watson, ed., Lecture Notes in Math. 630, Springer Verlag, Berlin.

Moré, J. J., Garbow, B. S. and Hillstrom, K. E. (1980). "User Guide for MINPACK-1", Report ANL-80-74, Argonne National Laboratory, Argonne, IL.

Perella, A. J., and Kuhn, W. W. (1996). "Cueing Performance Estimation Using Space Based Observations During Boost Phase", *Proc. 1996 Summer Comp. Sim. Conf.*, Portland, OR.

Press, W. H., et al. (1990). *Numerical Recipes in C*. Cambridge, UK: Cambridge University Press.

Rudd, J. G., Marsh, R. A. and Roecker, J. A. (1994). "Surveillance and tracking of ballistic missile launches", *IBM Journal of Research and Development*, vol. 38, no. 2, pp. 195-216.

Sviestins, E. (1995). "Multi-radar tracking for theater missile defense", *Proc. SPIE Conf. on Signal & Data Processing of Small Targets*, vol. 2561, pp. 384-394, San Diego, CA.

Taylor, J. H. (1979). "The Cramer-Rao Estimation error lower bound computation for deterministic nonlinear systems", *IEEE Transactions on Automatic Control*, vol. AC-24, no. 2, pp. 343-344.

Yeddanapudi, M., Bar-Shalom, Y., Pattipati, K. R. and Deb, S. (1995). "Ballistic missile track initiation from satellite observations", *IEEE Transactions on Aerospace and Electronic Systems*, vol.31, no. 3, pp. 1054-1071.

Yeddanapudi, M. (1996). "Estimation and Data Association Algorithms for Multisensor-Multitarget Tracking", Ph.D. Dissertation, *Electrical and Systems Engineering Department, University of Connecticut.*