Surveillance and tracking of ballistic missile launches

by J. G. Rudd
R. A. Marsh
J. A. Roecker

This paper begins with an overview of system requirements and design issues that must be considered in the design of algorithms and software for the surveillance and tracking of ballistic missile launches. Detection and tracking algorithms and approaches are then described for the processing of data from a single satellite and from multiple satellites. We cover track formation, missile detection, track extension, and global arbitration, and indicate how these functions fit together coherently. We include both profile-dependent and profile-free aspects of detection, tracking, and estimation of tactical parameters. In some instances, particularly in the area of track monitoring and in a discussion of how we accommodate intersatellite bias errors in line-of-sight measurements, we describe original work that has not been previously reported in the technical literature.

Introduction

Several infrared-based systems exist or are under development to provide satellite surveillance of ballistic missile launches. The most prominent of these are two U.S. Defense Department programs: the Defense Support Program, which provided Scud missile launch alerts to Patriot antimissile batteries during the Persian Gulf war, and the Brilliant Eyes program.

The primary purposes of satellite surveillance of ballistic missile launches are 1) to provide a timely report of each occurrence of a missile launch, 2) to estimate launch parameters (missile type, launch time, launch position and heading), and 3) to estimate present and future missile trajectories as a function of time during flight.

Within these broad purposes, however, the requirements and design characteristics of actual and hypothetical missile surveillance systems cover a wide spectrum. There are large differences in missions, sensors, potential scenarios, communications, and processing architecture. Figures 1 through 4 show simulations of hypothetical missile tracks, with a walking-dot approach to show target motion. This is accomplished on a display screen by showing multiple scans of target data while each data point is momentarily brightened in a rapid sequence over time.

The earliest satellite infrared (IR) surveillance systems were designed to detect and track a single bright missile (one emitting a relatively strong IR signal) of relatively long duration, as illustrated in **Figure 1**. Later systems were designed to accommodate multiple missiles that were widely separated in time or space and exhibited much shorter tracks, as illustrated in **Figure 2**. Such short tracks can originate from either a dim, short-duration missile or a relatively slow sensor scan rate or revisit rate, or both. (For most surveillance sensors, including scanning sensors

Figure 1

A hypothetical track from one bright, long-duration missile.

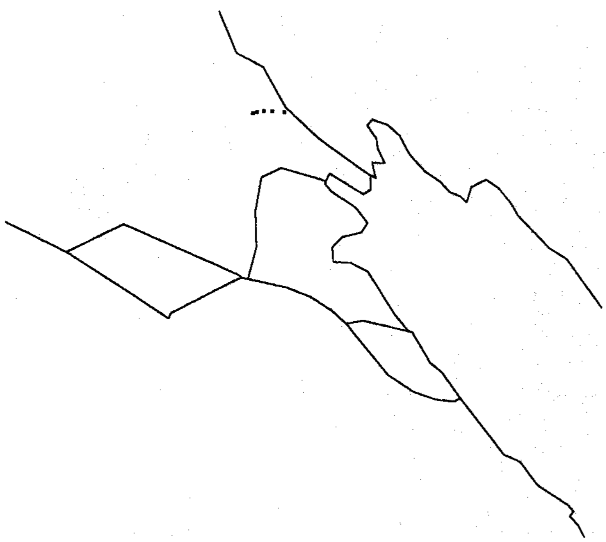


Figure 2

A hypothetical track from one dim, short-duration missile.

and certain kinds of staring mosaic sensors, the observations are organized into discrete scans or frame images, which here we also designate as *scans*.)

Modern surveillance systems are designed to accommodate a wide variety of scenarios, including salvos of closely spaced launches of either short- or long-duration missiles. Two such hypothetical salvo scenarios are illustrated in **Figures 3** and **4**.

Some typical design issues and challenges that must be considered in developing missile detection and tracking algorithms for modern surveillance systems are as follows:

- Monocular processing of observations from a single satellite vs. coordinated processing using concurrent observations from multiple satellites.
- Available computational resources and communication bandwidth and latency between sensors and processors.
- Variety of orbit characteristics and altitudes (low earth orbit to geosynchronous and beyond).
- Unknown or new missile types.
- Extent of sensor noise, background clutter, and sensor effects.

Several of these subjects are discussed at different points in this paper. However, our main thrust is to describe the following issues and approaches that critically affect the performance of missile detection and tracking algorithms:

- Profile-dependent and profile-free models of missile motion.
- Nominal missile flight vs. unpredictable maneuvers.
- Track processing in the presence of closely spaced multiple missiles.
- Sensor fusion and track segment fusion.
- Toleration of significant line-of-sight bias errors.
- Recalibrating the system in real time on the basis of the missile detections themselves.

Mission processing

The primary function of mission processing is to identify targets in real time from sensor data with a low false-alarm rate. Tracking and filtering are used to assemble candidate points for evaluation and to score the tracks so that the false tracks can be eliminated. To accomplish this, it is necessary to identify and associate one observation per scan per satellite into a single combined track for each actual missile being observed, so that an unambiguous report (and possibly one or more update reports) can be issued for each missile that has been launched. The processing flow necessary to accomplish this is illustrated in **Figure 5** and is organized as follows:

- Monocular (single-satellite) detection and tracking.
  - Track formation
  - Track detection
  - Track assignment
  - Track extension
- Stereo (two-satellite) detection and tracking.
  - Track formation
  - Track detection

- Track assignment
- Track extension
- N-satellite tracking.
  - Global arbitration
  - Track extension
- Event typing and tactical parameter estimation.

The *monocular detection and tracking* function is conceptually straightforward, and is described as follows. One or more *potential* tracks are formed, using simple intensity and motion criteria. Potential tracks are fitted in a least-squares sense to a generic model of missile motion. Those potential tracks that exhibit good fits to the model are designated as *feasible* tracks. Each feasible track is matched against one or more stored intensity and trajectory profiles of preselected known missile types. Those feasible tracks whose intensity and motion characteristics match those of at least one profile are designated as *detected* tracks. Detected tracks are submitted to track assignment. Track assignment resolves conflicts among the tracks and selects a set of nonconflicting detected tracks. Each of these detected tracks is sent to the global arbitration function, where a final decision is made whether to declare a detection. Each such track is also extended into subsequent scans until the track observations terminate.

The ideal result from monocular detection and tracking is a single detected monocular track per satellite for each observed missile launch.

When multiple satellites are viewing one or more missile launches, *stereo detection and tracking* is also performed. The stereo tracks are formed from the observations from selected pairs of satellites observing the missile activity. In earlier concepts, monocular detected tracks were the primary inputs to stereo track formation. However, in current concepts and work our group has focused on, stereo track formation occurs at the return level, in a process sometimes designated as *central-level fusion*.

For stereo track formation, as before, one or more *potential* tracks are formed using simple intensity and motion criteria. Potential tracks are fitted in a least-squares sense to a generic model of missile motion. Those potential tracks that exhibit good fits to the model are designated as *detected* stereo tracks. Detected stereo tracks are submitted to track assignment. As before, track assignment resolves conflicts among the tracks and selects a set of nonconflicting detected tracks; each of these detected tracks is sent to the global arbitration function, and each is extended into subsequent scans until the track observations terminate.

Stereo detection enables missile launches to be detected without any reliance on stored missile profiles. This is because the amount of measurement information in a stereo track has been sufficient in practice to detect the presence of a missile using generic models of missile dynamics without causing a high false-alarm rate. Depending on the specific mission, stereo detection also can enable shorter minimum tracks to be detected, and/or it can provide better tracking performance in the presence of closely spaced missile launches. (Of course, if the minimum detectable track were sufficiently short, such as one return from each of three satellites, it would no doubt

Figure 3

Hypothetical tracks from a salvo of closely spaced, long-duration missiles.

Figure 4

Hypothetical tracks from salvos of closely spaced, short-duration missiles.

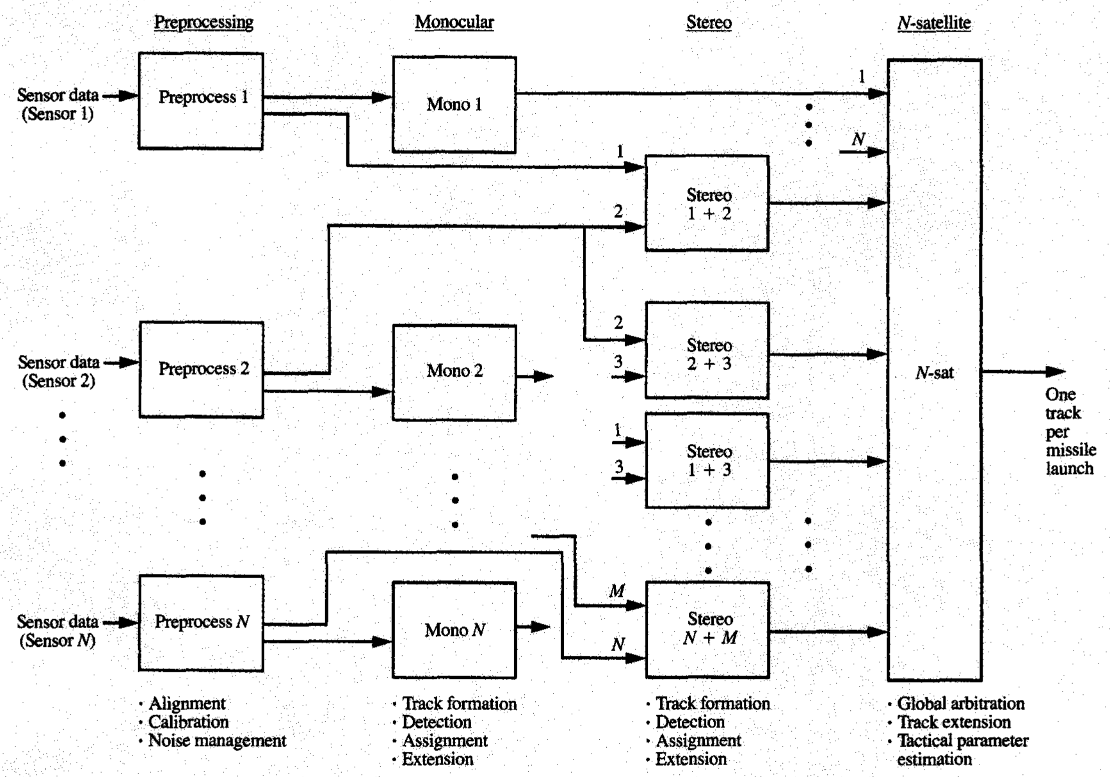


Figure 5

Mission processing flow.

be necessary to rely on stored profiles for multisatellite track formation and detection.)

Stereo detection and tracking may be performed in parallel with, and using the same observations as, monocular detection and tracking. In addition, it may be performed independently and in parallel using different pairs of satellites.

The ideal result from stereo detection and tracking is a single detected stereo track per satellite pair for each observed missile launch.

The primary functions of *global arbitration* are 1) to resolve conflicts and redundancy among the detected tracks that are received from the monocular and stereo detection and tracking functions; 2) to perform *track segment fusion*, i.e., to identify which track segments from different satellites or satellite pairs actually emanate from the same missile trajectory, and to merge them into a single N-satellite track; and 3) to extend the N-satellite tracks that are constructed.

The ideal result from global arbitration is a single N-satellite track for each observed missile launch.

Tactical parameter estimation includes *missile typing* and the estimation of launch parameters such as the launch location, the time and heading, the state vector (position and velocity) at specified times, and the estimated time and location of atmospheric re-entry or ground impact. Our group has performed much work in this area, but it is too voluminous to describe in this paper.

We describe the surveillance processing functions again and in more detail in later sections of the paper. Along the way we establish a foundation of surveillance concepts and algorithms that will enable those sections to be fully grasped.

- *Models of ballistic missile flight*

Ballistic missiles that are interesting for surveillance purposes are generally constrained to fly a *gravity turn* trajectory for at least the first stage of their flight. This is a trajectory where, after initial pitchover from vertical, the thrust vector is maintained in close alignment with the vehicle's velocity vector relative to the surrounding air, i.e., in ECF coordinates (see the last part of Appendix A). The angle between these two three-dimensional vectors is designated as the *angle of attack*. Its value is held to zero in an ideal gravity turn trajectory.

Flying such a trajectory allows gravitational acceleration gradually and naturally to complete the initial boost phase of the flight. Also, it minimizes the side forces on the relatively fragile skin of the rocket body while it is still within the earth's atmosphere.

Above the atmosphere this particular physical constraint does not apply. Nevertheless, to achieve maximum range for a given payload, or maximize payload for a given range, or maximize guidance accuracy, it is still very common to maintain a small angle of attack throughout the flight. Only a relatively few ballistic missile types are designed for significant maneuvers, whereby at some point in the flight the main thrust vector is dramatically pitched or yawed compared with the missile's velocity vector. Nevertheless, we have developed models of missile motion specifically to track and report such behavior.

Some models of missile flight that we have used and that we discuss in later sections of this paper are summarized below:

- Constant-acceleration polynomial model.
- Analytic tracker.
- Generalized analytic tracker.
- Profile-based models.

- *Line-of-sight bias problem*

While much design effort is devoted to accurate calibration and modeling of potential error sources, in the final result there are always some small errors in determining line-of-sight vectors to the missile from the sensor observations. Errors that vary almost independently from one scan time or sensor revisit time to the next are classified as *random* errors. Those that remain relatively constant are classified as *bias* errors.

Bias errors in time, line-of-sight determination, and satellite ephemeris all have similar effects on tracking behavior and launch parameter estimation. If the combined effect of such errors is even moderately large, a significant estimation error results from projecting the observations from the satellite to the missile. In addition, if two satellites each have large bias errors that are simultaneously projected to a common target, not only is there a resulting error in estimated target position, but also the observations will likely conflict (i.e., the projected line-of-sight observations will fail to intersect at a common point). This measured incompatibility in the observations, analogous to double vision, can result in poor performance in detection, tracking behavior, or position estimation accuracy, depending on the robustness of the algorithms.

Usually the combined effects of the bias errors are larger than those of the random errors. For such cases it has been necessary to design detection and tracking algorithms and parameter-estimation algorithms that are tolerant of these errors. In addition, we have devised an estimation procedure that is analogous to the application of corrective lenses to alleviate the symptoms of double vision. This is described later in the section on recalibration based on missile observations.

Initial processing of sensor observations

In a typical IR sensor data processing system, prior to missile detection and track processing, earlier processes have conditioned the raw sensor data from each sensor's data stream. The vast bulk of sensor noise and background data have been identified and discarded. Raw observations, or *returns* (see Appendix A) have been time-tagged and converted to attitude-corrected mission reference coordinates. Intensity measurements have been converted to engineering units. Spurious returns generated from internal sensor reflections, electronic crosstalk, or other sources have been identified and eliminated. Much of this front-end pre-track processing is very system-specific. However, some of these functions and design elements are common across systems, and precise design work in these areas is critical to successful mission performance. Some of these processes are discussed next.

- *Computation of satellite ephemerides*

Accurately computing the location of an IR source based on IR observations from a satellite requires accurate estimation of the satellite ephemerides, which determine the position and velocity of the observing satellite during the period of observation. The ephemerides of some satellites are estimated by using active ranging from tracking stations over extended periods to provide very precise state and modeling parameters. This enables a degree of processing autonomy through accurate computational propagation of ephemerides over extended future times. For some systems we have improved autonomy even further by developing methods to derive satellite position and velocity from sensor observations directly rather than from tracking stations. Some satellites are equipped with a Global Positioning System (GPS) receiver to self-determine the ephemerides. Finally, some systems use a combination of methods for robustness and improved accuracy.

- *Line-of-sight (LOS) determination*

Satellite attitude is normally controlled coarsely but autonomously in orbit using small jets and on-board sun and earth sensors. In addition, star measurements from an on-board star sensor are often used to estimate attitude angles and rates precisely. The attitude angle determination process must maintain nearly continuous attitude lock with small errors and high availability. It must also be robust in handling noise spikes from sensor-, celestial-, and solar-induced effects. Occasionally a physical disturbance of the vehicle adds to the complication.

Throughout the years of satellite operation and during daily or cyclic thermal loading, there are thermal distortions and mass property changes that affect the physical alignment of the sensors. The sensor focal plane itself expands and contracts during each orbit of the satellite, and this produces a distortion of the reference map of precise positions of the detector cells on the focal plane. To overcome these effects requires precise boresighting (see Appendix A), which is also performed using star observations from the satellite and modeling of focal plane motion.

- *Time determination and synchronization*

The precision alignments described above are highly dependent on an accurate understanding and accounting of the time reference systems used for each satellite and each processing station and on time synchronization. For example, since the satellites are some distance from the missile and from each other, there is a variable and nontrivial time delay (even at the speed of light) in communicating the sensor observations to the processors where precise time references are maintained.

Also, celestial time references for star catalogs and for orbit determination purposes are nontrivially different. Even variations in the missile-to-satellite distance and in the satellite's velocity relative to the earth can cause errors in estimated missile position. Also, when heterogeneous sensor fusion is performed, the use of observations from multiple satellites and different sensor types raises issues regarding the precise definitions of the time references within each of these surveillance programs. It is generally more difficult to communicate and resolve these differences across multiple large organizations than across a single program.

- *Sensor resolution*

The optical system, including the focal plane, is designed to provide detection of missiles at various ranges from the satellite. At long ranges, the pixel size and spacing may be larger than the size of the detectable missile plume being observed. At relatively short ranges, they may be smaller. In addition, detectors will often overlap; viz., several IR detector cells may receive energy from the same missile during the same scan. This resolution may limit the location accuracy that can be derived from the sensor observations.

To extract precise coordinates for a missile, care must be taken in merging nearly simultaneous observations from multiple detectors. This multiplicity of observations can increase accuracy by providing information which may be averaged to determine the center of the observed image. However, it also causes a complication in that the sensor observations from a given satellite on a given scan may not occur at the same time.

Another complication is that the use of a large number of cells provides the structure for internal reflections to occur. Any spurious observations caused by these reflections or crosstalk must be processed and eliminated to reduce the probability of false or inaccurate reports.

Finally, there is the issue of sensor resolution of closely spaced multiple missiles. Detectors respond to the IR energy received from the missiles, and there is an amplifier for each detector. The amplifiers are tuned to the sensor scan rate to provide an integration that is required for dim missiles (those emitting relatively weak IR signals) at long ranges. However, if two missiles are observed simultaneously by some of the same detectors, the signals combine, and it becomes difficult to resolve these sources. In such cases it is necessary to wait until later scans to detect separation of the missiles during some portion of the tracks.

- *Representative return formation*

This processing function collects a group of raw sensor returns that are closely spaced in time and observed position and are likely to have originated from the same missile or other IR source. It may occur both within one sensor array and across multiple sensor arrays within the same satellite. For each such group of returns, a two-dimensional centroid is computed, and the resulting single central return is called a *representative return* (see also Appendix A).

For most surveillance sensors (scanning sensors and certain types of staring mosaic sensors), these representative returns are organized into discrete scans (or frame images, which in this paper we also refer to as scans). Thus, the ideal result of the representative return formation function is a single representative return per missile per scan for each sensor viewing the launch and boost activity.

Monocular track formation

Monocular (i.e., single-satellite) track processing begins with the formation of feasible tracks from several contiguous scans of representative returns from a single sensor platform. A sliding window of scans is selected in which tracks are formed from the available representative returns (one return per scan per track) that meet prespecified limits or constraints in the following characteristics:

- Observed intensity
  - All returns in the track must be brighter than a predefined dim point threshold.
  - At least one return in the track must be brighter than a predefined bright point threshold.
- Observed two-dimensional motion
  - The observed motion from one scan to the next must be small enough to represent physically realizable missile speeds during the boost phase of interest.
  - Normally the observed motion from one scan to the next must be greater than what would be expected from random errors affecting the observations of a stationary IR source.
- Motion smoothness
  - In a sequence of three consecutive scans, the middle return must be within a prespecified distance from the straight-line segment connecting the first and last returns.

Combinations of returns which meet these constraints are retained as *potential* missile tracks, to be considered further in missile detection processing.

Track formation,[1] if performed inelegantly, can be a very large consumer of processing resources. If there are thousands of representative returns per scan, trillions of candidate tracks can be formed from all the returns in a four-scan sequence. This geometric explosion of candidate tracks is avoided by organizing the returns in either of two ways:

- Presorting the returns within each scan along a single coordinate: either a mission reference (MR) frame axis (x or y), or else the elevation coordinate, $\sin^{-1}\sqrt{x^2+y^2}$.
- Organizing the returns within each scan into two-dimensional bins in x and y mission reference coordinates.

With either procedure, the vast majority of infeasible combinations of returns are never considered. Processing resources are used to form and evaluate only those tracks that exhibit scan-to-scan motion that grossly resembles that of a missile.

Potential tracks are tested against either a constant-acceleration polynomial model or the analytic tracker model (described in the next two sections). Whichever motion model is used, tracks that fit this model reasonably well, i.e., that exhibit a sufficiently small goodness-of-fit score when tested against the model, are deemed *feasible* tracks; poorly fitting tracks are discarded.

Thus, a feasible track is a sequence of representative returns from a single satellite, one return per scan, all possibly corresponding to a single missile. A given representative return may be used by multiple feasible tracks within the same window of scans, since at this point no attempt is made to resolve tracking conflicts or ambiguities.

- *Constant-acceleration polynomial model*

For track formation and track extension, it has been common to model missile motion as a simple quadratic polynomial in each of three dimensions, usually in the ECF reference frame (see Appendix A). When projected onto the two dimensions of the mission reference frame of a single satellite, missile motion can then be approximated by a simple quadratic model in each of the two dimensions of observed motion. Ignoring the unobserved dimension and the actual distance to the missile enables the motion to be assumed independent in each of the two dimensions, so that the model is *linear*; i.e., the observations are linear functions of the states (polynomial coefficients) being estimated.

It is also possible in simultaneous coordinated processing of data from multiple satellites to assume that missile motion is quadratic in each of three dimensions. However, because the distance to the missile from each satellite cannot be ignored, this model is nonlinear.

In any case, the quadratic model falls far short of the mark in modeling the entire trajectory, or even the entire first stage of a long-range missile. However, for limited functions such as track formation, it is usually accurate enough to model short segments within a given missile stage. When the model is linear, it is appropriate to implement it via a simple sequential filter (as opposed to a batch least-squares filter or an extended recursive sequential filter). When this is done, the effects of the model error are reduced by the decaying memory of the sequential filter.

However, even within short segments of a trajectory, the simple quadratic model is suboptimal in its failure to represent the constrained angle of attack that ballistic missiles and rockets have during their first stage and usually thereafter. This consideration leads naturally to a modified polynomial model, described as follows.

- *Analytic tracker model*

This model assumes that in the absence of gravitational effects (which are removed separately by appropriate adjustments, as described later in this section), vehicle motion is along a straight line in three-dimensional space (as before in the ECF coordinate system), with constant acceleration along that line. Thus, vehicle motion is assumed to be quadratic in each of its three components versus time, with the constraint that the acceleration vector must be aligned with the velocity vector (zero angle of attack).

[1] Track formation is the only function described in this paper, other than missile typing, that makes even moderately strong use of intensity information. Missile typing makes very strong use of it.

Thus, in three dimensions (unless the average reference method is used, as described in a later section) this model contains seven states instead of nine polynomial coefficients:

- Three components of vehicle position at a reference time t_0.
- Three components of vehicle velocity at time t_0.
- One additional state k, which is the ratio of vehicle acceleration magnitude to vehicle speed at time t_0.

The analytic tracker model equations are given as follows:

$$x_i = x_p + x_v F_i,$$

$$y_i = y_p + y_v F_i,$$

$$z_i = z_p + z_v F_i,$$

where

- $F_i = \Delta t_i(1 + k\Delta t_i)$.
- $\Delta t_i = t_i - t_0$.
- (x_i, y_i, z_i) is the vehicle position at time t_i.
- (x_p, y_p, z_p) is the vehicle position at time t_0.
- (x_v, y_v, z_v) is the vehicle velocity at time t_0.

For monocular processing in two dimensions, this model projects naturally to five states:

- Two components of projected vehicle position at a reference time t_0.
- Two components of projected velocity at time t_0.
- One additional state k, which is the ratio of vehicle acceleration magnitude to vehicle speed at time t_0, as projected onto two dimensions.

Gravitational effects are removed separately for each observing satellite, as follows. Based on assuming a nominal vehicle altitude, the approximate position P of the vehicle is estimated, and the position S of the satellite is computed at the reference time t_0. Also, the value g of the earth's gravitational acceleration is computed. The line-of-sight radius R_L and zenith angle z are computed via

$$R_L = |P - S|,$$

$$z = \cos^{-1}\left(\frac{P}{|P|} \cdot \frac{P - S}{R_L}\right).$$

The elevation ε of the vehicle as viewed from the satellite is defined as

$$\varepsilon = \cos^{-1}\left(\frac{-S}{|S|} \cdot \frac{P - S}{R_L}\right).$$

Gravitational attraction to the earth from t_0 to t_i results in a change in vehicle position (approximately toward earth center) of

$$\Delta P_i = 0.5g\Delta t_i^2.$$

As projected onto the satellite focal plane, this is a two-dimensional vector with approximate magnitude

$$\Delta\varepsilon_i = \Delta P_i \frac{\sin z}{R_L},$$

pointing toward the center of the focal plane.

Thus the adjustment for gravitational effects is accomplished by subtracting $\Delta\varepsilon_i$ from the elevation ε_i of each observed return in the track window.

A strong advantage of the analytic tracker model over the simple quadratic model is that, since it more closely reflects the physical constraints on first-stage vehicle motion, it discriminates more powerfully against tracks formed from IR clutter and background phenomenology and against incorrectly formed tracks from closely spaced vehicle launches. However, the model is nonlinear, with the observations being a nonlinear function of the fifth state k. Therefore, this model is implemented via an iterative batch filter (or equivalent), rather than a simple sequential filter [1]. This model thus achieves more powerful discrimination than does a simple polynomial model, but at the cost of requiring more processing resources.

With a batch implementation, in order to avoid excessive model error and to reduce the computational load, the analytic tracker model is applied against a sliding window of returns from a few scans, rather than against an entire track of returns from many scans.

Once one accepts a batch implementation, additional possibilities for more powerful discrimination become more convenient and more attractive to implement. For example, large vehicles and space launches, when they are first detectable with shortwave IR sensors, exhibit values of k that are within a rather narrow and predictable range. Using a batch filter, one can apply the appropriate constraints directly on the value of this fifth state, k.

One special case of the two-dimensional version of the analytic tracker is of particular interest because it significantly reduces the processing resources necessary to implement the batch approach. If the standard deviations of line-of-sight errors are assumed to be constant for all observations in the track, there is a very fast closed-form solution available for this case, and no filter is needed. This closed-form variant is used in monocular track formation (described in a previous section) to discard infeasible tracks.

The closed-form solution of the analytic tracker equations for a single satellite is given below.

Given n normally distributed independent measurement pairs (X_i, Y_i) at times t_i, $i = 1 \cdots n$, where each X_i has zero mean and standard deviation σ_x, and each Y_i has zero mean and standard deviation σ_Y, we seek to minimize the cost function

$$\Phi = \sum_{i=1}^{n} \frac{[X_i - (X_0 + \dot{X}F_i)]^2}{\sigma_X^2} + \frac{[Y_i - (Y_0 + \dot{Y}F_i)]^2}{\sigma_Y^2}$$

with respect to the states X_0, Y_0, $\dot{X}$, $\dot{Y}$, and k, where $F_i = \Delta t_i(1 + k\Delta t_i)$ and where $\Delta t_i = t_i - t_0$, $i = 1 \cdots n$.

The solution is obtained by setting the partial of Φ with respect to each of the five states equal to zero, and solving for the values of the five states. The derivation is too long to include here, but the final result is given below.

To do so, we define the following quantities:

$$x_i = \frac{X_i}{\sigma_X} \quad i = 1 \cdots n, \qquad y_i = \frac{Y_i}{\sigma_Y} \quad i = 1 \cdots n,$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

$$u = \sum_{i=1}^{n} \Delta t_i, \qquad v = \sum_{i=1}^{n} \Delta t_i^2,$$

$$Q_x = \left(\sum_{i=1}^{n} x_i \Delta t_i^2\right) - \bar{x}v, \qquad Q_y = \left(\sum_{i=1}^{n} y_i \Delta t_i^2\right) - \bar{y}v,$$

$$W_x = \left(\sum_{i=1}^{n} x_i \Delta t_i\right) - \bar{x}u, \qquad W_y = \left(\sum_{i=1}^{n} y_i \Delta t_i\right) - \bar{y}u,$$

$$a = \left(n\sum_{i=1}^{n} \Delta t_i^4\right) - v^2, \qquad b = \left(n\sum_{i=1}^{n} \Delta t_i^3\right) - vu,$$

$$c = Q_x W_x + Q_y W_y, \qquad d = Q_x^2 + Q_y^2,$$

$$e = W_x^2 + W_y^2, \qquad f = nv - u^2,$$

$$A = ac - bd, \qquad B = ae - df, \qquad C = be - cf.$$

The solution for the fifth state k is given by the quadratic formula

$$k = \frac{-B + \sqrt{B^2 - 4AC}}{2A},$$

and the first four states are derived from k as follows:

$$p = \frac{n}{f + 2kb + ak^2}, \qquad h = \frac{(u + kv)}{n},$$

$$\dot{x} = p(W_x + kQ_x), \qquad \dot{y} = p(W_y + kQ_y),$$

$$x_0 = \bar{x} - h\dot{x}, \qquad y_0 = \bar{y} - h\dot{y},$$

$$\dot{X} = \dot{x}\sigma_X, \qquad \dot{Y} = \dot{y}\sigma_Y,$$

$$X_0 = x_0\sigma_X, \qquad Y_0 = y_0\sigma_Y,$$

Average reference method

A key problem in the development of tracking algorithms for surveillance systems has been the handling of line-of-sight (LOS) bias errors. Bias errors limit the accuracy of the launch location parameters and potentially degrade the tracking solutions for multiple satellite solutions. Some amount of LOS bias error is unavoidable due to the nature of the instrumentation. For a reasonably well calibrated system, the expected value of bias error is roughly one to two times the magnitude of the expected value of random error.

The primary method of reducing these bias errors is to conduct an on-orbit boresight alignment using stars and ground-based calibration sources in the field of view. This procedure works well; however, the results of the calibration are valid for a limited time interval because of the temperature changes of the satellite. The changes cause distortions of the structure and changes in mass distribution that change the alignment. To reduce the bias errors further, it is necessary to model the thermally induced changes and to conduct boresight calculations frequently. Other factors such as satellite ephemeris error or time reference differences can cause errors equivalent to the boresight errors, so these errors must be controlled as well.

Bias errors can be reduced even further by using a calibration beacon, which is a ground-based IR source that is precisely located. By placing one or more of these sources near the target area of interest, the bias errors can be minimized. Algorithms have been developed to use these measurements in real time and to feed the corrections into the mission processing. Even with this capability actively in place, however, there are times when bias errors are still present, such as during a satellite maneuver or when the beacons are obscured by clouds. The net effect is that the missile tracking and estimation algorithms must be designed to handle whatever bias errors remain after applying these procedures.

Algorithms using the nine-state polynomial model or the seven-state analytic tracker model of missile motion are vulnerable to these bias errors. When these models are used for simultaneous coordinated processing of data from multiple satellites, there can be a considerable degradation in the goodness of fit of model predictions against observations. This can degrade the ability of the tracking algorithms to distinguish correct tracks from tracks that are incorrectly formed from returns generated by multiple closely spaced missile launches.

We have used two basic approaches for these bias-induced problems. The first is to estimate one or more additional states that relate directly to measurement biases; the second is to model only relative missile motion and eliminate the states of missile position, so that the bias errors do not affect the cost function or the remaining states that are estimated. Each approach has its appropriate place.

For the first approach, consider a simplified case of two satellites, and assume, for observations emanating from a geographical area of interest, that the line-of-sight bias errors can be modeled as a simple (and small) translation of all the (x, y) coordinates of the focal plane. Thus, for two satellites there are four bias errors that could be considered. Given many precise observations of a missile over a relatively long period of time, and given a nearly perfect model of missile motion for this period of time (e.g., a free-fall model), all four bias errors may eventually become sufficiently observable to be estimated as filter states, merely from the many observations in this one stereo track. Such a case can occur, but it is unusual.

With the usual case of shorter tracks, imperfect models of missile motion, and less precise measurements, only the coplanarity residual (see Appendix A) is observable, and the other three components of bias error are indistinguishable from the estimated missile position states. Thus, usually it is practicable to add only this one bias-related state to the states that are estimated.

However, we have found that it is often preferable for purposes of tracking and detection to reduce the number of estimated states rather than increasing them. For example, if only three (x, y) observations have been made from each of two satellites, that represents a total of only 12 independent measurements. If the coplanarity residual is added to the seven states of the analytic tracker model, the total number of states is more than can reliably be estimated from the paucity of measurements. This produces numerical instability in the filter, exacerbated by the fact that the system is nonlinear; hence the overall estimation accuracy is degraded.

Suppose, then, that we temporarily "write off" the three position states and the coplanarity residual as if they were unobservable; i.e., we freeze the initial estimate of missile position (obtained through simple triangulation) that is normally used to initialize the filter. We then uncouple the remaining four analytic tracker states (three velocity states plus k) from the measurement bias errors.

One way that we have accomplished this historically is to reference all the returns in each single-satellite track to the first return from that track; i.e., if the *observation* pairs in the LOS coordinate frame (see Appendix A) are (X_i, Y_i), $i = 1 \cdots n$, we define $\Delta X_i = X_i - X_1$ and $\Delta Y_i = Y_i - Y_1$, $i = 1 \cdots n$; these "observations" would be the ones used in the filter.

This procedure can be loosely interpreted as computing a kind of finite-difference "velocity" of the observed track. We compute a corresponding "velocity" of the predicted track as follows.

Using the current estimate of missile states in ECF coordinates, we compute both missile and satellite position at the time of every return. We then compute the predicted unit line-of-sight vectors to the missile at these times, and convert them to LOS coordinates (see Appendix A), retaining only the (x, y) components in this frame. We perform these computations independently for each return from each observing satellite; thus, there is a different LOS coordinate system for each return.

If the *predicted* pairs in the LOS coordinate frame are (X_i', Y_i'), $i = 1 \cdots n$, we define $\Delta X_i' = X_i' - X_1'$ and $\Delta Y_i' = Y_i' - Y_1'$, $i = 1 \cdots n$. These are the "predicted" values obtained from the model of missile motion that are differenced with the corresponding "observations" to compute the residuals for the goodness-of-fit cost function.

Since each X_i is corrupted in the same manner as X_1 by the bias errors for that satellite, it follows that ΔX_i is not corrupted by bias errors, and neither are ΔY_i, $\Delta X_i'$, and $\Delta Y_i'$. The main price we pay for this benefit is to eliminate four observations (ΔX_1, ΔY_1, $\Delta X_1'$, $\Delta Y_1'$ per satellite) which now contribute no information. Thus, in our example, we now have eight measurements, with four states to be estimated. This reduction in states improves filter stability and reduces computational requirements. Most important, the behavior of this filter is unaffected by measurement bias errors.

In particular, assuming that the initial estimate of missile position is even approximately correct, the bias errors have virtually no effect on the estimates of the missile velocity states via this filter. Therefore, the estimates of missile speed, flight path angle, and heading are all essentially uncorrupted.

One disadvantage of referencing all returns in a track to the first return is that all of the resulting differences are corrupted in the same way by the random noise in the first return. Theoretically and ideally, the measurement covariance matrix should be constructed to take this effect into account. However, this would require significantly more computational resources, since we could no longer assume that the measurement covariance matrix is diagonal.

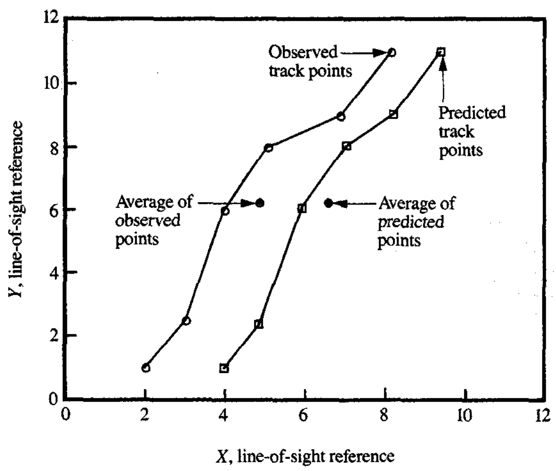
An attractive alternative solution is the average reference method, described as follows. As before, we segregate the track returns by satellite. But instead of referencing all returns from each single-satellite track to the first such return, we reference them to their centroid; i.e., if the observation pairs in the LOS coordinate frame are (X_i, Y_i), $i = 1 \cdots n$, we define $\Delta X_i = X_i - \bar{X}$ and $\Delta Y_i = Y_i - \bar{Y}$, $i = 1 \cdots n$. These "observations" are the ones used in the filter, and they are differenced with corresponding "predicted" values ($\Delta X_i'$, $\Delta Y_i'$) from the model of missile motion to compute the residuals for the goodness-of-fit cost function. **Figure 6** illustrates the use of this method in a single dimension.

As before, the centroid is corrupted by bias errors in the same way that each return is corrupted; hence, the (ΔX_i, ΔY_i) are not corrupted by measurement bias errors. However, statistically the centroid of a group of track returns has significantly less random error than does the first such return. Furthermore, when differencing from the centroid, it is optimal to assume that these "measurements" are independent; i.e., the measurement covariance matrix is diagonal. Precisely the same result is obtained using this method as when using the measurement covariance matrix that is derived by statistically rigorous methods.

Having estimated the missile velocity states in a bias-free manner, it remains to refine our initial estimate of the missile position states. This we accomplish by freezing the velocity state estimates and by estimating the position states using the full filter and the original (absolute) observations. In the case of the analytic tracker filter, this means formally using all seven states, but tightly constraining four of them with the *a priori* state covariance matrix.

In summary, compared with the alternative procedure of estimating additional states, the average reference method provides improved numerical stability, faster filter convergence, and reduced processing requirements. It also reduces the probability of selecting incorrectly formed tracks, and it improves tracking performance overall. However, it also increases the probability of favoring certain kinds of "ghost tracks," i.e., incorrect pairs of correctly formed monocular tracks that exhibit consistent velocity and acceleration characteristics. This relatively minor problem is the main price that is paid to achieve maximum toleration of measurement bias errors.

Appendix C contains a complete but concise implementation of the Analytic Tracker filter described here, including the Average Reference method.



Average reference method
Observation − prediction
$X_1 = -0.3$
$X_2 = -0.2$
$X_3 = -0.2$
$X_4 = -0.2$
$X_5 = +0.4$
$X_6 = +0.5$

X values	
Observed	Predicted
$X_1 = 2$	4.0
$X_2 = 3$	4.9
$X_3 = 4$	5.9
$X_4 = 5.1$	7.0
$X_5 = 6.9$	8.2
$X_6 = 8.2$	9.4
Ave. = 4.87	6.57

Figure 6

Illustrative use of average reference method.

Missile staging and maneuvering

Within a single stage of a liquid-propellant missile, in which there is an approximately constant mass flow rate, an assumption of constant thrust acceleration is reasonably valid over a small time period. However, solid-propellant missiles have a more variable thrust acceleration profile. Furthermore, for either kind of propellant, missile staging produces an abrupt change in thrust acceleration. In these circumstances the analytic tracker can be appropriate for weeding out grossly ill-fitting tracks, but it is often inadequate for more rigorous purposes.

In addition, some missiles are capable of achieving relatively large maneuvers during later stages, and this behavior departs significantly from the assumptions of the analytic tracker model. One or more abrupt changes in pitch or yaw may occur, with effects that are noticeable even during the relatively small window of scans over which the model is applied.

One approach to this problem is to add judiciously to the states of the analytic tracker filter, generalizing it just enough to accommodate this behavior, but not to the extent of eliminating the constraints completely. The resulting filter, which includes the analytic tracker as a special case, is described as follows.

- *Generalized analytic tracker (GAT)*

This profile-free filter estimates between four and seven states, depending on the application (track formation, track extension, or track segment fusion). It accepts as input representative returns from at least two satellites. Associated with each two-dimensional return position are assumed standard deviations of random line-of-sight error. Other inputs are the satellite position at the time of each representative return and a specified reference time near the times of the representative returns.

The outputs of this filter are a goodness-of-fit score (value of the cost function) which indicates the degree to which observed motion resembles that of a missile and the estimated filter states for the specified reference time.

If four states are estimated, the model is identical to the analytic tracker described previously, using the average reference method. The fifth state adds robustness in the presence of highly variable acceleration (e.g., missile staging). The sixth state adds the capability to track a maneuvering missile that yaws. The seventh state adds the capability to track a maneuvering missile that both pitches and yaws. These states are described next.

First three states:	Velocity vector in earth-fixed coordinates at the specified reference time.
State 4:	Ratio of acceleration magnitude to speed, for the first half of the time window containing the track returns.
State 5, if present:	Same as state 4, but for the second half of the time window.
State 6, if present:	Yaw angle at the specified reference time.
State 7, if present:	Pitch angle at the specified reference time.

Profile-dependent modeling and filtering

The gravity turn trajectory described previously can be characterized by simple rocket equations that nevertheless require numerical integration to produce the trajectory. Traditionally this integration has consumed too many processing resources to perform in real time for systems that must track many missiles simultaneously.

Therefore, mission processing has traditionally used off-line data bases that contain *a priori profiles* to represent nominal or possible missile intensities (measures of observed IR radiation) and trajectories as a function of time since launch.

A profile of a missile trajectory consists of tabular values of three-dimensional position as a function of time since launch. Usually these are expressed in terms of crossrange, downrange, and altitude relative to the launch point. (See the TR coordinate frame defined in Appendix A.) In the more complete implementations of profiles, these quantities have been compensated for the effects of earth rotation [2].

The portions of mission processing that use profiles are monocular detection, missile typing, and estimation of most of the tactical parameters (particularly the launch parameters). The other processing functions generally use profile-free models, as described in previous sections. An exception is when large gaps in the track data occur, i.e., when the accuracy of profile-free models is insufficient to bridge the gap satisfactorily.

Generally the processing database contains at least one profile for each known missile type or model, for each geographical region of interest. Early in monocular detection and tracking, when the particular missile type has not yet been determined, each feasible profile is fitted and scored against the sensor observations in a least-squares sense, using a *profile-dependent filter*.

Our profile-dependent filters are briefly characterized as follows:

- The estimation algorithm is a batch square root information filter (SRIF) [3, 4].
  - All observations from all sensors for the entire track may be used.
  - Each return contains standard deviations of the two-dimensional line-of-sight observations in LOS coordinates (see Appendix A).
- The position of each satellite is computed precisely at the time of each return from that satellite.
- Estimated filter states are
  - Launch time.
  - Launch azimuth, or heading.
  - (Optionally) launch latitude and longitude.
  - (Optionally) lofting angle, a typical first-stage profile variation.
  - (Optionally) pitch and yaw angles if the missile type is highly maneuverable in its later stages.

Profile-dependent filters are organized and implemented in much the same manner as is the analytic tracker filter illustrated in Appendix C. The main difference is in the prediction equations. The prediction equations for a four-state model (launch time, azimuth, latitude, and longitude) follow; see also Appendix B.

We define the following quantities:

T_{lp} = launch time,
α_{lp} = launch azimuth,
λ = geodetic latitude of the launch point,
ω = longitude of the launch point,
λ_c = geocentric latitude of the launch point,
P_{lp} = position vector of the launch point in ECF coordinates,
Q = local vertical unit vector associated with launch point position,
R_e = local earth radius at the launch point,
t = time since T_{lp} of a specified return;

from which it follows that

$$P_{lp} = R_e(\cos\lambda_c \cos\omega,\ \cos\lambda_c \sin\omega,\ \sin\omega),$$

$$Q = (\cos\lambda \cos\omega,\ \cos\lambda \sin\omega,\ \sin\omega).$$

We next define

ΔP = interpolated values of (crossrange, downrange, altitude) from the specified missile profile at time t,

$T_{\mathrm{TR}}^{\mathrm{LH}}$ = transformation matrix from TR to LH frame,

$T_{\mathrm{LH}}^{\mathrm{ECF}}$ = transformation matrix from LH to ECF frame (= transpose of $T_{\mathrm{ECF}}^{\mathrm{LH}}$);

$P = P_{\mathrm{lp}} - T_{\mathrm{LH}}^{\mathrm{ECF}} T_{\mathrm{TR}}^{\mathrm{LH}} \Delta P$ is the missile position at the time of the specified return in ECF coordinates.

Finally, we define

$$U_{\mathrm{LOS}} = T_{\mathrm{MR}}^{\mathrm{LOS}} T_{\mathrm{ECF}}^{\mathrm{MR}} \frac{P - S}{|P - S|}.$$

The first two components of U_{LOS} are the (x, y) predictions in the LOS frame. The remaining portions of this profile-dependent filter follow the design of the analytic tracker filter.

Monocular missile detection

In this function each feasible track is first evaluated by a profile-free filter. Here the missile motion is modeled as a quadratic polynomial in each of the two dimensions which the sensor observes, or with the analytic tracker model, or with the generalized analytic tracker model. Tracks whose goodness-of-fit scores from this profile-free filter are greater than a predefined threshold value are assumed not to represent a single missile track and are discarded.

Each feasible track that survives is submitted to a profile-dependent filter. Region determination is performed to determine the possible missile types which may correspond to the observed track. Corresponding profiles of missile characteristics in position and intensity are made available. The track is then evaluated by fitting the track returns in a least-squares sense to each of these profiles. The corresponding goodness-of-fit score is computed for each profile. The smallest score is then tested against a threshold. If the score passes the test, the track is retained for submission to track assignment.

Surviving tracks, along with their profile-dependent scores, are submitted as a group to an n-dimensional track assignment algorithm, which performs conflict resolution. The final group of detected tracks is an optimized set of nonconflicting tracks that survive track assignment.

Track extension

There are two basic methods of combining or updating tracks with new observations: Multiple Hypothesis Tracking (MHT) and Joint Probabilistic Data Association (JPDA) [5–9].

- MHT updates each track separately with each feasible new observation, creating several conflicting extensions for each old track, and decides subsequently which of these are the correct extensions.
- JPDA updates each track with a weighted sum of all feasible new observations.

Each approach in its optimal form uses a formidable amount of processor power and memory, and the processing required to compute these weights or probabilities increases exponentially with the number of closely spaced launches.

The track extension methods we generally use are suboptimal forms of MHT that strive to approach its performance with a fraction of the processing resources required. This requires that for each feasible extension a score be computed that measures how well the observations of each old track and each new observation are consistent with the motion of a single missile. This score is based on a least-squares fit to a model of missile motion. The model can range from polynomial fitting (position, velocity, and acceleration in each of two or three dimensions) to a data base of *a priori*, spin-compensated profiles of missile motion.

After these scores have been computed, an assignment algorithm decides which set of nonconflicting track extensions from this list is most likely to be correct. Depending on the system, the method used can range from a simple "greedy" algorithm to a fast optimal two-dimensional assignment algorithm [10, 11].

Track segment fusion

The general track segment fusion problem is as follows. Given a number of newly detected and historical tracks, determine which of the new tracks actually represent extensions of previously detected tracks and which are truly new detections. The problem can arise because an established track is sometimes prematurely terminated. This can occur when a missile becomes too dim to observe for some time, or when the observations are corrupted (by closely spaced multiple launches, plume persistence, or background clutter), or when a missile leaves the view of one satellite and/or enters the view of another. For some systems there is also the problem of correlating tracks that originate from the tankage of one stage of a missile as it separates from the subsequent stage.

The track segment fusion problem occurs in the following tracking contexts:

- New multisensor track / old multisensor track.
- New monocular track / old multisensor track.
- New multisensor track / old monocular track.
- New monocular track / old monocular track.

In each context, the problem is solved by computing a goodness-of-fit score for each pair of new and terminated detections. This score measures the degree to which the track pair exhibits consistent motion for a single missile.

When a matrix of such scores for all such pairs has been computed, a two-dimensional track-assignment algorithm is used to determine the "best" set of nonconflicting track pairs. Pairs which have sufficiently small scores determine which newly detected tracks are probable extensions of previously established tracks.

In determining these scores, it is not necessary or even desirable to use all the returns in each track segment. Usually all that is needed is the last two or three returns per sensor from the old segment in combination with the first two or three returns per sensor from the new segment.

In all contexts above that involve more than one sensor in the combined track, the scores are determined from the generalized analytic tracker filter. When only one sensor is involved in the combined track, the scores are determined from a monocular version of the generalized analytic tracker that is not described in this paper. In unusual cases, where the time gap is large between the last return of the old track segment and the first return of the new track segment, it is necessary to determine the score using a profile-dependent filter.

Sensor fusion

In the context of this discussion, sensor fusion involves the coordinated and simultaneous use of multiple satellites for missile detection, tracking, and estimation of launch parameters. There are three basic methods of sensor fusion described in [5, 6]: sensor-level tracking, central-level fusion, and hybrid fusion.

Sensor-level tracking. Tracks are formed independently within each sensor's data stream, and final decisions are made regarding which observations are assigned to which tracks; these tracks' state vectors are sent to a central location where the monocular tracks are combined (or correlated) to form central-level, multisensor tracks.

This approach can work well in limited-threat scenarios, particularly when the observations that pertain to each monocular track are communicated as part of the track information. Sometimes this is the best approach available when there are limitations in communication bandwidth or processor throughput.

However, an implicit assumption in this approach is that the monocular tracking process performs almost flawlessly for each sensor. Otherwise, when incorrect returns are included in a detected monocular track, that track may correlate very poorly with one or more companion tracks from other sensors which actually represent the same missile. This miscorrelation often occurs when this method is used in a densely spaced multiple-launch environment, and it often results in track assignment errors and in inaccurate estimates of launch parameters.

Central-level fusion Individual observations from multiple sensors are brought to a central location before decisions are made regarding which belong to missiles and which are clutter; the individual observations are then used to form central-level tracks.

Central-level fusion has more demanding requirements for processor loading and communication bandwidth; however, it is also more accurate than sensor-level fusion in general applications [12].

Hybrid fusion Sensor-level tracks are formed to determine which observations belong to tracks vs. clutter, and the observations associated with these tracks are sent independently to a central location where central-level tracking is performed.

Many variations on these three themes can be implemented [13]. One elaboration is to form all "feasible" monocular tracks for each sensor individually, communicate them all to the central location, and perform the final track correlation process at that location. This requires enough communication bandwidth to send all sensor observations which constitute these feasible tracks, but it does not require sending all sensor observations (e.g., from sensor noise and background clutter).

Historically we have used all of these approaches. In recent years we have used either central-level fusion or some variation of hybrid fusion. For systems that permit ground-based central-level processing, our general direction is toward central-level fusion. However, for systems that require the central-level processing to be space-based, limitations in communications bandwidth, processing throughput, and memory usually require a hybrid approach.

The ultimate concept in central-level fusion, for which only one observation per satellite is required, is discussed in [14]. Our claim for it today is limited to the statement that tactical parameters can be estimated surprisingly well using only one observation per satellite. However, we expect missile detection and tracking at this level to become practical in the era of massively parallel processing.

Space-based sensor fusion

There are surveillance programs that concentrate on the space-based aspects of surveillance and tracking applications. Both the individual sensor tracking in two dimensions and the three-dimensional sensor fusion tracking applications are space-based rather than ground-based. This situation brings unique constraints to the tracking and sensor fusion problems.

The space-based bandwidth and memory limitations generally rule out a central-level fusion approach. Sensor-level fusion requires the least amount of processing power and communication bandwidth; however, the individual track returns are not in a central location for other functions such as launch parameter estimation. Sensor-level fusion is also not as accurate as central-level fusion in general applications [12]. Hybrid fusion was chosen because the individual returns can be used at the central location for other system functions, and because in many circumstances hybrid fusion can approach the accuracy of central-level fusion.

Thus, two-dimensional tracking is performed in a processor that is collocated with the corresponding sensor. This processor then sends the returns that form the minimum detected tracks or that extend ongoing two-dimensional tracks to a central location to be combined with observations from other sensors into three-dimensional tracks.

The main factors in the choice of tracking algorithms pertain to system requirements. When single-sensor and multiple-sensor tracking are performed in a space-based processor, algorithm speed and simplicity are very important. For this reason, track extension in both two and three dimensions is performed recursively when a new scan or frame of observations is received via a Kalman filter, which requires relatively few computational resources.

The Kalman filter [15, 16] is a recursive minimum-variance estimator often used in target tracking [5–9]. The observation $z(k)$ at time k relates to the state of the true target by

$$z(k) = \mathbf{H}x(k) + w(k), \tag{1}$$

where $\mathbf{H}$ is the mapping of the state space onto the observation space, $x(k)$ is the state of the target (which can include position, velocity, and acceleration), and $w(k)$ is a zero mean Gaussian random noise with covariance matrix $\mathbf{R}$. The assumed model of motion is given by the process equation

$$x(k+1) = \Phi x(k) + n(k), \tag{2}$$

where Φ is the motion model from time k to time $k + 1$ and $n(k)$ is a zero mean Gaussian random variable (uncertainty in the model of motion) with covariance matrix $\mathbf{Q}$. The recursive estimate of the state at time k given that k observations have been received, $x(k|k)$, is given by

$$x(k|k) = \Phi x(k-1|k-1) + \mathbf{K}(k)v(k), \tag{3}$$

$$v(k) = z(k) - \mathbf{H}\Phi x(k-1|k-1), \tag{4}$$

where the Kalman gain matrix is given by

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^{\mathrm{T}}[\mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^{\mathrm{T}} + \mathbf{R}]^{-1}, \tag{5}$$

where the superior T indicates a matrix transpose, and the covariance matrix $\mathbf{P}$ of the state is given by

$$\mathbf{P}(k|k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}]\mathbf{P}(k|k-1), \tag{6}$$

$$\mathbf{P}(k|k-1) = \Phi\mathbf{P}(k-1|k-1)\Phi^{\mathrm{T}} + \mathbf{Q}. \tag{7}$$

The sensor revisit rates in the space-based programs are fast enough that a simple constant acceleration model of motion can be used for both the two-dimensional and three-dimensional track filtering.

Mistakes in two-dimensional hit-to-track assignments during track extension can result in poor tracking, since the Kalman filter in Equations (3) and (4) assumes that the observation updating of the state vector is from the target in track. A simple and effective track-monitoring algorithm can recognize these mistakes [17].

This track-monitoring algorithm for two-dimensional tracks examines the track residual or the difference in the assigned observation and predicted position given in Equation (4). The sequence of residuals is a Gaussian random sequence if the tracking system is working well. The monitoring algorithm is given by

$$d(k) = \sum_{i=k-M}^{k} [z(i) - \mathbf{H}\Phi x(i - 1/i - 1)]^{\mathrm{T}}\mathbf{S}_i^{-1} \times [z(i) - \mathbf{H}\Phi x(i - 1/i - 1)], \tag{8}$$

where M is the window of frames of observations monitored and $\mathbf{S}$ is the covariance matrix of the residual calculated in the Kalman gain formula in Equation (5) as the part inside the brackets. The quantity $d(k)$ is a chi-squared random variable.

A simple threshold can be set to determine whether the residual sequence is not chi-squared and the tracking system is not functioning correctly [17]. The residual sequence, however, can only indicate when the tracking system is no longer functioning properly, not the reason for the improper function. The residual sequence has been used to indicate sensor failure [18] and incorrect track-to-observation assignments [19] and target maneuver [20]. When an additional indicator is added to this monitoring scheme in three dimensions, the reason for poor tracking can be determined.

Fusion center operations

As mentioned earlier, the hybrid model of sensor fusion is used in space-based tracking. This means that the individual sensors are forming and updating their two-dimensional tracks and sending the observations assigned to those tracks to the fusion center to update the three-dimensional tracks.

The first function in the fusion center is to perform track-to-track association so that a three-dimensional track can be formed. This is accomplished by scoring the reasonable two-dimensional track pairs and using an assignment algorithm to choose a noncontending set [21]. After a three-dimensional track is formed, the Kalman filter is used to update the three-dimensional state vector whenever a newly assigned observation for one of the two-dimensional tracks (used to form this three-dimensional track) is received.

Poor tracking can result from sensor failure, incorrect two-dimensional track-to-observation assignments, target maneuver, and incorrect two-dimensional track-to-track assignments. A method to approximate the effects of target maneuvers in the pitch direction has been developed.[2] The track-to-track assignment problem is better known as ghost tracking [22]. The residual monitoring by itself cannot distinguish these problems. An expanded track-monitoring system presented in [17] can distinguish among these problems (excluding sensor failure, which is assumed to be detected in the front-end signal processing). The newly monitored feature is the hinge or inclination angle (see Appendix A).

The additional monitoring is accomplished through

$$i(k) = \sum_{i=k-M}^{k} \frac{[I_1(i) - I_2(i)]^2}{\sigma_{I_1}^2 + \sigma_{I_2}^2}, \tag{9}$$

where $I_1(i)$ is the inclination angle of the hit from sensor 1 at time i and $\sigma_{I_1}^2$ is the variance of $I_1(i)$ given by

$$\sigma_{I_1}^2 = \hat{u}_{I_1}^{\mathrm{T}} \mathbf{R} \hat{u}_{I_1}, \tag{10}$$

where $\hat{u}_{I_1}$ is the unit vector in angular units from the sensor in the inclination angle direction. The quantity $i(k)$ can be shown to be a noncentral chi-squared random variable [17], and a simple threshold test can determine when $i(k)$ is not correct.

The monitoring system for three-dimensional tracking is given by the following procedure:

1. Monitor the filter residuals using Equation (8).
2. Monitor the inclination angles using Equation (9).
3. If both $d(k)$ and $i(k)$ pass their thresholds, the tracks cannot be declared poor.
4. If $d(k)$ fails its threshold and $i(k)$ passes, the target is maneuvering, indicating that the model of motion in Equation (2) is incorrect and that the filter needs to be adapted to follow that maneuver.
5. If $i(k)$ fails its threshold and $d(k)$ passes, there is an error in the track-to-track association and a ghost target is in track.
6. If both quantities fail their thresholds, an incorrect observation-to-track association has occurred on one of the two-dimensional tracks used to form this three-dimensional track. The two-dimensional track monitoring system is now examined to determine which two-dimensional track is poor.

These distinctions are possible because the inclination angle monitoring system utilizes only the observations (or interpolated observations if they are not time-synchronized), while the residual monitoring uses both the observations and the assumed model of motion. This monitoring is fast and can be used to find poor tracks. The poor tracks can then be terminated and correct tracks reinitiated for good tracking performance. An analytical evaluation of the effectiveness of this monitoring system can be found in [17]. This fast track-monitoring method was developed to increase the effectiveness of simple tracking algorithms in a processing-limited space environment.

Recalibration based on missile observations

For systems that feature regular and precise calibration from earth-based sources and stars, and where line-of-sight monitoring is performed regularly to ensure that the bias errors remain well determined, it may become possible to ensure that the line-of-sight bias errors are reliably smaller than the random line-of-sight measurement errors. When this can be assumed, tests on filter scores and on the value of the coplanarity residual can be used very powerfully to discriminate against incorrect multisensor tracks. Any such track that exhibits an unreasonable filter score, an unreasonable triangulated altitude, an unreasonable velocity vector, or a moderately large coplanarity residual could be dismissed as representing something other than a sequence of observations from a single missile.

However, the tracking performance in a multisensor processing environment can degrade rapidly as the unknown intersatellite line-of-sight bias errors increase. In particular, the tolerances used in the coplanarity test must be loosened. If absolute coordinates are used in the detection filters, the goodness-of-fit scores degrade, and the thresholds on those scores must also be loosened in order to maintain the specified probability of detection for tracks representing actual missiles. But this also allows more incorrectly formed tracks to survive and to corrupt the results. This could have catastrophic implications in a closely spaced multiple-launch scenario.

An approach we have developed to alleviate this problem for multiple-launch environments is to use the first few detected missiles themselves as a calibration source, estimate the measurable biases (i.e., the coplanarity residuals) with an estimation filter, and then restart the tracking process from the time at which the first missile was detected.

In conjunction with filters that use the average reference method to tolerate large line-of-sight bias errors, this is a good remedial procedure. Day-to-day processing proceeds with a relatively loose (but not unreasonably large) coplanarity tolerance, until a few selected missiles have been detected. In a dense launch scenario it is important to

[2] J. T. Krafcik, "Simple Approximate Method for Determining the Range–Altitude History for Ballistic Missiles," Foreign Aerospace Science and Technology Center, Dayton, OH, October 1991 (Secret).

select tracks for this purpose that are among the easiest to separate.

The measured coplanarity residuals for these selected missile detections are used to estimate the intersatellite line-of-sight bias errors for each pair of observing satellites and for the system as a whole. Track processing is then restarted, and the computations involving triangulation and coplanarity residual take into account these newly estimated biases. The coplanarity tolerance is tightened, and the result is often a significant improvement in overall tracking performance.

Concluding remarks

Satellite technology has provided the capability to achieve continuous global missile surveillance using infrared sensors. The IBM Federal Systems Company has developed key tracking and estimation algorithms in use by current surveillance systems, contributing in this area for more than 25 years.

We have described some of the key aspects of tracking algorithms for the IR surveillance and tracking of ballistic missile launches, several of which have not previously been described in the technical literature.

A key challenge in surveillance systems is developing algorithms that are tolerant of residual measurement bias errors, particularly in the context of simultaneous coordinated processing of data from multiple sensors. Two methods that we have found useful in that regard are the average reference method and recalibration using selected missile observations.

We have described alternative tracking algorithms for space-based processing, where computational resources are more constrained; we have shown methods used to accommodate missile trajectory variations such as staging and maneuvering; and we have described the use of *a priori* missile profiles, specifying which processing functions require them.

We anticipate that as numerically intensive computing capabilities increase, surveillance and tracking solutions will exploit more complete physical modeling, more multiple hypothesis testing, and in general more thorough processing of all collected sensor data. We also expect increasing requirements to combine sensor data from IR surveillance systems with sensor data from surveillance systems that make use of other observables. We expect that the capabilities of all of these applications will increase considerably in the era of massively parallel processing.

Acknowledgments

We acknowledge the significant and closely related contributions of the algorithm development team of the IBM Federal Systems Company at Boulder, particularly the contributions by E. Kennedy, N. Hansen, J. Sawyer, and M. Munger. In addition, we acknowledge the past contributions of G. Weiss of IBM FSC at Santa Clara.

Appendix A—Definitions

- *Boresight*

The alignment of the telescope or focal plane with the vehicle axis used for attitude determination. Boresighting is generally accomplished using simultaneous star measurements collected via an attitude determination sensor and an IR telescope.

- *Return*

An observation; generally contains time, intensity, and two-dimensional line-of-sight information, either in mission reference coordinates or in a reference frame that is aligned with the focal plane.

- *Representative return*

A single centroided return that is computed from a group of raw returns that are closely spaced in time and position. Assuming that these returns originated from a single missile, the time, intensity, and two-dimensional position of this return are taken to "represent" the missile position at the representative time.

- *Inclination angle*

The angle between the plane containing (S_1, S_2, and V_1) and the plane containing (S_1, S_2, and 0), where

- S_1 and S_2 are the position vectors for two satellites at a common time T.
- V_1 is a line-of-sight vector emanating from S_1 at time T.
- 0 is the position of the center of the earth.

The inclination angle is depicted in **Figure 7**. Note that it is *not* generally the angle between line V_1 and the plane containing (S_1, S_2, and 0).

- *Coplanarity residual*

The angle between the plane containing (S_1, S_2, and V_1) and the plane containing (S_1, S_2, and V_2), where

- S_1 and S_2 are the position vectors for two satellites at a common time T.
- V_1 is a line-of-sight vector emanating from S_1 at time T.
- V_2 is a line-of-sight vector emanating from S_2 at time T.

If S_1 and S_2 are given, and if the distance of closest approach of lines V_1 and V_2 is small, the coplanarity residual is approximately proportional to this miss distance, and it is almost identical to the difference in inclination angles computed from lines V_1 and V_2 and the common satellite positions.

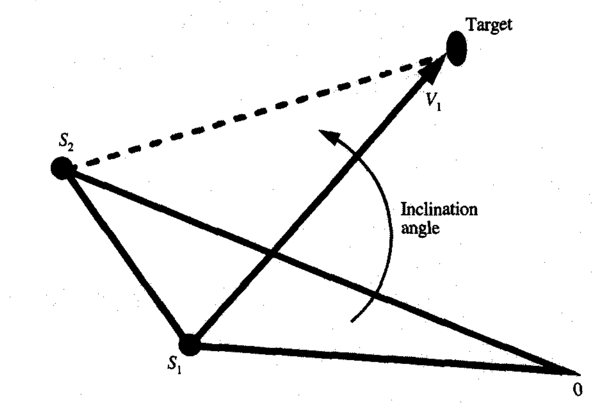


Figure 7

Inclination angle geometry.

• *Yaw*
The horizontal angle between velocity and acceleration vectors.

• *Pitch*
The vertical angle between velocity and acceleration vectors, in the absence of gravitational effects.

• *Mission reference frame (MR)*
An east-north-up coordinate system normally centered at the satellite; more precisely, it is a coordinate system with the x axis pointing east, the y axis pointing north, and the z axis pointing from earth center toward the satellite.

• *Local horizontal frame (LH)*
An east-north-up coordinate system normally centered at the estimated position or launch point of a target. The orientation of the LH frame is defined in precisely the same way as that of the MR frame, except that the "up" axis is defined as pointing toward local vertical rather than from earth center through the target.

• *Line-of-sight reference frame (LOS)*
A reference frame which is centered at the estimated position of an observed event. The z axis points from the event toward the satellite center, the x axis points along the positive (clockwise) azimuth direction, and the y axis points along the positive elevation direction, away from earth center. (That is, when the event is located directly north of the satellite subpoint, the x axis points eastward and the y axis points northward.)

• *Trajectory reference frame (TR)*
A crossrange-downrange-up coordinate system centered at the estimated launch position of a target; when the target is launched precisely northward, this is identical to the LH frame.

• *Earth-centered fixed frame (ECF)*
An earth-centered coordinate frame whose x and y axes lie in the equatorial plane at respective longitudes of 0 and 90 degrees, and whose z axis points toward the North Pole. This coordinate frame rotates with the earth in inertial space and has a specific reference time or location.

Appendix B—Coordinate transformations

• *Transformation matrix from ECF to MR*
Let S be the satellite position vector (i.e., the vector from earth center to the satellite) in ECF coordinates. Let

$$U_p = \frac{S}{|S|}.$$

Define the *north pole vector* to be $N_p = (0, 0, 1)$. Let

$$E_{ast} = \frac{N_p \times U_p}{|N_p \times U_p|}$$

and

$$N_{orth} = U_p \times E_{ast}.$$

The transformation matrix from ECF to MR is then given by

$$T_{\mathrm{ECF}}^{\mathrm{MR}} = \begin{bmatrix} E_{ast} \\ N_{orth} \\ U_p \end{bmatrix}.$$

• *Transformation matrix from ECF to LH*
Let P be the target position vector (that is, the vector from earth center to the target) in ECF coordinates. Let Q be the corresponding local vertical unit vector, and let

$$U_p = \frac{Q}{|Q|}.$$

From this point on, the definition of $T_{\mathrm{ECF}}^{\mathrm{LH}}$ is identical to that of $T_{\mathrm{ECF}}^{\mathrm{MR}}$ above.

• *Transformation matrix from MR to LOS*
Given a unit line-of-sight observation vector (x, y, z) in the MR frame, the transformation matrix from MR to LOS is given by

$$T_{\mathrm{MR}}^{\mathrm{LOS}} = \begin{bmatrix} \frac{y}{\sqrt{x^2+y^2}} & \frac{-x}{\sqrt{x^2+y^2}} & 0 \\ \frac{-zx}{\sqrt{x^2+y^2}} & \frac{-zy}{\sqrt{x^2+y^2}} & \sqrt{x^2+y^2} \\ -x & -y & -z \end{bmatrix}.$$

- *Transformation matrix from TR to LH*

Given α_{lp} as the target launch azimuth (measured clockwise from local north), the transformation matrix from TR to LH is given by

$$T_{\mathrm{TR}}^{\mathrm{LH}} = \begin{bmatrix} \cos\alpha_{\mathrm{lp}} & \sin\alpha_{\mathrm{lp}} & 0 \\ -\sin\alpha_{\mathrm{lp}} & \cos\alpha_{\mathrm{lp}} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Appendix C—APL2® implementation of the analytic tracker filter

Readers who understand least-squares filter theory but who are unfamiliar with APL2 are nevertheless encouraged to study this filter implementation, preferably in conjunction with [23]. Some of the APL2 expressions used in this implementation are described as follows:

```
⍝           Comment
C←A+.×B     Matrix multiplication
A×B         Componentwise product of the elements of arrays A and B
X←B⌹A       Solve the least-squares matrix equation Ax=b
⍉M          Transpose of matrix M
1 1⍉M       Diagonal of matrix M
H←(N,4)⍴0   Initialize H as an N-by-4 matrix with all zeros
N←↑⍴V       Number of elements in vector V or rows in matrix V
S←+/V       Vector sum or matrix sum across each row
S←W⋆.5      Square root
V[K]        Vector subscript (K may be an array of indices)
M[;K]       Matrix subscript (Kth column(s) of matrix M)
M[K;]       Matrix subscript (Kth row(s) of matrix M)
⍳N          The vector of integers 1..N
V+[1]M      Add vector V to every column of matrix M
```

- *Analytic tracker filter for N-satellite processing*

```
⍝ Inputs:
⍝ TREF      Reference time at which the missile states are estimated
⍝ XX        Initial estimated state vector, to be updated by the filter
⍝ PECF      Estimate of missile position in ECF at time TREF
⍝ S         Matrix of satellite positions at the times of the returns
⍝ JSAT      Index vector indicating which satellite observed each return
⍝ YGMT      Vector of times associated with the returns
⍝ X         Vector of x-values of the returns in the MR frame
⍝ Y         Vector of y-values of the returns in the MR frame
⍝ SIGX      Vector of standard deviations of the errors in x
⍝ SIGY      Vector of standard deviations of the errors in y

ZOUT←FILTER INPUT
(TREF XX PECF S JSAT YGMT X Y Z SIGX SIGY)←INPUT

N←↑⍴YGMT                ⍝ Number of returns
ΔT←YGMT-TREF            ⍝ Time of every return relative to reference time
SE←((X×X)+Y×Y)⋆.5       ⍝ Sine(elevation) for every return
Z←-(1-SE×SE)⋆.5         ⍝ Z-values of the returns in MR frame
SIG←SIGX,SIGY           ⍝ Sigma random errors for every return (x,y)
```

```
⍝ Compute components of ECF-to-MR transformation matrices
UP←S÷[1]MAG S
EAST←⍉(3,N)⍴(-UP[;2]),UP[;1],N⍴0
EAST←EAST÷[1]MAG EAST
XNOR←(UP[;2 3 1]×EAST[;3 1 2])-UP[;3 1 2]×EAST[;2 3 1]
⍝ Compute 2 components of MR-to-LOS transformation matrices
LX←⍉(3,N)⍴(Y÷SE),(-X÷SE),(N⍴0)
LY←⍉(3,N)⍴(-Z×X÷SE),(-Z×Y÷SE),SE

VOBSMR←⍉(3,N)⍴X,Y,Z                 ⍝ Observations in MR coordinates
VOBS←(+/LX×VOBSMR),(+/LY×VOBSMR)  ⍝ Convert to LOS coordinates
VOBS←JSAT AVEREF VOBS               ⍝ Apply Average Reference method

⍝ (Reader exercise:  What is the range of values that VOBS can take on?)
⍝ Compute 2 components of ECF-to-LOS transformation matrices
EX←(LX[;1]×[1]EAST)+(LX[;2]×[1]XNOR)+(LX[;3]×[1]UP)
EY←(LY[;1]×[1]EAST)+(LY[;2]×[1]XNOR)+(LY[;3]×[1]UP)

Q←4 4⍴0                            ⍝ Inverse square root of
(1 1⍉Q)←1÷0.05 .05 .05 0.01        ⍝  a-priori state covariance matrix
Δ←0.001                            ⍝ Finite-difference state perturbation
NITER←4                            ⍝ Number of filter iterations
ITER←0
L0:ITER←ITER+1   ⍝ --------------- Filter iteration loop ----------------------

VPRED←PREDAT(XX PECF S ΔT EX EY)  ⍝ Measurement prediction vector (LOS frame)
VPRED←JSAT AVEREF VPRED           ⍝ Apply Average Reference method
PHI←+/(VPRED÷SIG)*2               ⍝ Cost function
→(ITER=NITER)/EXIT                ⍝ Exit if last iteration

⍝ ---------------------------------------------------------------------------
H←((2×N),4)⍴0          ⍝ Compute gradient matrix H via finite differences
J←0
L1:J←J+1                           ⍝ -------- Loop over states --------------------
 XS←XX                                     ⍝ Obtain current state vector
 XS[J]←XX[J]+Δ                             ⍝ Perturb state J
 V←PREDAT(XS PECF S ΔT EX EY)              ⍝ Perturbed prediction vector (LOS frame)
 V←JSAT AVEREF V                           ⍝ Apply Average Reference method
 H[;J]←(V-VPRED)÷Δ                         ⍝ Jth column of gradient matrix
→(J<4)/L1                          ⍝ ------- End loop over states -----------------

⍝ ---------------------------------------------------------------------------
⍝ Following line implements Bierman's square root information filter (SRIF),
⍝  which is equivalent to a least-squares batch filter
XX←XX+(((VOBS-VPRED)÷SIG),0 0 0 0)⌹(H÷[1]SIG),[1]Q  ⍝ State correction
→(ITER<NITER)/L0   ⍝ -------  End filter iteration -------------------------
EXIT:ZOUT←XX,PHI      ⍝ Output the estimated states and the cost function
```

• ***Prediction function***

```
⍝ Uses estimated state vector to predict line-of-sight measurement vectors
⍝ in LOS coordinates
VPRED←PREDAT INPUT
(X PECF S ΔT EX EY)←INPUT

VECF←X[⍳3]                            ⍝ Missile velocity vector
AECF←X[4]×VECF                        ⍝ Missile thrust acceleration
AECF←AECF-(g÷2)×PECF÷MAG PECF         ⍝ Subtract gravitational acceleration
P←PECF+[2](ΔT∘.×VECF)+(ΔT×ΔT)∘.×AECF  ⍝ Predicted missile position at time
⍝                                          of every return in ECF coordinates
D←P-S                                 ⍝ Subtract satellite positions
D←D÷[1]MAG D                          ⍝ Normalize each line-of-sight vector
VPRED←(+/EX×D),(+/EY×D)               ⍝ Transform to LOS coordinates, retain
⍝                                           only x and y components
```

• *Average reference function*

```
 AVE←JSAT AVEREF V
 N←↑ρJSAT                     ⍝ Number of returns
 ISAT←0
 L0:ISAT←ISAT+1              ⍝ --- Loop over satellites ------
 I←(ISAT=JSAT)/⍳N            ⍝ Determine which returns were seen by this
 →(0=ρI)/L0E                 ⍝  satellite; if none, next satellite
 V[I]←V[I]-(+/V[I])÷ρI         ⍝ Subtract average x from each x element
 V[I+N]←V[I+N]-(+/V[I+N])÷ρI   ⍝ Subtract average y from each y element
L0E:→(ISAT<⌈/JSAT)/L0       ⍝ -----End loop over satellites ---
 AVE←V
```

• *Matrix and vector magnitude function*

```
Z←MAG V
Z←(+/V×V)*0.5
```

Appendix D—APL2 implementation of profile-dependent prediction

This subroutine of the four-state profile-dependent filter has a function directly analogous to that of the subroutine PREDAT in Appendix C, and it has analogous inputs and outputs.

```
VPRED←PREDPROF INPUT
(IAVE X Δ YGMT JSAT S EX EY)←INPUT

(XGMT AZL TLAT TLON)←X        ⍝ Launch time, azimuth, latitude, longitude
⍝                               Δ is a missile profile in TR frame
Q←TLAT UNIT TLON              ⍝ Local vertical unit vector at launch point
PLP←POSITION Q                ⍝ Unit position vector of launch point (ECF)
LH2E←⍉ECFTOMR Q               ⍝ LH-to-ECF transformation matrix
PLP←PLP×EARTHRAD PLP          ⍝ Launch position in ECF
⍝                               TR-to-LH transformation matrix
TR2LH←(3,3)ρ(2○AZL),(1○AZL),0,(-1○AZL),(2○AZL),0,0,0,1
TR2E←LH2E+.×TR2LH             ⍝ TR-to-ECF transformation matrix
NPR←(ρΔP)[2]                  ⍝ No. of seconds of data in the profile
T←YGMT-XGMT                   ⍝ Time since launch of each track return
M←1⌈(NPR-1)⌊⌊T                ⍝ Linearly interpolate into the profile
FR←T-M                        ⍝ for these times; obtain positions relative
ΔP←Δ[;M]+FR×[2]Δ[;M+1]-Δ[;M]  ⍝ to launch position in TR frame
P←PLP+[1]TR2E+.×ΔP            ⍝ Positions of target in ECF at time of returns
D←(⍉P)-S                      ⍝ Positions in ECF relative to sensor positions
D←D÷[1]MAG D                  ⍝ Unit LOS vectors to target in ECF
VPRED←(+/EX×D),(+/EY×D)       ⍝ Transform to LOS coordinates, retain only
→(IAVE=0)/0                   ⍝   2 dimensions
VPRED←AVEREF(VPRED JSAT)      ⍝ Apply Average Reference method if specified
```

• *Transformation from ECF to MR coordinates*

```
E2AP←ECFTOMR S;UP;EAST;NORTH
⍝ inputs:  S     satellite position vector
⍝ outputs: E2AP transformation matrix from ECF to MR coordinates

UP←S÷(+/S*2)*.5                   ⍝ UP is S normalized
EAST←0 0 1 CROSSPROD UP           ⍝ North pole vector cross UP
EAST←EAST÷(+/EAST*2)*.5           ⍝ Unit vector pointing east
NORTH←UP CROSSPROD EAST           ⍝ Unit vector pointing north
E2AP←3 3ρEAST,NORTH,UP            ⍝ Transformation matrix
```


References

1. A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, Inc., New York, 1970.

2. Lance D. Newman, "Target Scene Generation for IR Sensor Simulations," Technical Operating Report, The Aerospace Corporation, El Segundo, CA.
3. P. G. Kaminski, A. E. Bryson, and S. F. Schmidt, "Discrete Square Root Filtering; A Survey of Current Techniques," *IEEE Trans. Automatic Control* **AC-16,** 727–736 (1971).
4. G. J. Bierman, "Sequential Square Root Filtering and Smoothing of Discrete Linear Systems," *Automatica* **10,** 147–158 (1974).
5. *Multitarget-Multisensor Tracking: Advanced Applications, Vol. 1,* Yaakov Bar-Shalom, Ed., Artech House, Norwood, MA, 1990.
6. Edward Waltz and James Llinas, *Multisensor Data Fusion,* Artech House, Norwood, MA, 1990.
7. Samuel S. Blackman, *Multiple Target Tracking with Radar Applications,* Artech House, Dedham, MA, 1986.
8. Yaakov Bar-Shalom and Thomas E. Fortmann, *Tracking and Data Association,* Academic Press, Inc., Orlando, FL, 1988.
9. *Multitarget-Multisensor Tracking: Applications and Advances, Vol. 2,* Yaakov Bar-Shalom, Ed., Artech House, Norwood, MA, 1992.
10. O. E. Drummond, D. A. Castañon, and M. S. Bellovin, "Comparison of 2-D Assignment Algorithms for Sparse, Rectangular, Floating Point, Cost Matrices," *Proceedings of the SDI Panels on Tracking,* Dec. 1990, pp. 4-81–4-97.
11. David A. Castañon and M. S. Bellovin, "New Assignment Algorithms for Data Association," *Proceedings of SPIE on Signal & Data Processing of Small Targets,* Orlando, FL, April 20–22, 1992, Vol. 1698, pp. 313–323.
12. J. A. Roecker and C. D. McGillem, "Comparison of Two Sensor Tracking Methods Based on State Vector Fusion and Measurement Fusion," *IEEE Trans. Aerospace & Electron. Syst.* **AES-24,** 447–449 (1988).
13. M. A. Kovacich, O. E. Drummond, C. Bowman, and B. Fridling, "Tracking Algorithm Taxonomy: Further Developments," *Proceedings of the SDI Panels on Tracking,* December 1991, pp. 2-15–2-40.
14. J. G. Rudd and R. A. Marsh, "Single-Scan Tracking Using *N* Infrared Sensors," *Proceedings of SPIE on Signal & Data Processing of Small Targets,* Orlando, FL, April 20–22, 1992, Vol. 1698, pp. 394–401.
15. A. Gelb, *Applied Optimal Estimation,* The M.I.T. Press, Cambridge, MA, 1974.
16. H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part I,* John Wiley, New York, 1968.
17. J. A. Roecker, "Track Monitoring with Single and Multiple 2-D Passive Sensors," *Proceedings of SPIE on Signal & Data Processing of Small Targets,* Orlando, FL, April 20–22, 1992, Vol. 1698, pp. 416–424.
18. P. S. Maybeck, "Failure Detection Without Excessive Hardware Redundancy," *Proceedings of the IEEE National Aerospace & Electronic Conference,* Dayton, OH, May 1976, pp. 315–322.
19. T. W. Jeffery, "Track Quality Estimation for Multiple-Target Tracking Radars," *Proceedings of the IEEE 1989 National Radar Conference,* Dallas, TX, March 1989, pp. 76–79.
20. P. L. Bogler, "Tracking a Maneuvering Target Using Input Estimation," *IEEE Trans. Aerospace & Electron. Syst.* **AES-23,** 298–310 (1987).
21. T. J. Janssens, D. F. McAllister, and D. C. Pridmore-Brown, "Testing Track Initiation Algorithms that Fuse 2-D Tracks," *Proceedings of SPIE on Signal & Data Processing of Small Targets,* Orlando, FL, March 1989, Vol. 1096, pp. 284–297.
22. G. Van Keuk, "Deghosting in an Automatic Triangulation System," *Proceedings of the IEEE International Radar Conference,* October 1982, pp. 291–295.
23. J. A. Brown, S. Pakin, and R. P. Polivka, *APL2 at a Glance,* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988.

Received March 26, 1993; accepted for publication November 22, 1993

Jack G. Rudd *IBM Federal Systems Company, 6300 Diagonal Highway, Boulder, Colorado 80301 (JACKRUDD at BLDFVM9, jack_rudd@vnet.ibm.com).*

Mr. Rudd is a Senior Technical Staff Member at the Boulder facility of the IBM Federal Systems Company, working on algorithms for surveillance systems. He received a B.A. in mathematics (magna cum laude, Phi Beta Kappa) from the University of Iowa in 1964. Recently he has led algorithm development and prototyping activities in the areas of target detection and tracking, tactical parameter estimation, multisensor data fusion, and related areas of the Defense Support Program, which is the U.S. satellite early warning system. Related efforts include the development of high-fidelity simulations of surveillance focal plane arrays, and the development of algorithms for resolving closely spaced multiple events and tracking highly maneuvering events using simultaneous and coordinated processing of data from multiple satellites. He has extended several of these algorithms to systems with multiple bistatic radar sensors. Mr. Rudd received an IBM Outstanding Innovation Award in 1979 and an IBM Outstanding Technical Achievement Award in 1985. From 1990 to 1991, he was chairman of the IBM Federal System Division's Boulder Technical Advisory Board; he is currently a member of the IBM Federal Systems Company's Technical Advisory Board.

Richard A. Marsh *IBM Federal Systems Company, 6300 Diagonal Highway, Boulder, Colorado 80301 (rmarsh@bldfvm9.vnet.ibm.com).*

Mr. Marsh is a Senior Systems Engineer in the Advanced Programs Department at the Boulder facility of the IBM Federal Systems Company. He received a B.S. degree in mechanical engineering from the University of Denver and an M.S. degree in mechanical engineering from the University of Southern California. Mr. Marsh joined the Aerospace Corporation in 1968, working primarily on infrared surveillance systems. In 1984 he received an Aerospace Corporation Outstanding Accomplishment Award. In his last assignment there, he managed the Sensor Office of the Defense Support Program. In 1987 Mr. Marsh joined IBM, where he has worked on satellite data processing systems and advanced concepts. His current activities include infrared phenomenology, missions, and systems aspects. He is currently Program Manager for Advanced Technologies, working primarily with the Lawrence Livermore National Laboratories and the U.S. Air Force Space Command on methods to better utilize surveillance systems. Mr. Marsh has served as a member of the IBM Federal Systems Company's Technical Advisory Board since 1990.

James A. Roecker *IBM Federal Systems Company, 6300 Diagonal Highway, Boulder, Colorado 80301 (ROECKER at BLDFVM9, roecker@vnet.ibm.com).*

Dr. Roecker received a B.S.E.E. degree from the Rose–Hulman Institute of Technology in 1983, an M.S.E.E. from Purdue University in 1985, and a Ph.D. from Purdue in 1988. From 1983 to 1988 he was a teaching assistant at Purdue University, where he taught classes in random variables, stochastic processes, estimation, detection, digital filters, and modern communications. Since 1989 he has worked at the Boulder facility of IBM's Federal Systems Company. Dr. Roecker's primary areas of interest include estimation, detection, passive/active multiple sensor fusion, and multiple-target tracking. He has worked on space-based surveillance programs such as the Defense Support Program (DSP), the Follow on Early Warning System (FEWS) program, and the Brilliant Eyes (BE) program. He has also been active in related independent research and development tasks. Dr. Roecker is a member of the IEEE Aerospace and Electronics System Society and the SPIE.