J.L. Gertz

Multisensor Surveillance for Improved Aircraft Tracking

Cross-range measurements of aircraft travelling at distances of 50 to 200 miles include significant errors. Therefore, heading estimates for medium-to-long-range aircraft are not sufficiently accurate to be useful in conflict-detection predictions. Accurate cross-range measurements can be made—by using two or more sensors to measure aircraft position—but such measurements must compensate for the effects of system biases and aircraft turns. A set of algorithms has been developed that are resistant to system biases, that detect turns, and that track successfully through both biases and turns. These algorithms can be incorporated into a complete multisensor system, with good intersensor correlation of aircraft tracks and no added delays to the air traffic control processing chain.

Single-Sensor Tracking

The Mode S sensor has brought major improvements to the accuracy of aircraft surveillance reports [1]. With this sensor, the expected measurement noise has a standard deviation (σ) value of 25 feet in range by 1 milliradian in azimuth (if the constant bias offsets are ignored). However, this high degree of accuracy still introduces considerable error in the measured cross-range (position component in the azimuth direction). The 1-mrad noise in azimuth translates into a cross-range σ many times larger than the range σ for distant targets.

For a range measurement with a standard deviation of 25', for example, the standard deviation of the cross-range can be fifty times greater:

Range of target	1-σ range	1-σ cross-range
10 nmi	25'	60'
50 nmi	25'	300'
100 nmi	25'	600'
200 nmi	25'	1,200'

Large cross-range errors cause very noisy tracker inputs. The sample data of Fig. 1 illustrate the severity of the problem. The data produced with the current, single-sensor system contain a great deal of noise. To produce stable heading estimates, therefore, the data must be heavily smoothed by a Kalman filter. But such a tracker follows turns very poorly—initial turn reports are assumed to be measurement noise and, even after a turn has been identified, the true turn rate cannot be accurately estimated.

Figure 2 shows how a typical linear Kalman filter handles turning data. Note how severely the predicted aircraft heading lags the true heading.

These tracking problems are exacerbated when the aircraft enters a sensor diffraction zone. In such a zone, the radar signal is dif-

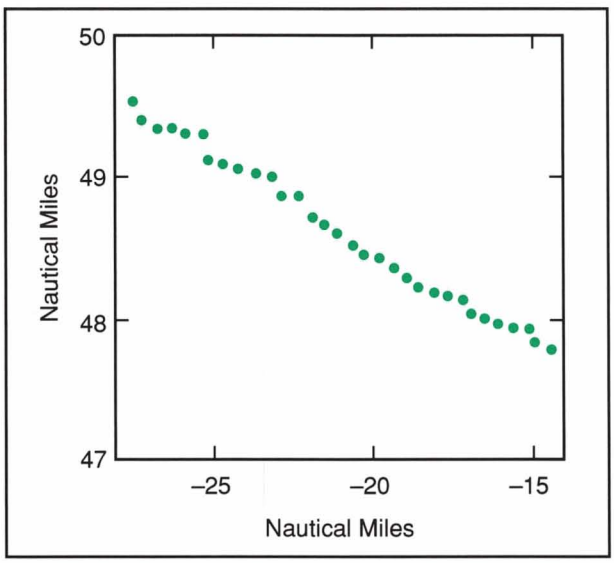


Fig. 1—The current single-sensor surveillance system produces extremely noisy data.

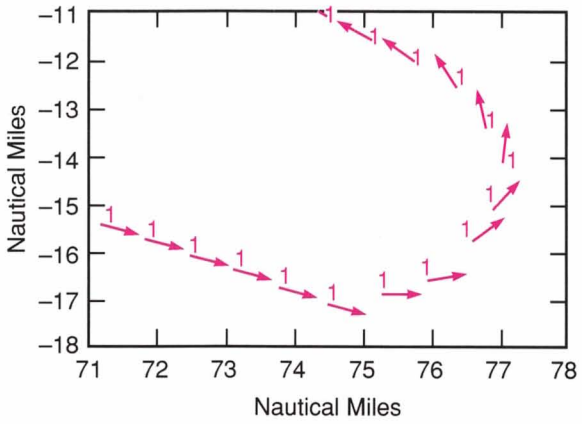


Fig. 2—Kalman filter performance during a turn. The "1"s show the true aircraft position. Performance of the filter is poor: the head of each arrow should touch the tail of the succeeding one.

fracted by a narrow object (such as a smokestack). The resulting beam curvature introduces very large errors in the azimuth estimates. In the worst case, in fact, the azimuth value read by the sensor is randomly distributed over the full antenna beamwidth.

In a real-life application, diffraction errors can be dangerously misleading. The results of the conflict-detection experiment shown in Fig. 3 clearly demonstrate the hazard of diffraction errors. The measured positions and headings look safe; the actual trajectories are heading for collision.

Since aircraft are typically observed by two or more sensors, it should be possible to use multisensor measurements to improve surveillance and tracking. The current air traffic control system makes use of multiple-sensor coverage, but only as a backup mode when primary-sensor coverage is missing. That is, the current mosaicking system switches its data source from preferred to supplemental only when required; only one sensor's reports are used on any given scan, and other available data are ignored.

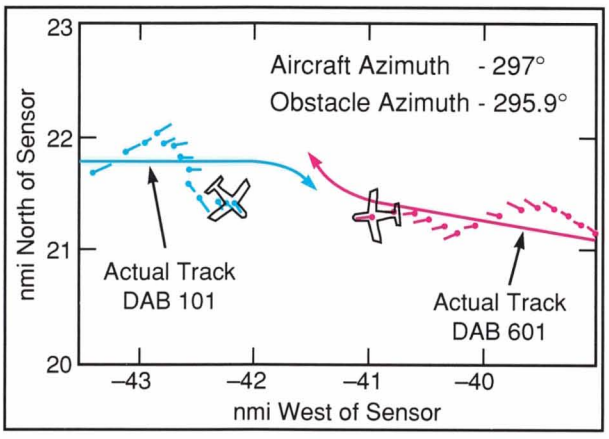


Fig. 3—Surveillance degradation due to diffraction.

A true multisensor system would provide far better surveillance and tracking of aircraft than the current system. For this reason, an ongoing program at Lincoln Laboratory has developed a set of algorithms and has implemented them in a system that offers multisensor processing of aircraft reports.

Multiple-Sensor Tracking Approaches

Multiple-sensor tracking can be implemented in one of two different ways. The simplest method is to feed unmodified reports from the sensors in time order into a common Kalman filter. Each report then improves the tracker covariance matrix (the matrix of standard deviations of all pairs of measured variables) along its range coordinate. If the sensors are at different aspect angles relative to the aircraft, the entire covariance matrix will tighten and good tracking will result.

The second approach is multilateration. In this method, illustrated in Fig. 4, range measurements from the sensors are time aligned to a common time, and position is determined by the intersection of the range arcs. Assuming different aspect angles, the measurement error ellipse becomes nearly circular. The improved reports are then fed into a Kalman filter.

Multilateration offers four advantages:

(1) improved accuracy in measurements shown on the controller display,
(2) quicker detection of aircraft turns,
(3) no diffraction errors, and
(4) estimates of aircraft altitudes.

For these reasons, Lincoln Laboratory has pursued most strongly the multilateration approach.

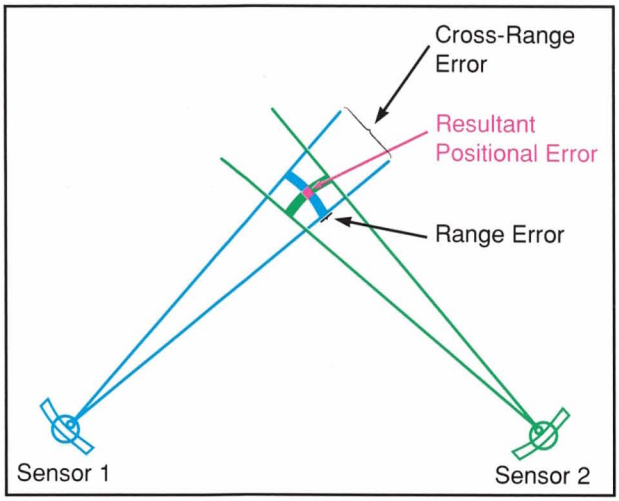
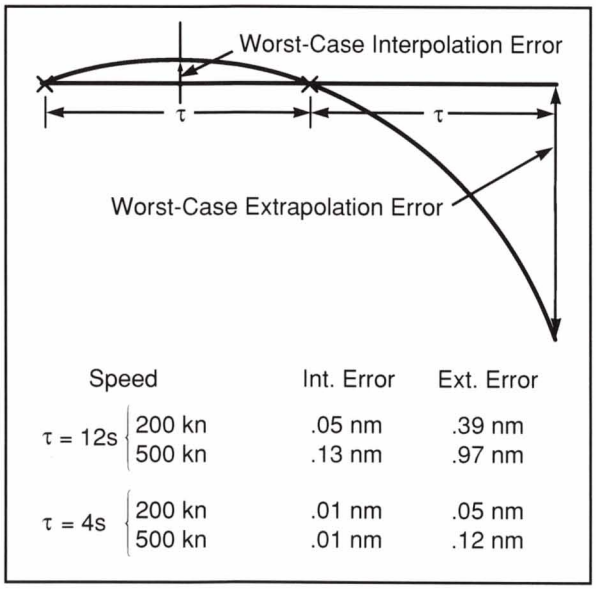


Fig. 4—Multilateration error ellipse.



Speed		Int. Error	Ext. Error
τ = 12s	200 kn	.05 nm	.39 nm
	500 kn	.13 nm	.97 nm
τ = 4s	200 kn	.01 nm	.05 nm
	500 kn	.01 nm	.12 nm

Fig. 5—Interpolation and extrapolation accuracy during 3°/s turn.

Time Alignment for Multilateration

Both interpolation (estimating between two data points) and extrapolation (estimating beyond the last data point) are used to time-align sensor measurements. Interpolation is always more accurate than extrapolation.

Interpolation and extrapolation are both accurate when aircraft are flying straight, but linear extrapolation has severe error potential during aircraft turns. Figure 5 shows an example of an error due to linear extrapolation during a turn, an error so large that it nullifies the improvements of multilateration. Thus turns must be identified prior to time alignment, and circular interpolation and extrapolation must then be used during turning periods.

Turn Detection

Turns are usually detected by observing a heading change in the aircraft's trajectory. However, the heading measurement depends upon the azimuth, and severe noise in the measurement of the azimuth is the reason that multisensor surveillance is needed. Thus the turn-detection technique for multilateration must use only measured ranges. The most successful such technique uses the second difference of successive range measurements, as shown in Fig. 6. The actual observed $\Delta(\Delta\rho)$ is

$$\Delta(\Delta\rho)_n = (\rho_n - \rho_{n-1}) - (\rho_{n-1} - \rho_{n-2})$$

where n is current scan. This value is compared with the predicted one for a straight trajectory. If the difference exceeds a predefined parameter, a turn is declared to be in progress.

This technique was applied to a sample of real data; the results are shown in Fig. 7. Note that true turns were detected, but a false apparent

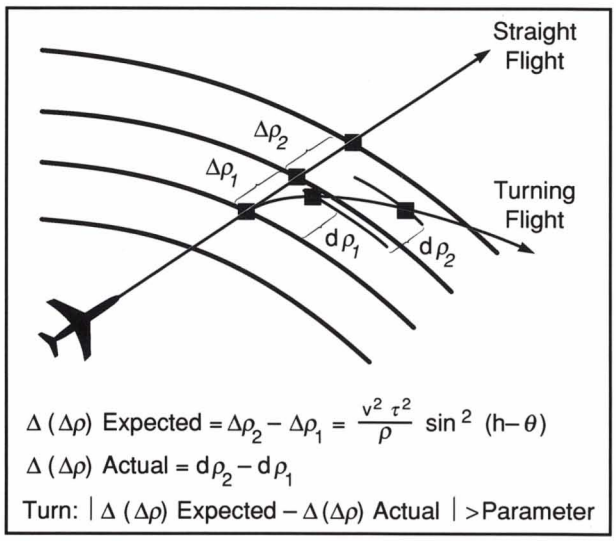


Fig. 6—Turn-detection algorithm.

turn caused by noise did not trigger the detector.

This technique cannot detect all turns. In particular, it can be insensitive to radial turns. However, turn detection is needed only to produce accurate range extrapolations. The turns that this approach misses are just the ones in which straight and circular extrapolation produce nearly the same result, and thus no harm results.

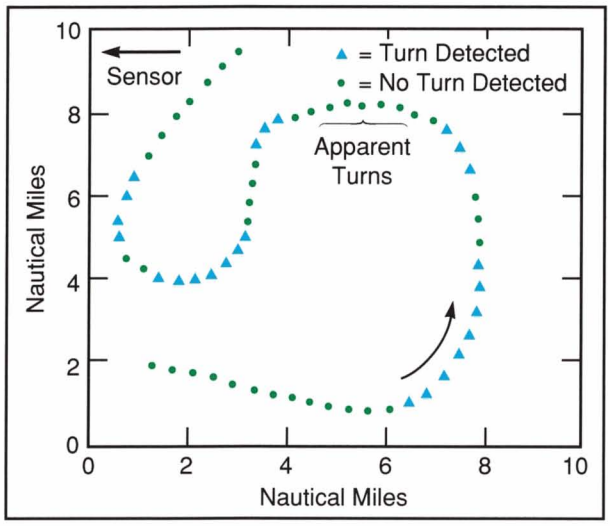


Fig. 7—Single-sensor range-only turn detection.

Coordinate Transformations

For two reasons, multisensor processing requires very accurate coordinate transformations. First, if the preferred-sensor report is absent, reports from supplemental sensors must be transformed into the primary-sensor coordinate system. Second, the distance from one sensor to another is the base of the computation triangle for multilateration. Figure 8 illustrates these two tasks.

Since the transformation procedure requires knowledge of the earth's radius, it must be known exactly. The earth is not a true sphere; its radius varies with latitude. Most coordinate-transformation procedures apply the spherical-

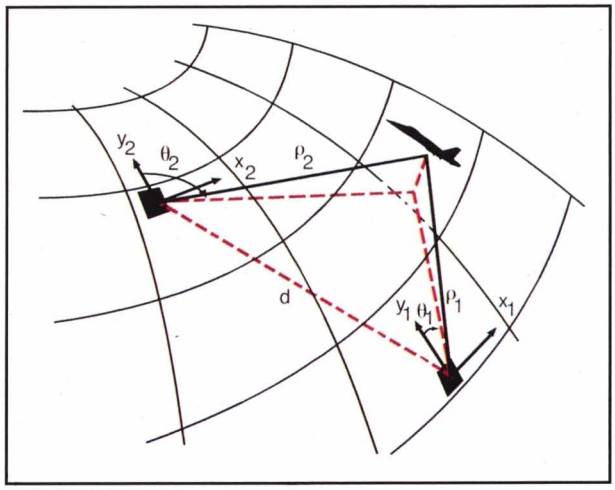


Fig. 8—Coordinate-transformation cases: (1) Conversion from ρ_1, θ_1 to ρ_2, θ_2; (2) Determination of d *for multilateration.*

earth transformation formulas, using the average earth's radius over the region of interest. Our studies, shown in Table 1, have shown that this approach is incorrect. The proper radius to use in the spherical equations is the local *radius of curvature* of the earth. This quantity matches the earth surface shape rather than its distance to the true earth center and is largest at the poles. By contrast, the local earth radius is largest at the equator.

Accuracy can be further increased by performing multiple transformations. In Fig. 9 the measurement from a distant sensor is transformed step by step at a constant longitude (variable latitude) to the latitude of the near sensor, and then transformed in one longitude step over to the near sensor location. Each step uses the local radius of curvature applicable to the latitude at the center of the step.

Mathematically, the multiple steps are a multiple convolution of the transformation equations. Thus a single transformation matrix can be defined for each sensor pair that describes the overall result in one step, as illustrated by the equation in Fig. 9. In actual application, run-time conversions are always single operations, independent of the number of steps employed. The 11-step transformation shown in Table 1 demonstrates the success of this ap-

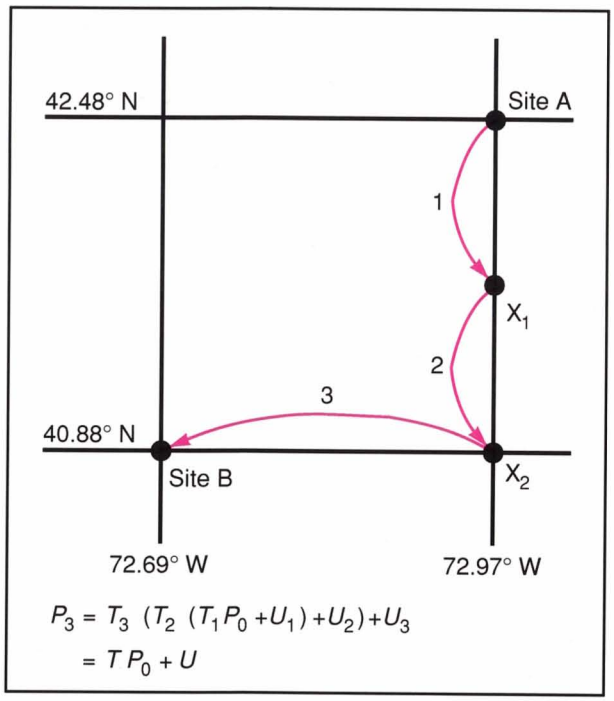


Fig. 9—Three-step coordinate transformation.

proach in reducing the coordinate-transformation error to essentially zero.

System Biases

An accurate cross-range measurement system must compensate for the effects of both turns and biases. Now that we have addressed the problem of turn recognition, we must take a look at the built-in biases of the system.

All sensors have intrinsic sources of measurement bias. The measured aircraft range depends on delays in the sensor's electronics, in clock errors, in signal refraction, and by the transponder turnaround delay error. The aircraft azimuth can be corrupted by north-mark error, antenna tilting, or signal diffraction. In a single-sensor system, only the transponder delay causes a relative error between two aircraft positions; the rest of the errors simply shift the positions of all aircraft equally and therefore do not affect separation monitoring. Since the transponder bias only affects range and is always small, system biases are of no consequence in a single-sensor system.

The situation changes markedly when multisensor data are used. First, a new class of biases appear: sensor-location errors and other errors that directly produce intersensor registration errors. Second, the biases of different sensors will produce different results, so that when different sources supply data for two aircraft, the relative aircraft separation is compromised. Finally, the transponder-delay bias has a greater effect when multilateration is used. Range errors are now transformed directly into azimuth errors, as shown in Fig. 10.

The graph in Fig. 11 illustrates the effect of these biases. These measurements were taken from the current ATC mosaicking system—the hops due to changes in the data source are so

Table 1. Location after Transformation from Site A to Site B Coordinates

	11 Steps	*1 Step*
Local-Radius Method	Δ = .3993 nmi	Δ = .3993 nmi
Radius-of-Curvature Method	Δ = .0002 nmi	Δ = .1823 nmi

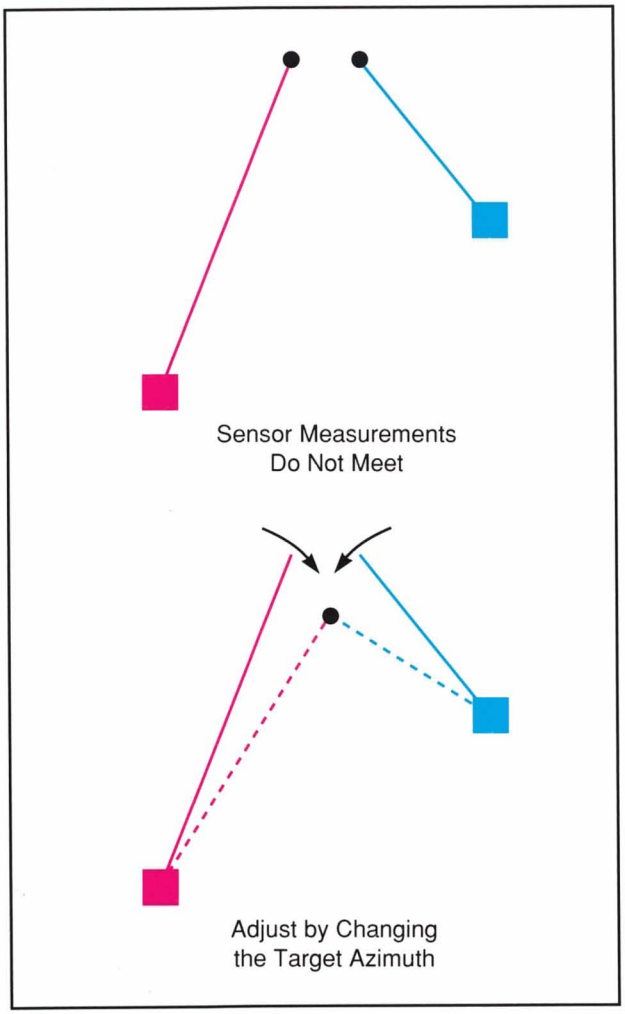


Fig. 10—Multilateration resolution of range biases.

large that aircraft tracking has become nearly impossible.

Procedures to remove registration biases by fitting large quantities of aircraft data to bias models have been developed. However, no such procedure can remove all bias effects, because not all biases can be modeled. Figure 12 presents data from one system after bias removal. The three sensors clearly give different results for the single aircraft.

Biases in a multilateration system can easily produce aircraft azimuths less accurate than those of the original single-sensor system. Thus a multilateration algorithm must be bias resistant.

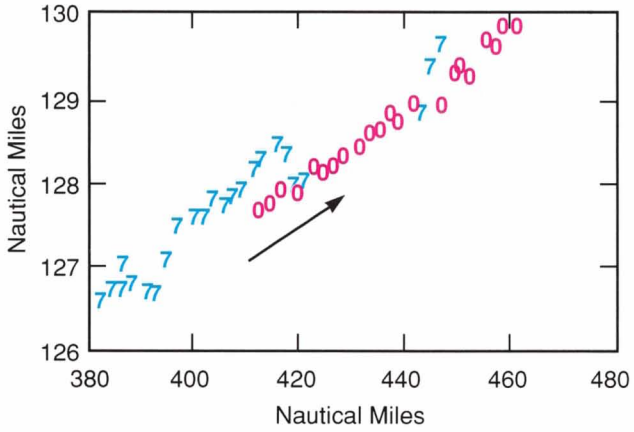


Fig. 11—Mosaic-system registration errors.

Incremental Bilateration

The standard scenario for a two-sensor system with biases is illustrated in Fig. 13(a). Each sensor is located at a known fixed position, and it can specify the location of the aircraft. The difference in the two aircraft locations results from the biases of the sensors. When bilateration is used, the calculated aircraft position (where the range arcs intersect) will be at yet a third location. Thus, whenever the measurement source changes from one scan to the next (sensor 1 only, sensor 2 only, or bilateration), the aircraft position will hop.

An alternate view of this scenario is presented in Fig. 13(b). In this method, called *incremental*

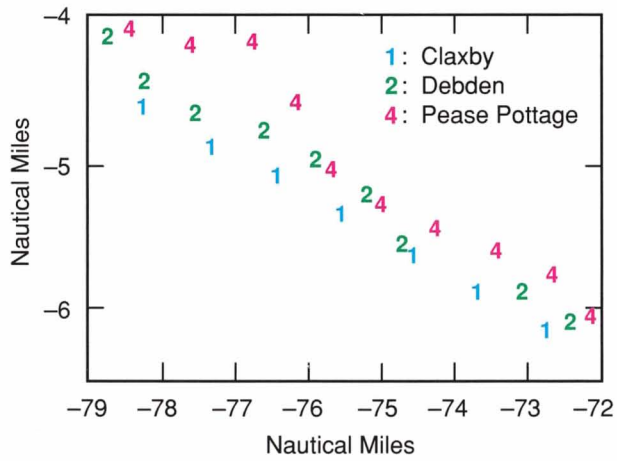


Fig. 12—Raw sensor reports.

bilateration, sensor 1 is the master sensor. Its location is fixed as before, and its aircraft measurement location is taken to be the true location. Sensor 2's measurements are used to locate sensor 2; that is, they are reversed, starting at the aircraft location and going back to earth. The resulting apparent position for sensor 2 differs from the real one by an amount that compensates for the system biases.

Incremental bilateration offers data consistency—as long as sensor 2 is assumed to be at its apparent location, sensor 1 measurements, sensor 2 measurements, and bilateration will all agree on the aircraft location, and data switching will not produce hops. The aircraft location may, in fact, be in error because of sensor 1 biases, but all aircraft that have sensor 1 as the master sensor will be in relative agreement. Thus this method removes all system bias effects.

Incremental bilateration is bias resistant. That is,

(1) the measurement positional bias is identical to that seen by using the master sensor as a single sensor, independent of any intersensor registration errors; and
(2) the azimuthal measurement noise with incremental bilateration is less than that seen with single-sensor surveillance.

The major problem with incremental bilateration is the requirement that the secondary-sensor position be calculated for each scan and for each aircraft from the reverse sensor measurements. Because the earth is a sphere, the computations are very complex and require an iterative solution. The next section resolves this problem.

Also, since the apparent location is a function of system biases, which are themselves functions of geometry, the apparent position will slowly move during any aircraft flight. A time-smoothing procedure, called outlier desensitization, therefore must be used. Figure 14 shows the raw and the smoothed data for the change in apparent distance between two sensors over time for a sample flight. As shown in Fig. 15, outlier desensitization weighs outliers (bad data points) less heavily than expected points and

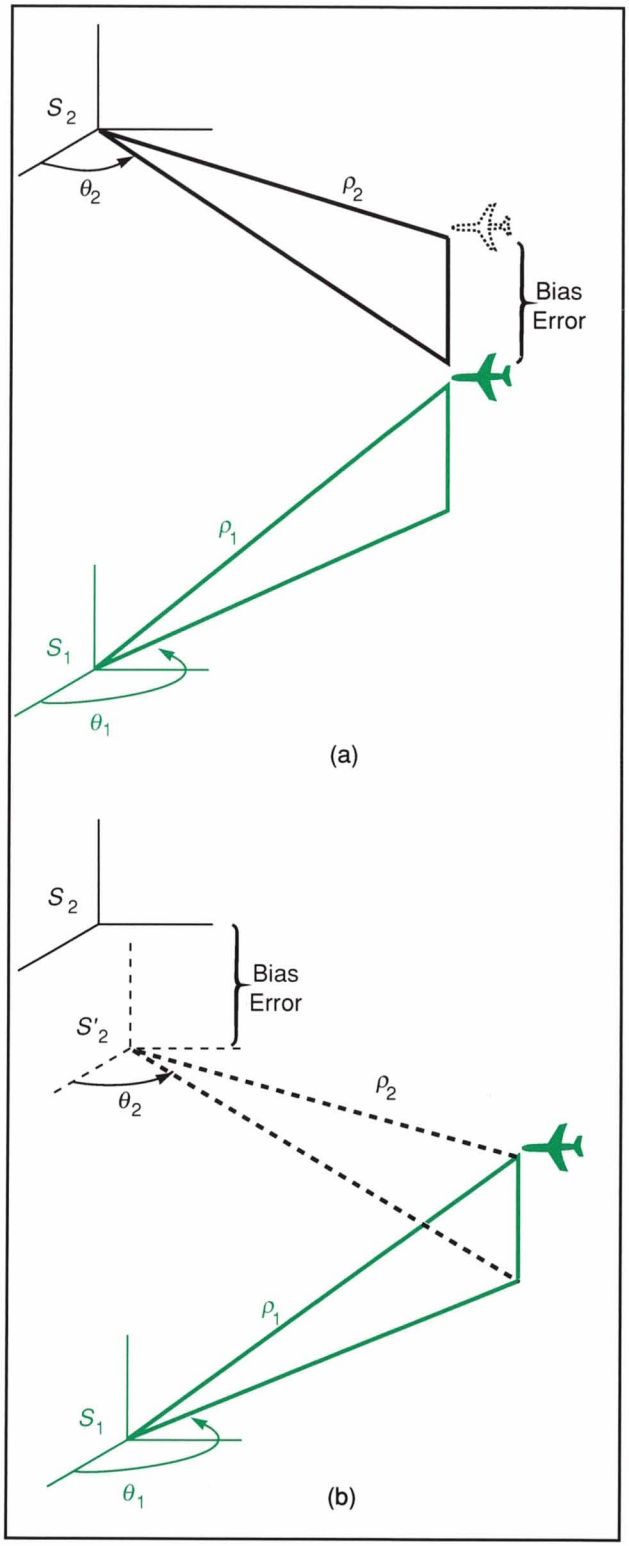


Fig. 13—Bilateration scenarios. (a) Standard. (b) Incremental. S1 represents the location of sensor 1 and S2 the location of sensor 2.

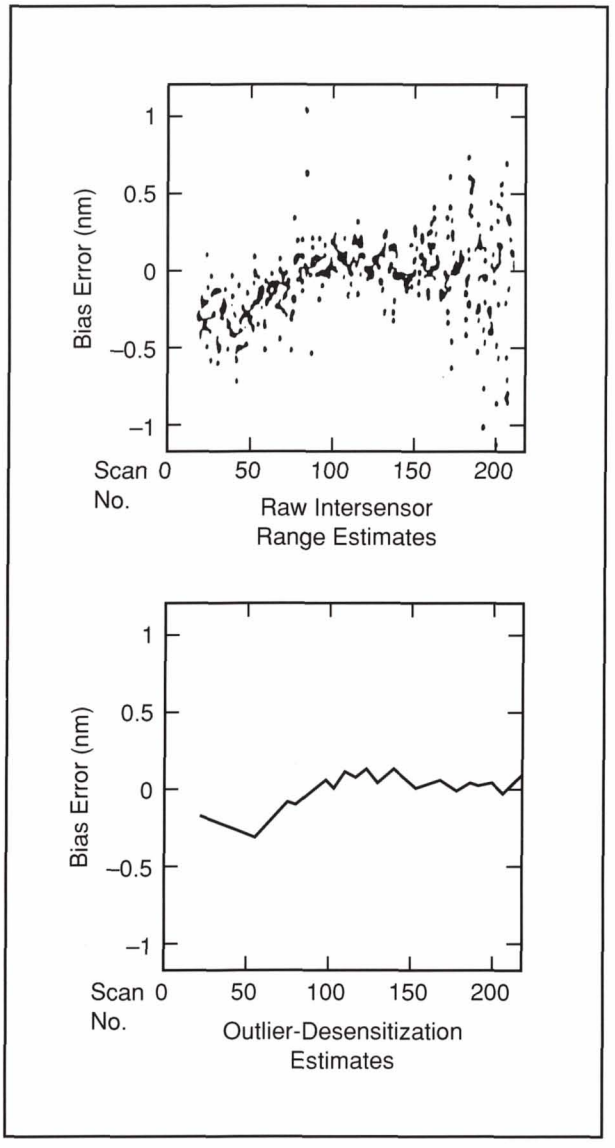


Fig. 14—Incremental bilateration smoothing.

thus produces smoother results. Outliers cannot be ignored; if they were, and if the smoothed position drifted far from the true position, all future good points would be rejected. But the desensitization approach always returns to the true position after an error.

Spherical-Equivalent Flat Earth

The key to the use of incremental bilateration is the development of a flat-earth model that is exactly equivalent *mathematically* to the spherical-earth model. No approximation is possible,

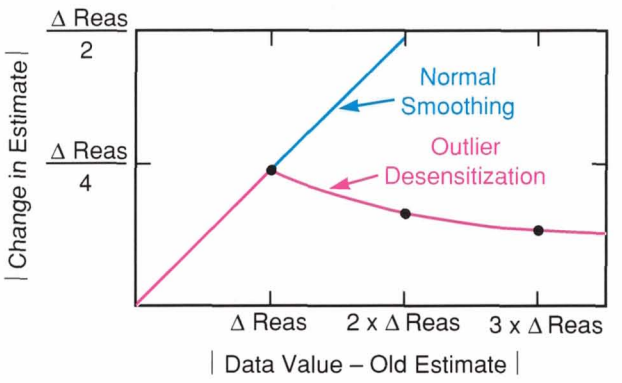


Fig. 15—Outlier-desensitization smoothing algorithm. The red curve is the function, f, described in the equation: new estimate – old estimate = f (data value – old estimate).

because errors from the approximation would compromise the accuracy of the bilateration. In addition, the model must allow the use of planar mathematics directly on the ρ, θ sensor measurements; a model requiring measurement transformations would add time-consuming calculations to the approach. In particular, the model must permit the use of the representation shown in Fig. 16, where

ρ_1, θ_1 = sensor 1 measurements
ρ_2, θ_2 = sensor 2 measurements
ρ_{ig} = ground range = $\sqrt{\rho_i^2 - z_i^2}$
z_i = height above sensor i (computed from altimeter reading)
d = apparent distance between the sensors
ψ_{ij} = apparent azimuth of sensor j from sensor i.

Using this model, we can easily compute the apparent secondary sensor location from the raw sensor measurements:

$$d = \sqrt{(\rho_1 \sin\theta_1 - \rho_2 \sin\theta_2)^2 + (\rho_1 \cos\theta_1 - \rho_2 \cos\theta_2)^2}$$

$$\psi_{12} = \theta_1 \pm \cos^{-1}\left(\frac{\rho_{1g}^2 + d^2 - \rho_{2g}^2}{2\rho_{1g}d}\right)$$

where the correct sign is a function of the actual sensor geometry. In actual use, one sign will give a physically plausible result and the other will be implausible, but the choice of the correct sign can not be predetermined.

Conversely, once the apparent secondary-

sensor location is known via the smoothing process, the true aircraft azimuth can be determined by bilateration:

$$\theta_1 = \psi_{12} \pm \cos^{-1}\left(\frac{\rho_{1g}^2 + d^2 - \rho_{2g}^2}{2\rho_{1g}d}\right).$$

The property of a planar system that permits the use of these simple equations is the alignment of the x- and y-coordinate systems of the two sensors. This alignment can be expressed for any aircraft location as

$$|x_1 - x_2| = |\rho_1 \sin\theta_1 - \rho_2 \sin\theta_2| = x \text{ component of } d$$
$$|y_1 - y_2| = |\rho_1 \cos\theta_1 - \rho_2 \cos\theta_2| = y \text{ component of } d.$$

By studying the spherical coordinate-transformation equations, and seeking to match the alignment of the x- and y-coordinates, the Spherical-Equivalent Flat-Earth Theorem (see the box) can be proved [2].

This theorem applies even if the aircraft altitude h is unknown. For any estimate of h, the spherical and spherical-equivalent flat-earth models produce identical results.

One apparent problem with the theorem is the need to determine ρ_0. A precise calculation of ρ_0 requires spherical-earth mathematics, which we are trying to avoid, and knowledge of θ_1, which we are trying to calculate. Fortunately, ρ_0 needs only be known approximately to introduce no altitude error:

$$\rho_0 \approx \sqrt{\rho_1^2 + \left(\frac{d}{2}\right)^2 - 2\rho_1\left(\frac{d}{2}\right)\cos(\theta_1 - \psi_{12})}.$$

Aircraft Tracking

To generate expected aircraft trajectories, the surveillance reports generated by the multisensor-processing algorithms are entered into a smoothing filter, or tracker. Since the predictions are used for conflict detection, a good tracker is essential.

Most current FAA multisensor tracking employs a Kalman filter and assumes straight flight dynamics. Tight filter gains are used to prevent bias effects from causing heading variations. Thus external turn detectors must determine the occurrence of turns; after a turn is detected, the Kalman filter is restarted or adjusted.

An improved tracker has been developed as part of the multisensor project. This tracker assumes that aircraft flight can be modeled as having a *constant turn rate*. (Straight flight is modeled as a constant turn rate of zero.) The advantage of the constant turn rate is that the filter successfully tracks aircraft through turns without the need for external turn detectors or filter adjustments. The constant-turn-rate Kalman filter has five state variables:

x = x position
y = y position
h = heading
$\dot{h}$ = turn rate
v = velocity.

The equations of motion that define the filter can be specified in derivative form:

$$\dot{x} = v \sin(h)$$
$$\dot{y} = v \cos(h)$$
$$\dot{h} = \dot{h}$$
$$\ddot{h} = 0$$
$$\dot{v} = 0.$$

The derivative definition leads directly to a nonlinear extended Kalman filter form of solution. Unfortunately, this approach is very com-

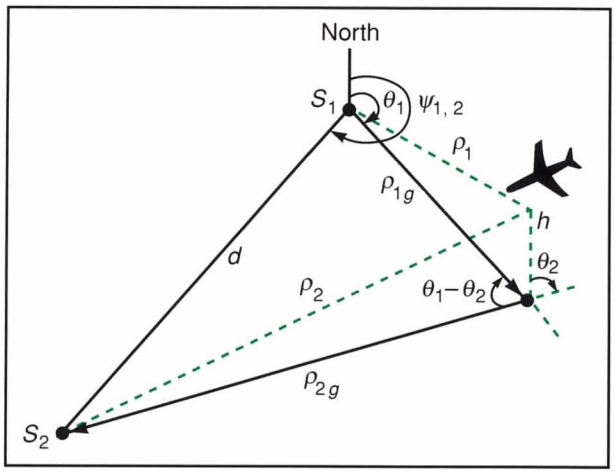


Fig. 16—Flat-earth sensor location.

Spherical-Equivalent Flat-Earth Theorem

Given: a two-sensor system defined by

(*a*) A spherical-earth model
(*b*) Sensor 1 located at latitude λ_1, longitude γ_1, and height h_{s1}
(*c*) Sensor 1 reading measurements ρ_1, θ_1, and

$$z_1 = \frac{(h - h_{s1})^2 + 2(h - h_{s1})(E + h_{s1}) - \rho_1^2}{2(E + h_{s1})}$$

(*d*) Sensor 2 located at latitude λ_2, longitude γ_2, and height h_{s2}, producing a sensor position at range d_2 and azimuth ψ_{12} relative to sensor 1
(*e*) Sensor 2 reading measurements ρ_2, θ_2, and

$$z_2 = \frac{(h - h_{s2})^2 + 2(h - h_{s2})(E + h_{s2}) - \rho_2^2}{2(E + h_{s2})}$$

Then: an *equivalent* two-sensor system can be defined by

(*f*) A flat-earth model
(*g*) Sensor 1 located at $x_{s1} = y_{s1} = z_{s1} = 0$
(*h*) Sensor 1 reading the same measurements ρ_1, θ_1, and z_1 as in (*c*)
(*i*) Sensor 2 located at x_{s2}, y_{s2}, and $z_{s2} = 0$, with x_{s2} and y_{s2} the values that locate sensor 2 relative to sensor 1 at the same azimuth ψ_{12} as in (*d*), and at the adjusted range

$$D = \left(1 + \frac{z_0}{E}\right)\left[d_2 - \frac{(h_{s1} - h_{s2})^2}{2d_2} - \frac{(h_{s1} + h_{s2})d_2}{2E} + \frac{(h_{s1} + h_{s2})(h_{s1} - h_{s2})^2}{2d_2E}\right]$$

where

$$z_0 = h + \frac{h^2 - \rho_0^2}{2E}$$

and ρ_0 is the slant range of the target measured from the location given by latitude $(\lambda_1 + \lambda_2)/2$, longitude $(\gamma_1 + \gamma_2)/2$, and height $h = 0$ of the original spherical earth model
(*j*) Sensor 2 reading the same measurements ρ_2, θ_2, and z_2 as in (*e*)

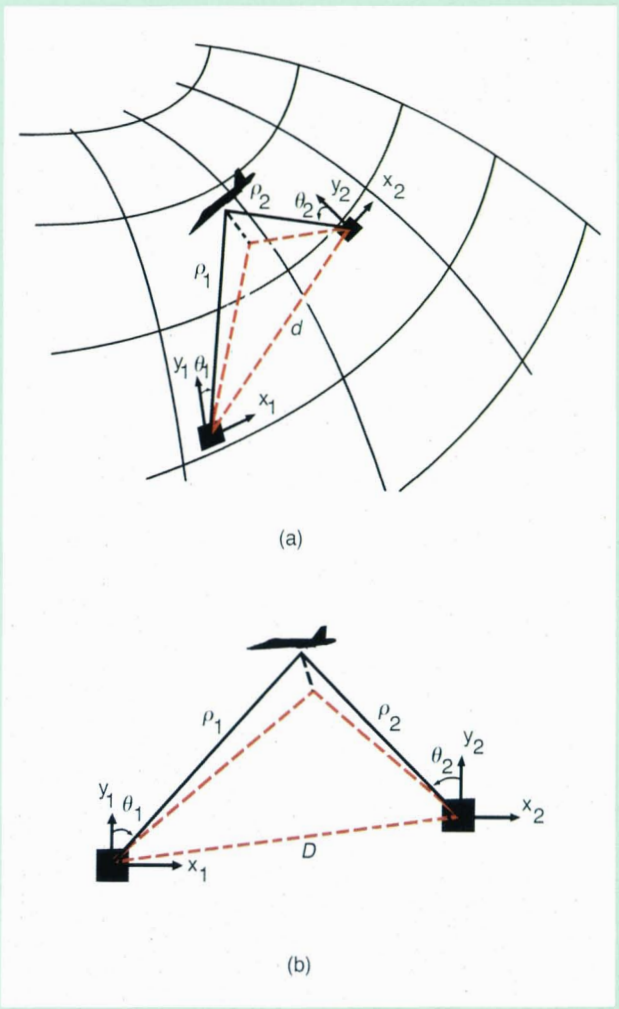


Fig. A—(a) Spherical earth. (b) Spherical-equivalent flat earth.

plex and time consuming, and can lead to unstable and divergent results. A linear Kalman filter is needed.

Fortunately, the derivatives expressed above are all integrable. For example, the x update equation can be expressed as

$$\begin{aligned} x(t+T) - x(t) &= \int_t^{t+T} v(\beta)\sin[h(\beta)]d\beta \\ &= \int_0^T v\ \sin(h+\dot{h}\alpha)d\alpha \\ &= \int_0^T v\Big[\sin(h)\cos(\dot{h}\alpha) \\ &\qquad + \sin(\dot{h}\alpha)\cos(h)\Big]d\alpha \\ &= \frac{v}{\dot{h}}\Big\{\sin(h)\sin(\dot{h}T) \\ &\qquad - \cos(h)[\cos(\dot{h}T) - 1]\Big\} \end{aligned}$$

The update equations can be expressed as

$$x(t+T) = x(t) + vT\left[\sin(h)\frac{\sin(\dot{h}T)}{\dot{h}T} - \cos(h)\frac{\cos(\dot{h}T)-1}{\dot{h}T}\right]$$

$$y(t+T) = y(t) + vT\left[\cos(h)\frac{\sin(\dot{h}T)}{\dot{h}T} + \sin(h)\frac{\cos(\dot{h}T)-1}{\dot{h}T}\right]$$

$$\begin{aligned} h(t+T) &= h(t) + \dot{h}T \\ \dot{h}(t+T) &= \dot{h}(t) \\ v(t+T) &= v(t). \end{aligned} \tag{1}$$

For a linear Kalman filter the update equations must be expressed in matrix form as

$$\mathbf{X}(t+T) = \boldsymbol{\phi}(t)\,\mathbf{X}(t).$$

Clearly the preceding equations cannot be expressed in this form—x and y are not linear functions of t. However, by using partial derivatives, an approximate form of the $\boldsymbol{\phi}$ matrix can be built:

$$\boldsymbol{\phi} = \begin{vmatrix} \frac{\delta x}{\delta x} & \frac{\delta x}{\delta y} & \frac{\delta x}{\delta h} & \frac{\delta x}{\delta \dot{h}} & \frac{\delta x}{\delta v} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \frac{\delta v}{\delta x} & \frac{\delta v}{\delta y} & \frac{\delta v}{\delta h} & \frac{\delta v}{\delta \dot{h}} & \frac{\delta v}{\delta v} \end{vmatrix}$$

where each partial derivative is computed from the corresponding update equation. For example, using Eq. 1, we find that

$$\frac{\delta x}{\delta x} = 1$$

$$\frac{\delta x}{\delta y} = 0$$

$$\frac{\delta x}{\delta h} = vT\left[\cos(h)\frac{\sin(\dot{h}T)}{\dot{h}T} + \sin(h)\frac{\cos(\dot{h}T)-1}{\dot{h}T}\right]$$

and so on.

The $\boldsymbol{\phi}$ matrix can then be used by the linear Kalman filter to calculate the covariance matrix for one-step prediction. In the absence of noise, the covariance update becomes

$$\mathbf{P}(t+T) = \boldsymbol{\phi}(t)\,\mathbf{P}(t)\,\boldsymbol{\phi}^{\mathrm{T}}(t)$$

where $\mathbf{P}$ is the state-variable covariance matrix,

$$\mathbf{P} = \begin{vmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{xh}^2 & \sigma_{x\dot{h}}^2 & \sigma_{xv}^2 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \sigma_{vx}^2 & \sigma_{vy}^2 & \sigma_{vh}^2 & \sigma_{v\dot{h}}^2 & \sigma_v^2 \end{vmatrix}$$

The steps in the recursive Kalman filter method are then the same for the turn-rate Kalman filter as for the standard linear Kalman filter [3]. That is, starting with an estimate $\mathbf{X}$ and its covariance matrix $\mathbf{P}$, and after receiving a new report $\mathbf{Y}_{\mathbf{meas}}$, a new estimate $\mathbf{X}_{\mathbf{sm}}$ (smoothed values of $\mathbf{X}$) and its covariance matrix $\mathbf{P}_{\mathbf{sm}}$ are obtained, which are then used for the starting values of the next scan.

To cover aircraft deviation from the assumed constant-turn-rate flight, noise components still must be added to the turn-rate Kalman filter. The independent noise variables chosen to represent the accelerations are

$$\ddot{h} = \text{change in turn rate}$$
$$\dot{v} = \text{change in velocity}.$$

These variables modify the update equation by increasing the prediction uncertainty.

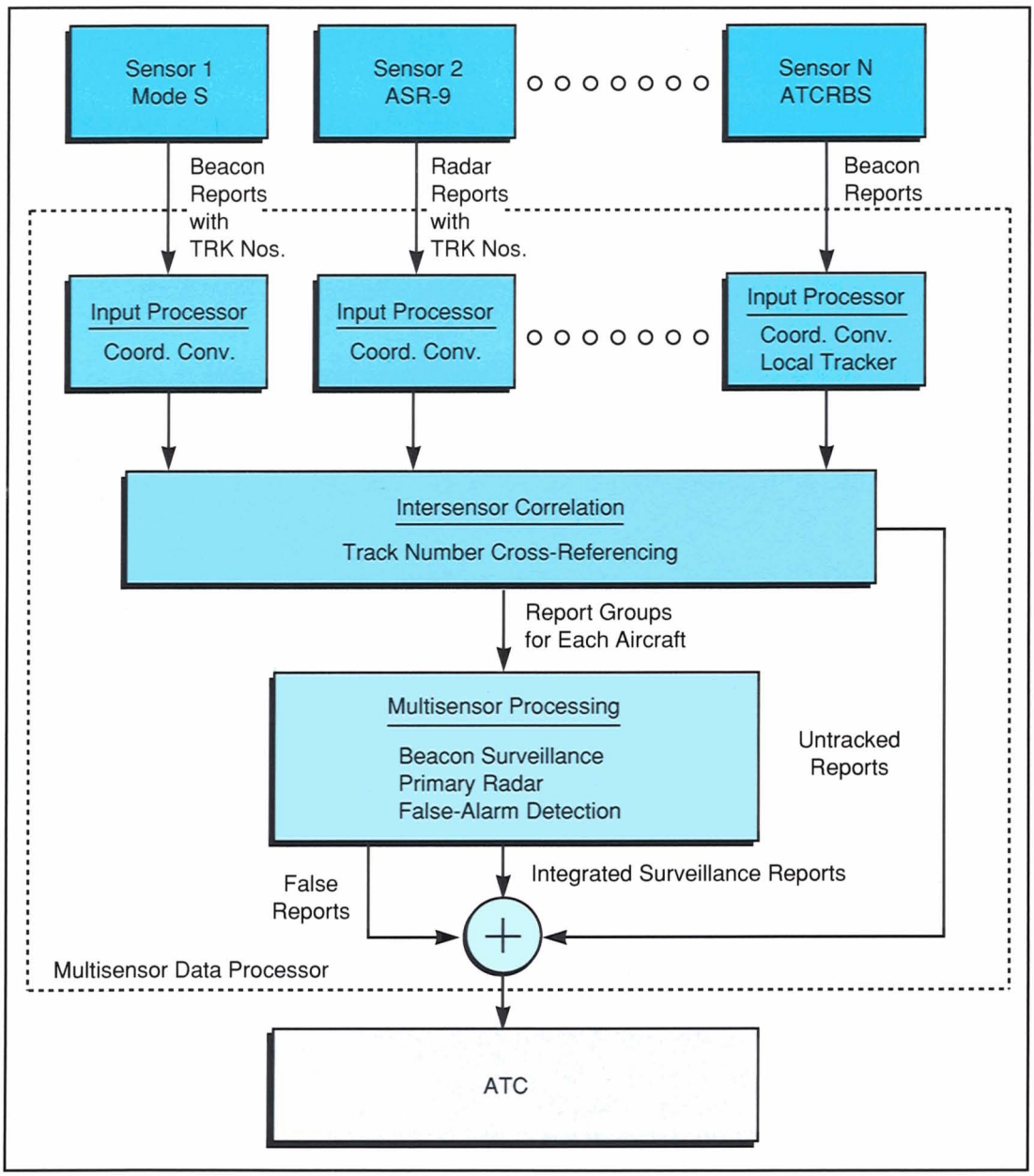


Fig. 17—Multisensor-processing system. The system is divided into three parts: intrasensor processing, intersensor correlation, and multisensor processing.

For the linear filter, the new equation becomes

$$\mathbf{P}(t+T) = \boldsymbol{\phi}(t)\,\mathbf{P}(t)\,\boldsymbol{\phi}^{\mathrm{T}}(t) + \boldsymbol{\Gamma}(t)\,\mathbf{Q}(t)\,\boldsymbol{\Gamma}^{\mathrm{T}}(t)$$

where $\mathbf{Q}$ is the noise covariance matrix:

$$\mathbf{Q} = \begin{vmatrix} \sigma_h^2 & 0 \\ 0 & \sigma_v^2 \end{vmatrix}$$

and $\boldsymbol{\Gamma}$ is analogous to $\boldsymbol{\phi}$ in the way it relates noise values to state variable updates. A linearized noise model of $\boldsymbol{\Gamma}$ has been developed.

Multisensor-Processing System

The algorithms presented in this article have been implemented in a multisensor-processing system. As shown in Fig. 17, the system is divided into three segments: intrasensor pro-

cessing, intersensor correlation, and multisensor processing.

Intrasensor processing correlates and tracks the reports from one sensor. If the sensor is Mode S, this function has already been performed. Other sensors, such as the existing FAA radar and beacon sensors, provide no such service or only partial service.

Intrasensor processing must also time-control the output of reports to intersensor correlation. Therefore, it holds its reports until a fixed delay has been reached; this delay is the same as the delay that would be experienced by reports from a Mode S sensor, because it has the greatest internal delay. The delay function thus guarantees that the stream of reports from the various sensors entering intersensor correlation is in correct time sequence.

Intersensor correlation has the task of matching the tracks from different sensors that correspond to the same aircraft. It also creates packets of reports for each aircraft and passes them to the multisensor-processing functions. The packets contain all reports from all sensors received during a scan of the preferred sensor. (A supplemental sensor with a higher rotation rate may produce two or more reports during this period.)

To meet the delay constraint imposed on the multisensor system, the release of the report packet when the preferred sensor sees the aircraft is critical: the multisensor system must not delay the output of surveillance reports. Since a single-sensor system would obviously send its report when it sees an aircraft, the rule for the multisensor system insures that the output times of the reports match those of current systems, and add no delay.

The multisensor-processing subsystem determines multilateration positions by operating on the reports in each packet. The subsystem then tracks and filters the data, and supplies accurate position and heading estimates to the air traffic controllers.

Intersensor Track Correlation

The heart of the intersensor processing system is the intersensor track correlator, the routine that determines which tracks from the

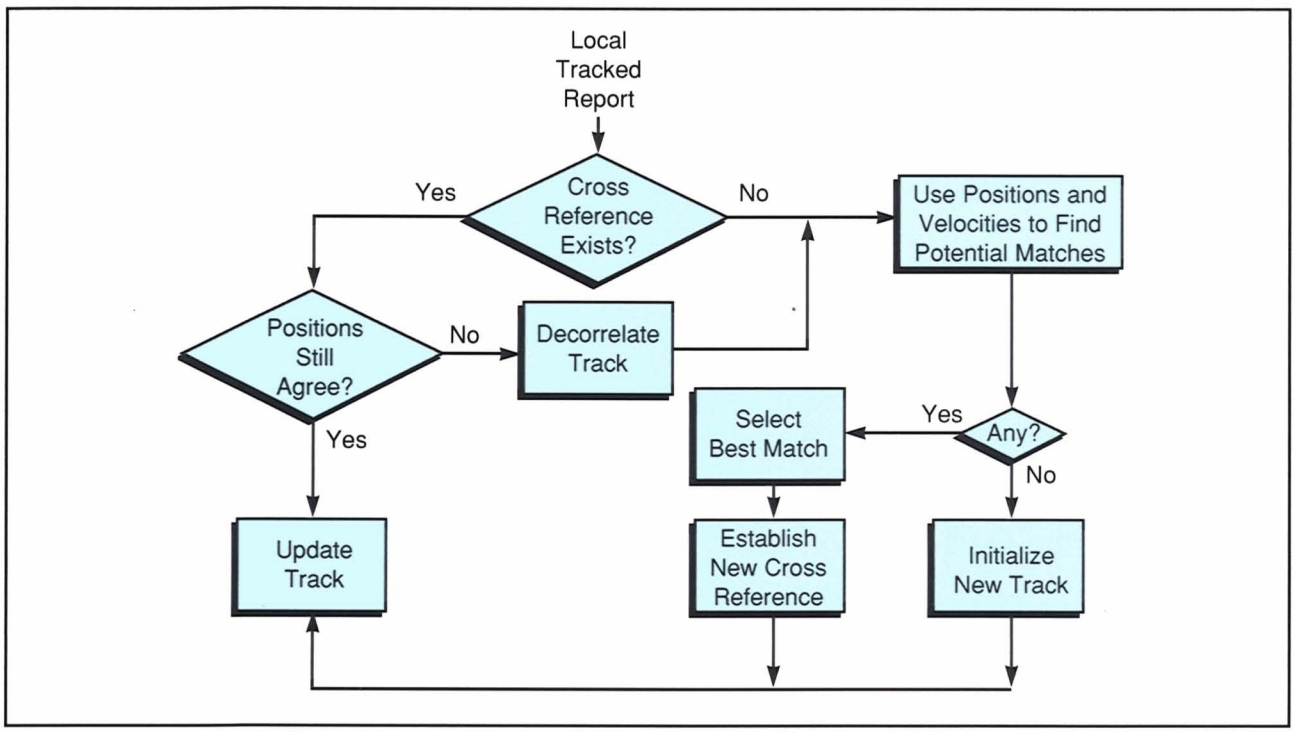


Fig. 18—Intersensor track correlation.

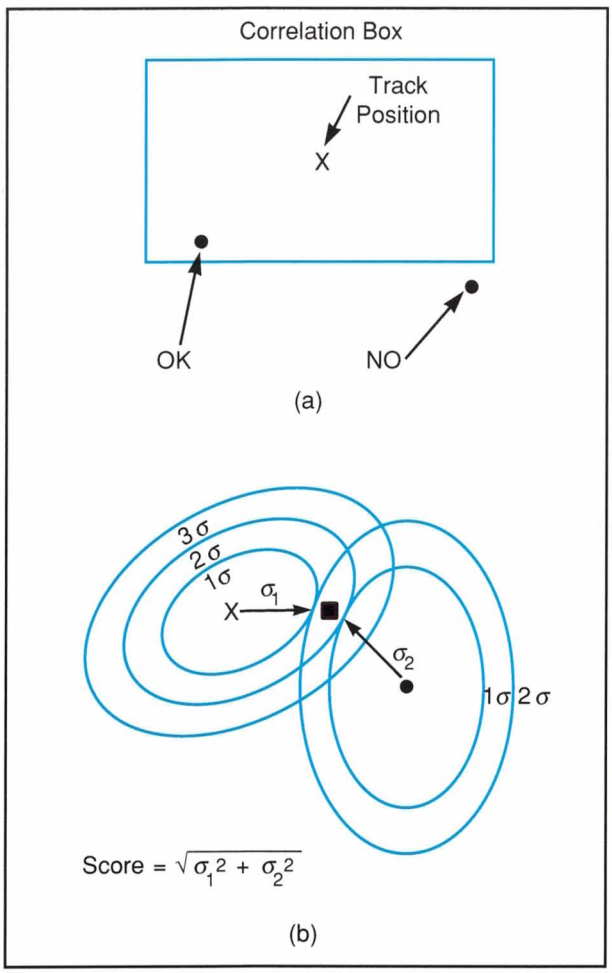


Fig. 19—Kalman position test. (a) Usual correlation. (b) Kalman correlation.

different sensors correspond to the same aircraft. The key to the routine (outlined in the flowchart of Fig. 18) is establishing and maintaining a cross-reference array that translates local sensor tracks into global system tracks.

Whenever a new local sensor track is encountered, the intersensor processor selects the proper global track (unless it is the first sighting of the aircraft) with which to assign it. The first step in the process is to apply a coarse screen that filters out noncandidate global tracks. To qualify as the global track to which the local track should be assigned, the global track must

(*a*) not already contain a track component from the sensor,

(*b*) be reasonably close in position to the local track,

(*c*) agree in code (beacon tracks only) with the local track, and

(*d*) agree in altitude (if known).

All qualifying tracks are then put through position and velocity matching tests. The position test for radar systems has traditionally used a rectangular ρ, θ box. However, a multisensor track's error ellipse has a completely different shape and orientation than a single-sensor track. Thus this usual test is a very poor selection discriminant. Instead, a Kalman position test has been developed.

The Kalman position test is illustrated by Fig. 19. By implementing the standard Kalman filter formulas and the known error ellipses of the global and local tracks, the system can use the local track position to update the global track position. The new position is then scored by its σ-distance from the two tracks, as shown in the figure. If the score is less than a preset threshold, a position match is declared.

The velocity match test does not compare speeds and headings, as in the usual procedure. Instead, both tracks are predicted ahead T seconds by using their own velocity vectors, and the resultant positions compared by using the above position test. This form of the velocity test is simpler than the usual method, adds no new parameters, and better accounts for measurement errors.

If the position and velocity tests are both passed, a match is declared. If only the position test is passed, a potential match is recorded. Such a match is retested on subsequent scans. If the velocity test is later passed, or if the position test is passed for M scans, a match is declared. The latter type of success acknowledges that successive position matches verify a velocity match, even if fishtailing causes each velocity test to fail.

If one or more matches are found for a new local track, the best match is selected as the global track for it to join. If no matches occur, a new global track is initiated.

Since intersensor matching errors can occur, either by failing to find a proper match or by choosing the wrong one, subsequent scans are checked. Failure to find a proper match is rectified by reattempting during each scan to join any

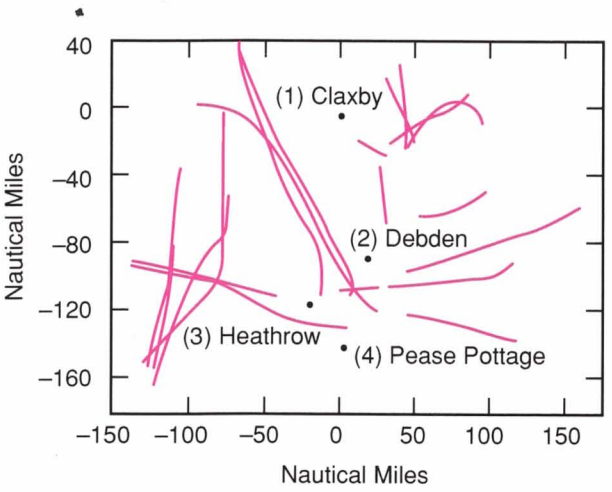


Fig. 20—Test of multilateration in Great Britain data.

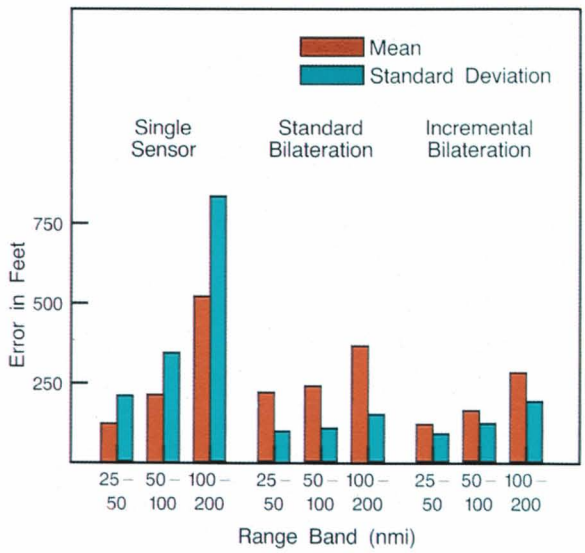


Fig. 21—Multisensor measurement of aircraft cross-range

single-sensor global track to other global tracks. Choosing the wrong track is fixed by performing a reasonableness check during each scan on the position differences among the various components of a global track. If divergence is detected, the divergent component is stripped from the global track and the matching process is repeated for it.

Real-Data Results

Testing of the multisensor-processing system to date has consisted of analyzing its performance on data collected in Great Britain, where seven sensors provide overlapping coverage of the country's airspace. Figure 20 shows the location of the three sensors selected for processing, and the trajectories of 20 aircraft studied.

The key test was an evaluation of the accuracy of the cross-range measurement obtained through multilateration. Figure 21 presents the results as a function of range and of algorithm. To have more meaning in this application, the mean (μ) and standard deviation (σ) were redefined:

$$\mu = \sum_i \rho \left| \theta_{i,\,\mathrm{meas}} - \theta_{i,\,\mathrm{true}} \right|$$

$$\sigma = \sum_i \rho \left| \left(\theta_{i,\,\mathrm{meas}} - \theta_{i,\,\mathrm{true}}\right) - \left(\theta_{i\text{-}1,\,\mathrm{meas}} - \theta_{i\text{-}1,\,\mathrm{true}}\right) \right| .$$

Thus μ measures the average cross-range position error; σ measures the average noise in the cross-range position error and hence is a measure of heading consistency.

As expected, single-sensor measurements grew linearly less accurate with range, but multilateration performance remained good for all ranges. This result confirms the tracking improvement of multilateration. Also, as predicted, incremental bilateration had a smaller mean error than standard bilateration. This result confirms the expectation that the algorithm is bias insensitive and is unaffected by registration errors.

The turning Kalman filter performance was

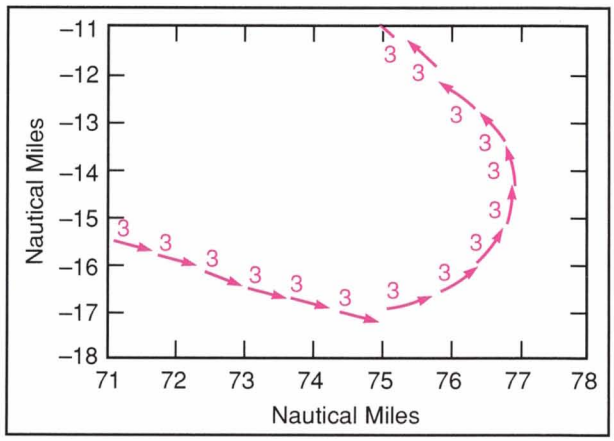


Fig. 22—Performance of turning Kalman filter during aircraft turn.

also tested with these data. Figure 22 presents filter outputs from the new turning Kalman filter, which are far superior to the results reported in Fig. 2. The new filter quickly matches the turn rate of the aircraft at turn onset (note the curved prediction arrows) and just as quickly returns to straight smoothing at turn end.

Conclusions

The work to date on multisensor data processing has provided a significant performance improvement in aircraft surveillance and tracking. Future work should extend the performance benefits to false alarm rejection, especially for primary skin radar systems.

References

1. V.A. Orlando, "The Mode S Beacon Radar System," *Lincoln Laboratory Journal* **2,** 345 (1989).
2. J.L. Gertz, "Mode S Surveillance Netting," *Project Report ATC-120,* Lincoln Laboratory (4 Nov. 1983), FAA-RD-DOT-FAA-PM-83-17.
3. A. Gelb, *Applied Optimal Estimation* (MIT Press, Cambridge, MA, 1974).

JEFFREY L. GERTZ is a staff member in the System Design and Evaluation Group. He received bachelor's, master's, and Ph.D. degrees in electrical engineering from MIT in 1965, 1966, and 1970, respectively. Jeff came to Lincoln Laboratory from Bell Telephone Laboratories in 1973. His work is now focused on surveillance and tracking of aircraft for the FAA.