DATA FUSION III: Estimation Theory

Date: March 9, 2006 **Time:** 5:00 – 7:30 PM
Location: B-300-2-3 (AAR-400) (Main Building, 2nd floor, near freight elevators)
Instructor: Dr. James K Beard
Credits: 1 **Course Code:** 0901-501-05 **Registration Number:** 13460

Today's topics:

1 Experiment Design and Hypothesis Testing

1.1 Overview

Experiment design is similar to writing a test plan, and is a milestone in evaluation of estimator or tracker performance with Monte Carlo simulations as well as real data. The steps include

- Determine and document the objectives of the experiment.
- Define methodologies to meet each of the objectives.
- Design a plan to execute each methodology.
- Procure or build, test, validate, and document the hardware and software required to execute each methodology.
- Run the experiment and collect the data.
- Analyze the data and produce reduced data to support each of the objectives of the experiment.
- Document the results.

An experiment is a project. The steps we give in this Section are examples of milestones of an experiment. Each milestone has a requirement as summarized above, and an exit criteria – usually sign-off by one or more of the project manager, technical director, or stakeholders.

1.2 Determination of Objectives

Objectives of an experiment are the requirements, and are set by the stakeholders. Stakeholders in an experiment include

- Those who pay for the experiment.
- Those who need the results of the experiment.
- Those who receive the results of the experiment.
- Those who have sign-off authority on exit criteria for each milestone of the experiment.
- The owners of equipment or other property evaluated, used, or otherwise directly impacted by the results of the experiment.

In setting objectives for an experiment, inputs should be available in some form from all identified stakeholders, and these objectives should be documented. All stakeholders should have an opportunity to review the experiment objectives. This can be as simple as a letter that clearly states the objectives of the experiment that is forwarded to the stakeholders with notice that the experiment will proceed after inputs are received, giving a specific date by which any responses are required. Or, it may consist of a technical report or technical interchange meeting (TIM) which has the report and meeting minutes formally signed off by the attendees.

In the initial stages of development of an estimator to perform a function defined in a larger project, such as remedying biases in multiple radars that will have their data fused later in the project, the experiment objective can be to define the requirements and characteristics of a model to generate synthetic data, or to provide a repository of real or synthetic data for use in development of estimation and fusion methodologies and other technology. Existence of this data repository, and verification and documentation of the characteristics of this data, will satisfy the requirements for subsequent experiments.

Typical experiment objectives for evaluation of an estimation algorithm are to evaluate the effect of specific biases in calibration between radars for a data fusion estimator, and to determine the size, shape and orientation of localization ellipsoids as a function of time for a specific set of scenarios and biases.

1.3 Define methodologies to meet each of the objectives.

Given a clearly defined set of objectives, the methodology to meet each should be evident. Some quantization of objectives may be required at this step.

Methodologies for data generation can include selection of random number generator, definition of tests for the random number generators, and definition of the scenarios to be used in simulation. Real data collection can include selection of radars and documentation of applicable calibration data for these radars. Use of modified real data can include introduction of azimuth biases, and the methodology would include specifics of how this is done.

Methodologies for data reduction can include definition of a recursive estimator (Kalman filter), one or more batch algorithms, or a combination. Methodologies to be applied in real-time and in post-experiment data analysis are defined in this stage.

1.4 Plan for Execution with Milestones

For a simple computer run generating or using real or synthetic data, this step may be as simple as assigning resources and setting a date for completion. Most uses of data for algorithm definition and development are multi-stage processes, and some definition of the process by the responsible engineering authority (REA) and documentation is valuable to the project manager. As the project progresses, lessons-learned will be documented and some of these will modify the direction of the effort. Documenting these changes as they occur is essential to keeping control of the project objectives, its schedule, and its budget.

1.5 Hardware and Software to Execute the Methodologies

Every model or algorithm that is used in an experiment must be validated for the results of the final experiment to withstand scrutiny essential to using the results in any decision process. Hardware must have calibration documentation, and instrument calibration must have a calibration traceability to NIST standards. Software must be validated to be used in simulations where results will be reported for use in making decisions. Software calibration can be one of the following:

- FAA or NTSB standard models such as the Next Generation Model for the National Aerospace System, Multi-Organizational Human-In-The-Loop Simulation, and other models accepted by the FAA, or DoD models such as EADSIM, industry-standard models such as STK, etc.
- Internal WJHTC models with a documented validation trail.

- New models for the current project that have a documented validation trail using real or synthetic data.
- Submodules or modules that have documented tests on file.

The software development process involves testing and sign-off for every module and assembly of modules. This may be very informal or it may involve a formal review and approval process, but whatever it is should be documented and made part of the project documentation database.

1.6 Running the Experiment

When the date scheduled for running the experiment arrives, you will have all the elements required to execute the experiment:

- Requirements for meeting each objective of the experiment.
- Hardware and software required to run the experiment, take the data, and monitor the experiment.
- Hardware and software required to store and perform post-experiment data analysis.
- Resources required to run the experiment, including hardware, software, computers, and assigned personnel.

1.7 Data Analysis

Post-experiment data analysis is the most important and usually the most valuable work performed in connection with an experiment. This is where most objectives are met – validation of data, determination of results, certification of completion of objectives, etc.

1.8 Document the results

The outcomes and conclusions of the experiment must be provided to the stakeholders in a form suitable for use in their intended purposes such as decision-making. This usually means a test report, and escrow of the data and software used in its analysis, and calibration trail documentation for any hardware used in preparation or execution of the experiment. Soft copy should be provided on magnetic media or CDROM to allow verification of results, repeating of important parts of the data analysis, or other reexamination of the experiment, data produced by the experiment, or the results of the experiment.

2 Hypothesis Testing

2.1 Defintion of Hypothesis Testing

Many experiment conclusions are defined through hypothesis testing. Hypothesis testing is a decision process. Its elements are

- Quantitative definition of the hypothesis, such as the existence of a significant bias in azimuth of a radar set.
- Definition of a statistic that has known probability distribution function when the hypothesis is true, false, or both.

- Either *a priori* definition of a confidence level to which the hypothesis is tested, reporting of the confidence level of the test of the statistic, or both.
- Computation of the statistic and testing it against a threshold determined by an *a priori* confidence level, computation of the confidence level, or both.

2.2 Designing a Threshold Test

2.2.1 Example 1: Looking for a Bias

Designing a threshold test for hypothesis testing requires that you develop a statistic that represents a numerical indication of whether your hypothesis is true or false. For example, we want to determine whether a specified azimuth offset between two radars is observable. We develop two target tracks in our simulation, one from each radar. Our test statistic is the Bhatacharya distance between the target positions, taken at a specific time – at a point when data fusion would be beneficial, such as at the mid-point in the overlapping portion of the coverage areas of the two radars.

We will denote the two target positions that the trackers report as $\underline{x}_1$ and $\underline{x}_2$. We will take the state vector with the latest update for position for one of the tracks and extrapolate the other track forward in time for the other track position so that they are effective at the same epoch, or instant in time. The covariance of their difference is

$$Cov\left\{\underline{x}_1 - \underline{x}_2\right\} = P_1 + P_2 \tag{2.1}$$

Remember, here we are looking at only the position states, not the entire state vector. If we have altitude from both target tracks but the altitude tracker is not incorporated into the same Kalman filter as the position tracker, augment the state variable and use both covariances in the augmented covariance matrix, with zeros for the cross-correlation terms between position and altitude.

The Bhatacharya distance is

$$J = \left(\underline{x}_1 - \underline{x}_2\right)^T \cdot \left(P_1 + P_2\right)^{-1} \cdot \left(\underline{x}_1 - \underline{x}_2\right) \tag{2.2}$$

We know that this quantity is chi-square distributed if there is no bias between the radars. Suppose that our desired confidence level for the test is 95%, and that we have latitutde, longitude, and altitude available. Then, the test statistic *J* will, if there is no bias between the radars, be chi-square with three degrees of freedom. We can use the 95% point of this distribution, about 7.8, as a threshold for the statistic *J*. We will know that we will have a threshold crossing only 5% of the time when there is no bias.

If no significant bias is detected using this test, a fused track that uses position data should not be degraded by the bias. However, a good experiment that provides in-depth insight will provide the bias estimate and its confidence level over the time the aircraft is in the coverage area of both radars. A second data reduction experiment that compares fused tracks with and without a specified bias could also be conducted to add perspective on the problem.

2.2.2 Example 2: Estimating the Bias

Another statistic that can be used to evaluate bias in data fusion is estimation of the bias using the radar track data. This can be done in any or all of a number of ways: However, the principle of requirements flowdown in definition of experiment objectives tells us that we should define a test statistic so that it either tells us whether we have impact, or provides us with insight into the issue that will help us with the next phase of design. A test that is particularly useful in evaluation of real data is the estimation of bias.

When we estimate the bias itself, it is helpful to make the state the parameter that is being estimate. If a bias in azimuth of one radar is the estimated parameter, the difference between the positions in terms of latitude and longitude will be affected differently by the bias for different geometries between the aircraft and the two radars. So, it makes sense to estimate the azimuth bias itself, not a resulting bias in tracked position.

Use of tracker positions in the previous example provided us with an indication of how bias might affect a fused track, particularly one that uses tracker outputs or is initialized using tracker states and covariances. Here, we are estimating the bias and we want the best possible estimate – an error variance that approaches the Cramer-Rao bound. So, we design an estimator specifically for this purpose:

- We prefer a batch estimator designed using the method of maximum likelihood.
- We use raw radar return data, not tracker states, as data for the estimator.
- We make the azimuth bias one of the states in the estimator.
- We test the mean against the variance of the estimator to determine whether the bias is detectable using radar returns.

2.2.3 Formulation of Recursive Estimator

A recursive estimator may be required if the aircraft track is not accompanied by truth data, or cannot be accurately characterized other than the radar tracks. Here we will pose the azimuth bias in data from one of the radars as a constant angle offset and estimate it from radar return data from both radars.

We will denote one of the radars as reference radar, and we denote the state vector and covariance matrix by suffix “r”, and the other as the bias analysis radar, and we denote the state vector and covariance matrix from that radar with the suffix “b”.

We will be working with the range and azimuth measurements from each of the radars. This is very important because the measurements are uncorrelated, and conventional estimation techniques can be used. If we use tracker outputs, the effect of the Kalman filter on the noise is to average it, producing correlations. Correlated measurement noise in a recursive estimator will cause large errors unless accounted for in the model. This is called measurement differencing, and is treated in Gelb pages 133-136. The complexity of measurement differencing in Kalman filters is sufficiently complex that only the problems and a sample solution for continuous Kalman filters (used in autopilots, not trackers) and the discrete case that we need for radar trackers is referred to a classical paper, not treated directly in Gelb. An additional consideration is that the radar trackers may use a nonlinear transformation between the measurement (polar) and the states (Cartesian – East-North-Up or latitude-longitude-altitude) so that the Kalman update is not a minimum variance unbiased estimator; these errors may be negligible for ATC purposes but can overwhelm biases that we are looking for.

Our "measurements" are the range and azimuth data from each radar. We will use complex variables for positions in a North-East-Down coordinate system and assume that bearing measurements are measured clockwise from North. We will give our measurement model and our state vector, then define the measurement noise matrix R, find the measurement sensitivity matrix H and the system propagation matrix Φ. We will formulate the Kalman filter and then show how the process noise matrix Q is formulated appropriately for aircraft and mapped into the state vector space through a mapping matrix G. Last, we will address the issues of initialization.

2.2.3.1 Measurement Model

We will use complex variables for radar and aircraft positions, but denote them using underlines. The variables are collected in a table below.

Table 1: Variables for Recursive Bias Estimator

Variable Name	Description
$\underline{pr}$	Position of the reference radar
$\underline{pb}$	Position of the bias analysis radar
$\underline{pa}$	Position of the aircraft
$\underline{va}$	Velocity of the aircraft
Rr	Range from reference radar to aircraft
Rb	Range from bias analysis radar to aircraft
br	Bearing of aircraft from reference radar
bb	Bearing of aircraft from bias analysis radar
$bias$	Bearing bias of bias analysis radar
R_0	Normalization range, a constant selected for convenience

We note that

$$\frac{Rr}{R_0} \cdot \exp\left(j \cdot br \right) = \frac{\underline{pa} - \underline{pr}}{R_0} \tag{2.3}$$

so we can write

$$\ln\left(\frac{Rr}{R_0} \right) + j \cdot br = \ln\left(\frac{\underline{pa} - \underline{pr}}{R_0} \right) \tag{2.4}$$

This provides an analytic relationship between the positions and the measurements that also is posed in numerical and physical units that are close to each other, which will provide a basis for an implementation that has better numerical properties than one that mixes units such as nautical miles and radians in a covariance matrix.

We now have a measurement model:

$$\underline{y} = \underline{h}\left(\underline{x}\right) + \underline{v} = \begin{bmatrix} Rr \\ br \\ Rb \\ bb \end{bmatrix} + \underline{v} = \begin{bmatrix} \mathrm{Re}\left\{ \ln\left(\dfrac{\underline{\underline{pa}} - \underline{\underline{pr}}}{R_0} \right) \right\} \\ \mathrm{Im}\left\{ \ln\left(\dfrac{\underline{\underline{pa}} - \underline{\underline{pr}}}{R_0} \right) \right\} \\ \mathrm{Re}\left\{ \ln\left(\dfrac{\underline{\underline{pa}} - \underline{\underline{pb}}}{R_0} \right) \right\} \\ \mathrm{Im}\left\{ \ln\left(\dfrac{\underline{\underline{pa}} - \underline{\underline{pb}}}{R_0} \right) \right\} + bias \end{bmatrix} + \underline{v} \tag{2.5}$$

2.2.3.2 Measurement Noise Covariance Matrix

The measurement noise covariance is

$$R = \begin{bmatrix} \dfrac{\sigma_{Rr}^2}{R_0^2} & 0 & 0 & 0 \\ 0 & \sigma_{br}^2 & 0 & 0 \\ 0 & 0 & \dfrac{\sigma_{Rb}^2}{R_0^2} & 0 \\ 0 & 0 & 0 & \sigma_{bb}^2 \end{bmatrix} \tag{2.6}$$

2.2.3.3 State Vector

Here we will use the state vector that incorporates the aircraft position and velocity, and the bias:

$$\underline{x} = \begin{bmatrix} \underline{pa} \\ \underline{va} \\ bias \end{bmatrix} \tag{2.7}$$

Other alternatives are the aircraft position and velocity in polar coordinates about either of the radar, latitude-longitude-altitude, etc. Other formulations may behave better because the states may be linearly related to the measurements, particularly when we use polar coordinates about the bias investigation radar. We use the state vector incorporating aircraft position and velocity for simplicity here.

2.2.3.4 The Measurement Sensitivity Matrix

We note that

$$\frac{\partial \ln\left(\frac{\underline{pa}-\underline{pr}}{R_0}\right)}{\partial \underline{pa}} = \frac{1}{\underline{pa}-\underline{pr}} \tag{2.8}$$

We also note the Cauchy-Riemann conditions for an analytic function of a complex variable. The Cauchy-Riemann conditions are equivalent to the condition that the derivative of a complex function of a complex variable is not a function of the way that the limit is taken in the definition of the derivative,

$$\frac{df(z)}{dz} = \lim_{\delta z \to 0} \frac{f(z) - f(z+\delta z)}{\delta z} \tag{2.9}$$

That is, the derivative is the same for any definition of δz – real, imaginary, or any combination. The Cauchy-Riemann conditions are posed in terms of the real and imaginary parts of the function and the dependent variable,

$$f(x + j \cdot y) = u(x + j \cdot y) + j \cdot v(x + j \cdot y) \tag{2.10}$$

The Cauchy-Riemann conditions are

$$\begin{aligned} \frac{\partial u(x+j\cdot y)}{\partial x} &= \frac{\partial v(x+j\cdot y)}{\partial y} \\ \frac{\partial v(x+j\cdot y)}{\partial x} &= -\frac{\partial u(x+j\cdot y)}{\partial y} \end{aligned} \tag{2.11}$$

With the Cauchy-Riemann conditions and the definition of the measurement model (2.5), we can write the measurement sensitivity matrix as

$$H = \frac{\partial \underline{y}}{\partial \underline{x}} = \begin{bmatrix} \mathrm{Re}\left\{\frac{1}{\underline{pa}-\underline{pr}}\right\} & -\mathrm{Im}\left\{\frac{1}{\underline{pa}-\underline{pr}}\right\} & 0 & 0 & 0 \\ \mathrm{Im}\left\{\frac{1}{\underline{pa}-\underline{pr}}\right\} & \mathrm{Re}\left\{\frac{1}{\underline{pa}-\underline{pr}}\right\} & 0 & 0 & 0 \\ \mathrm{Re}\left\{\frac{1}{\underline{pa}-\underline{pb}}\right\} & -\mathrm{Im}\left\{\frac{1}{\underline{pa}-\underline{pb}}\right\} & 0 & 0 & 0 \\ \mathrm{Im}\left\{\frac{1}{\underline{pa}-\underline{pb}}\right\} & \mathrm{Re}\left\{\frac{1}{\underline{pa}-\underline{pb}}\right\} & 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

Note that we have only two complex quantities to compute, and that the real and imaginary parts of these two complex numbers are used twice. Also note that the columns corresponding to the velocity states are zero, which means that velocity is not observable from a single measurement – and intuitively satisfying point, but one that raises questions of initialization that we will address later.

2.2.3.5 The State Vector Extrapolation Equation

We note that our aircraft motion model is non-accelerating (constant velocity), so that position extrapolation is simply

$$\underline{\tilde{pa}}_i = \underline{\hat{pa}}_{i-1} + \underline{\hat{va}}_{i-1} \cdot \left(t_i - t_{i-1}\right) \quad (2.13)$$

The aircraft velocity remains constant, as does the bias. Thus the system transition matrix is

$$\Phi = \begin{bmatrix} 1 & 0 & t_i - t_{i-1} & 0 & 0 \\ 0 & 1 & 0 & t_i - t_{i-1} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

For the state vector as we have posed it here, and a nonaccelerating target model, this is a very simple definition, and this simplicity is one of the reasons that we use the state vector definition in (2.7). Use of a better definition such as polar coordinates about the bias investigation radar would require that we write an algebraic expression for $\underline{\tilde{pa}}_i$ in terms of $\underline{\hat{pa}}_{i-1}$ and find Φ from the more general equation

$$\Phi = \frac{\partial \underline{\tilde{pa}}_i}{\partial \underline{\hat{pa}}_{i-1}} \quad (2.15)$$

This more general equation applies here, but we have stated the state vector so that we can write Φ directly.

2.2.3.6 Formulation of the Kalman Filter

The Kalman filter is conventional here. The extrapolation equations are

$$\begin{aligned} \underline{\tilde{x}}_i &= \Phi \cdot \underline{\hat{x}}_{i-1} \\ \tilde{P}_i &= \Phi \cdot P_{i-1} \cdot \Phi^T + G \cdot Q \cdot G^T \cdot \left(t_i - t_{i-1}\right) \end{aligned} \quad (2.16)$$

We will address the process noise matrix Q and the noise mapping matrix G below. The update equations are

$$\begin{aligned} K_i &= \tilde{P}_i \cdot H_i^T \cdot \left(H_i \cdot \tilde{P}_i \cdot H_i^T + R\right)^{-1} \\ \underline{\hat{x}}_i &= \underline{\tilde{x}}_i + K_i \cdot \left(\underline{y} - \underline{h}\left(\underline{\tilde{x}}_i\right)\right) \\ P_i &= \left(I - K_i \cdot H_i^T\right) \cdot \tilde{P}_i \cdot \left(I - K_i \cdot H_i^T\right)^T + K_i \cdot R \cdot K_i^T \end{aligned} \quad (2.17)$$

where $\underline{h}\left(\underline{\tilde{x}}_i\right)$ is given as part of (2.5).

2.2.3.7 Mapping the Process Noise to the State Vector

We want to use process noise that meets these requirements:

- Separate process noise for position and velocity states.
- Separate process noise for across the flight path and along the flight path for aircraft position and velocity.
- No process noise for the bias.

This can be done by defining the process noise variances, per unit time, individually as desired. We list them in the table below in the order that they appear down the main diagonal of the process noise matrix Q. The off-diagonal terms of Q are zero.

Table 2. Process Noise Coefficients

Symbol	Defintion	Remarks
qpa	Position along flight path	Usually zero (see "Tuning the Filter" below)
qpc	Position across flight path	Usually zero (see "Tuning the Filter" below)
qva	Velocity along flight path	Aircraft, autopilot, and weather dependent
qvc	Velocity across flight path	Aircraft, autopilot, and weather dependent
qb	Noisy bias changes with time	Nearly always zero

2.2.3.8 The Process Noise Mapping Matrix

We need to map process noises that are defined in terms of the aircraft orientation to North-East-Down, the coordinate system used in the state vector as defined in (2.7). Thus there is a coordinate rotation in the mapping matrix that rotates from aircraft forward-aircraft starboard (right)-aircraft down to North-East-Down. We omit altitude states here, and we note that we can map a variable along the aircraft flight path to NED by

$$\langle \text{NED variable} \rangle = \frac{\underline{va}}{|\underline{va}|} \cdot \langle \text{Along-path variable} \rangle \tag{2.18}$$

and that we can map a variable across the aircraft axis to starboard (right) by

$$\langle \text{NED variable} \rangle = \frac{j \cdot \underline{va}}{|\underline{va}|} \cdot \langle \text{To-starboard varible} \rangle \tag{2.19}$$

Thus we can write the rotation matrix as

$$\langle \text{Rotation, Aircraft to NED} \rangle = \begin{bmatrix} \frac{\operatorname{Re}\{\underline{va}\}}{|\underline{va}|} & -\frac{\operatorname{Im}\{\underline{va}\}}{|\underline{va}|} \\ \frac{\operatorname{Im}\{\underline{va}\}}{|\underline{va}|} & \frac{\operatorname{Re}\{\underline{va}\}}{|\underline{va}|} \end{bmatrix} \tag{2.20}$$

If we formulate our process noise matrix Q with only the velocity process noise, which is reasonable because the other process noises are usually zero, the mapping matrix G is

$$G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\operatorname{Re}\{\underline{va}\}}{|\underline{va}|} & -\frac{\operatorname{Im}\{\underline{va}\}}{|\underline{va}|} \\ \frac{\operatorname{Im}\{\underline{va}\}}{|\underline{va}|} & \frac{\operatorname{Re}\{\underline{va}\}}{|\underline{va}|} \\ 0 & 0 \end{bmatrix} \tag{2.21}$$

Inclusion of process noise states for aircraft position or radar azimuth bias will require addition of appropriate columns to G.

2.2.3.9 Initialization of the Kalman Filter

Initialization of the Kalman filter is the specification of the initial state vector and covariance. When all of the states are observable from a single measurement, this can is done by simply mapping the first measurement and its errors into the states and their errors. Usually, however, some states are not observable using a single measurement. In our current problem, the aircraft velocity states require at least two measurements, which can be, at a minimum, one measurement from each radar. The radar azimuth bias requires at least three measurements. Initialization usually is done by one of two methods:

- Initialization of unobservable states is done using nominal values, along with initialization of their covariances with very large values, reflecting the uncertainty of the nominal values. This can cause problems in settling of the filter because this is pseudo information that affects the early behavior of the output of the Kalman filter, and the mapping of the covariance into the Kalman gain can cause lasting transients, particularly in nonlinear filters. However, adequate performance is obtained in many applications, particularly when performance immediately after initialization is not important.
- Saving one, two or more measurements and using a batch estimator to initialize the Kalman filter. This ensures that the Kalman filter has the best possible information when sequential estimation is started.

Clearly a small batch estimator with two or more measurements, including at least one from each radar, is the best option.

2.2.3.10 Tuning the Filter

Tuning a Kalman filter is the setting of the process noise coefficients in the Q matrix. Occasionally the Kalman filter designer will modify the measurement noise matrix R to reflect correlations that may exist between measurements but may or may not be supplied to the Kalman filter. A few points are important in tuning the Kalman filter by varying the process noise coefficients in the Q matrix:

- Adding process noise in the position states decreases filter accuracy by increasing the position mappings in the Kalman gain, and the position variances in the covariance matrix decrease as one over time – i.e., slowly.
- Adding process noise in the velocity states keeps the covariance from collapsing by providing increases to the appropriate terms for velocity covariances, and, through the extrapolation equation, the system propagation matrix Φ will map velocity covariances to the position covariances. This is one-way; position state covariance does *not* map to velocity states through extrapolation or update. Velocity state variances are decreased as one over time cubed by the successive updates.

- Changes in appropriate process noise during the operation of the Kalman filter can be detected, and the process noise changed, by computing and using the squared filter error in the most accurately measured variable. In the case of MTI radars without monopulse, this is the range measurement. This squared error can be mapped into changes in the process noise. This results in an adaptive Kalman filter.
- States that have no process noise, either directly or through mapping in the covariance extrapolation or update, will eventually exhibit a variance which is effectively zero. This can be acceptable if this variable is the only state of real interest and the run ends when this is detected, as the case here with the azimuth bias state, but this is covariance collapse and essentially cessation of the tracker to continue to estimate that state. Often this will cause numerical problems in the covariance extrapolation, update, and the Kalman gain computation, and results in erroneous computations and loss of stability of the Kalman filter.

The recommended process here is to compute the squared range error from the reference radar measurements and use this as a multiplying factor on the process noise coefficients for the velocity states, leaving the other process noise terms as zero. The remaining two process noise coefficients are minimized through trial-and-error to make them the smallest values that provide good performance of the tracker. The run ends when the covariance of the bias state goes to zero, or decreases to a value that indicates that the estimation of bias has ceased. If this is too early in the run, process noise for the bias state can be introduced; the minimum value that keeps the covariance from collapsing should be used.

2.2.4 Example 3: The Impact of Bias on Fusion Performance

This experiment requires that we run a scenario with three trackers: trackers for each radar, and a fused track. We compare the results of the fused track with the tracks from the two radars for performance.

A bias in either or both radars may be small enough so that, from the perspective of the position states, it appears to be unlikely to degrade performance of a fused track. The velocity states inherently differentiate the measurements, so that alternate updates between radars that have a bias between their measurements will present the fusion tracker with aircraft positions that zig-zag slightly. The result is that the velocity and heading states will be noiser than the corresponding single-radar tracker states. We specifically look at the ensemble covariance (not the tracker or Kalman filter covariance) to provide us with an estimate of performance of the tracker. We might also look at maximum peak deviation of the state from truth, which was telling in showing the difference between an alpha-beta and a Kalman tracker in our simulations.

3 The Localization Ellipse

3.1 Definition

When looking at visualization aids for simulations, we may consider the localization ellipse. The localization ellipse is the chi-square limit for state errors for target position, displayed as a graphic on a PPI or cartographic display. The ellipse can be displayed as a stretched octagon for simplicity in generating graphics. Here we show how to find the base parameters for generating the graphic: the size and orientation of the ellipse.

3.2 The Localization Ellipse and the Covariance Matrix

The localization ellipse on a display with coordinates of East-North will be only a function of the target position in those coordinates, and the two-by-two portion of the state covariance matrix. We can consider the covariance matrix as used internally by the Kalman filter, and we can use the sample covariance from tracker errors computed from a Monte Carlo, and compare them to evaluate how we might interpret the localization ellipse as computed from the covariance from a Kalman filter or even correct it. We would like the ellipse to display a contour of constant Bhatacharya distance from the position of the target track, or the locus of the $\Delta\underline{x}$ in

$$J = \Delta\underline{x}^T \cdot P_{EN}^{-1} \cdot \Delta\underline{x} \tag{3.1}$$

We know that the contour parameter *J* is chi-square with two degrees of freedom and thus has a mean value of two, so it makes sense to find the contour for a value of *J* of two. Thus we need to find the orientation and size of the ellipse for the equation

$$\begin{bmatrix} xe \\ xn \end{bmatrix}^T \cdot \begin{bmatrix} p_{EE} & p_{NE} \\ p_{NE} & p_{NN} \end{bmatrix}^{-1} \cdot \begin{bmatrix} xe \\ xn \end{bmatrix} = 2 \tag{3.2}$$

We look at a rotation by an angle θ that will produce an equation for an ellipse in new variables *x* and *y* without cross terms. The transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = R(\theta) \cdot \begin{bmatrix} xe \\ xn \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} xe \\ xn \end{bmatrix} \tag{3.3}$$

where $R(\theta)$ is a rotation matrix that will rotate the vector *(xe,xn)* clockwise by an angle of θ. Its inverse is

$$R^{-1}(\theta) = R(-\theta) = R^T(\theta) \tag{3.4}$$

so that

$$\begin{bmatrix} xe \\ xn \end{bmatrix} = R(-\theta) \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.5}$$

We want to substitute (3.5) into (3.2) and use the result to find the value of θ that makes the off-diagonal terms zero. The substitution gives us

$$\begin{bmatrix} x \\ y \end{bmatrix}^T \cdot R^T(\theta) \cdot \begin{bmatrix} p_{EE} & p_{NE} \\ p_{NE} & p_{NN} \end{bmatrix}^{-1} \cdot R(\theta) \cdot \begin{bmatrix} x \\ y \end{bmatrix} = 2 \tag{3.6}$$

We note that, when the matrices *A*, *B*, and *C* are nonsingular,

$$\left(B \cdot A^{-1} \cdot C\right)^{-1} = C^{-1} \cdot A \cdot B^{-1} \tag{3.7}$$

so that the rotated covariance matrix is

$$R^{-1}(\theta)\cdot\begin{bmatrix} p_{EE} & p_{NE} \\ p_{NE} & p_{NN}\end{bmatrix}\cdot R^{-T}(\theta)=R(-\theta)\cdot\begin{bmatrix} p_{EE} & p_{NE} \\ p_{NE} & p_{NN}\end{bmatrix}\cdot R(\theta)=\begin{bmatrix}\cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta)\end{bmatrix}\cdot\begin{bmatrix} p_{EE} & p_{NE} \\ p_{NE} & p_{NN}\end{bmatrix}\cdot\begin{bmatrix}\cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta)\end{bmatrix} \quad (3.8)$$

$$=\begin{bmatrix} p_{EE}\cdot\cos(\theta)-p_{NE}\cdot\sin(\theta) & p_{NE}\cdot\cos(\theta)-p_{NN}\cdot\sin(\theta) \\ p_{EE}\cdot\sin(\theta)+p_{NE}\cdot\cos(\theta) & p_{NE}\cdot\sin(\theta)+p_{NN}\cdot\cos(\theta)\end{bmatrix}\cdot\begin{bmatrix}\cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta)\end{bmatrix}$$

$$=\begin{bmatrix} p_{EE}\cdot\cos^2(\theta)-2p_{NE}\cdot\cos(\theta)\cdot\sin(\theta)+p_{NN}\cdot\sin^2(\theta) & (p_{EE}-p_{NN})\cdot\cos(\theta)\cdot\sin(\theta)+p_{NE}\cdot\left(\cos^2(\theta)-\sin^2(\theta)\right) \\ (p_{EE}-p_{NN})\cdot\cos(\theta)\cdot\sin(\theta)+p_{NE}\cdot\left(\cos^2(\theta)-\sin^2(\theta)\right) & p_{EE}\cdot\sin^2(\theta)+2p_{NE}\cdot\cos(\theta)\cdot\sin(\theta)+p_{NN}\cdot\cos^2(\theta)\end{bmatrix}$$

Now we can write the rotated quadratic form as,

$$\begin{bmatrix} x \\ y\end{bmatrix}^T\cdot\begin{bmatrix} p_{xx} & 0 \\ 0 & p_{yy}\end{bmatrix}^{-1}\cdot\begin{bmatrix} x \\ y\end{bmatrix}=2 \quad (3.9)$$

where

$$\begin{aligned} p_{xx} &= p_{EE}\cdot\cos^2(\theta)-2p_{EN}\cdot\cos(\theta)\cdot\sin(\theta)+p_{NN}\cdot\sin^2(\theta) \\ p_{yy} &= p_{EE}\cdot\sin^2(\theta)+2p_{EN}\cdot\cos(\theta)\cdot\sin(\theta)+p_{NN}\cdot\cos^2(\theta) \\ 0 &= (p_{EE}-p_{NN})\cdot\cos(\theta)\cdot\sin(\theta)+p_{NE}\cdot\left(\cos^2(\theta)-\sin^2(\theta)\right)\end{aligned} \quad (3.10)$$

The equation for the off-diagonal term gives us θ as

$$\begin{aligned}\cos(2\theta) &= \frac{p_{EE}-p_{NN}}{\sqrt{(p_{EE}-p_{NN})^2+4\cdot p_{NE}^2}} \\ \sin(2\theta) &= \frac{-2\cdot p_{NE}}{\sqrt{(p_{EE}-p_{NN})^2+4\cdot p_{NE}^2}}\end{aligned} \quad (3.11)$$

There is a sign ambiguity in the square root that, if reversed, will rotate the ellipse by 90 degrees and interchange the semimajor and semiminor axes, which leaves the displayed localization ellipse unchanged. For the square root in (3.11) taken as positive, we have p_{xx} and p_{yy} as

$$\begin{aligned} p_{xx} &= \frac{1}{2}\cdot\left((p_{EE}+p_{NN})+\sqrt{(p_{EE}-p_{NN})^2+4\cdot p_{EN}^2}\right) \\ p_{yy} &= \frac{1}{2}\cdot\left((p_{EE}+p_{NN})-\sqrt{(p_{EE}-p_{NN})^2+4\cdot p_{EN}^2}\right)\end{aligned} \quad (3.12)$$

The semimajor and semiminor axes lengths are

$$\begin{aligned} x_{MAX} &= \sqrt{2\cdot p_{xx}} \\ y_{MAX} &= \sqrt{2\cdot p_{yy}}\end{aligned} \quad (3.13)$$

The semimajor axis is counterclockwise from the East axis by an angle of θ.

3.3 Plotting the Localization Ellipse

The equation of the localization ellipse, in a coordinate system rotated counterclockwise by an angle of θ,

$$\frac{x^2}{x_{MAX}^2}+\frac{y^2}{y_{MAX}^2}=1 \quad (3.14)$$

In polar coordinates,

$$x = r \cdot \cos(\phi)$$
$$y = r \cdot \sin(\phi) \tag{3.15}$$

the ellipse is traced out by the equation

$$r^2(\phi) = \frac{1}{\frac{\cos^2(\phi)}{x_{MAX}^2} + \frac{\sin^2(\phi)}{y_{MAX}^2}} \tag{3.16}$$

At the special angle ϕ_M given by

$$\cos(\phi_M) = \frac{x_{MAX}}{\sqrt{x_{MAX}^2 + y_{MAX}^2}}$$
$$\sin(\phi_M) = \frac{y_{MAX}}{\sqrt{x_{MAX}^2 + y_{MAX}^2}} \tag{3.17}$$

the distance from the center of the ellipse to the ellipse is

$$r^2(\pm\phi) = r^2(\pm\phi \pm \pi) = \frac{x_{MAX}^2 + y_{MAX}^2}{2} \tag{3.18}$$

This defines four points off the major axes. These four points, with the four points at the ends of the major axes, provides a basis for plotting the vertices of a stretched octagon that adequately depicts the localization ellipse for many display graphics.

4 Probability Distributions for Hypothesis Testing

4.1 Principles in Designing a Hypothesis Test

A hypothesis test is designed by defining a metric to be tested, such as bias in a noisy variable available in the simulation, defining a statistic using that metric that has a known probability distribution function, and setting a threshold determined by the desired or required confidence level. In most cases the probability distribution function needs to be available as a standard utility. A selection of the most common probability distribution functions is given in the next Section.

Most of the probability distribution functions are those of statistics defined from functions of zero-mean Gaussian random variables. All of these can be computed either by simple user-generated utilities or when an incomplete Beta function or an error function is available. Exceptions are the non-central chi square, F and t distributions. These have a non-centrality parameter that is the square of the mean of the variables divided by the variance. The non-central distributions have the added parameter of non-centrality which must be included in the definition of the threshold for the confidence limit. Probability distribution functions for the non-central distributions are not available because the extra parameter makes implementation of an accurate general-use algorithm much more difficult and they are rarely used in practice. Although requirements may drive us to use of non-central distributions, we should avoid them when we can by basing confidence limits on existence of a bias, as opposed to a bias of an *a priori* specified amount.

4.2 Distributions Most Used in Hypothesis Testing

Normal Distribution

The normal distribution occurs for Gaussian variables.

Chi-Square Distribution

A sum of υ squared Gaussian random variables with zero mean and unit variance is distributed according to the chi-square distribution with υ degrees of freedom.

The F-Distribution

The ratio of two chi-square distributed variables with υ_1 degrees of freedom for the numerator and υ_2 for the denominator is distributed according to the F-distribution. The normalization used for the standard definition of the F-distribution requires that both the numerator and denominator chi-square variables be divided by their means (which, since they are chi-square, is equal to the respective number of degrees of freedom) so that they have mean 1.

Student's t-Distribution

The ratio of a Gaussian random variable with zero mean and unity variance, divided by a chi-square variable with υ degrees of freedom that is itself divided by υ so that its mean is one, is distributed according to Student's t-distribution.

The Binomial Distribution

If a set of *n* events such as threshold crossings, each with probability *p* that the event will be scored a 1 and *(1-p)* that the event will be scored a 0, the probability that the score will be between 0 and *a* is distributed according to the binomial distribution.

Non-Central Distributions

Non-central distributions have been studied for the chi-square, F, and Student's t distributions.

4.3 The Incomplete Beta Function

The incomplete beta function is needed to find several probability distribution functions in this Section. It is defined by

$$I_x\left(a,b\right)=\frac{1}{B\left(a,b\right)}\cdot\int_0^x t^{a-1}\cdot\left(1-t\right)^{b-1}\cdot dt \tag{4.1}$$

where *B(a,b)* is the (complete) beta function,

$$B\left(a,b\right)=\frac{\Gamma\left(a\right)\cdot\Gamma\left(b\right)}{\Gamma\left(a+b\right)} \tag{4.2}$$

and $\Gamma\left(z\right)$ is the gamma function

$$\Gamma\left(z\right)=\int_0^\infty t^{z-1}\cdot\exp\left(-t\right)\cdot dt \tag{4.3}$$

An identity that is useful in working with the incomplete beta function is

$$I_x(a,b) = 1 - I_{1-x}(b,a) \tag{4.4}$$

See Abramowitz and Stegun, *Tables of Functions*, NBS AMS 55, second printing (1972) for methods to compute the incomplete beta function and its use in probability and spastics. It is available from the U.S. Government Printing Office or in paperback from Dover.

4.4 Table of Distribution Functions

The table below for relationships that can be used to compute most of the distributions used here.

Distribution	***p(x)***	***P(x)***	***Q(x)***
Normal	$\frac{1}{\sqrt{2\pi}\cdot\sigma}\cdot\exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)$	$P(x)$	$Q(x)$
Chi-Square	$\frac{1}{2^{\upsilon/2}\cdot\Gamma\left(\frac{\upsilon}{2}\right)}\cdot x^{\upsilon/2-1}\cdot\exp\left(-\frac{t}{2}\right)$	$P\left(\chi^2\middle\|\upsilon\right)=\int_0^{\chi^2} p(t)\cdot dt$	$Q\left(\chi^2\middle\|\upsilon\right)=\int_{\chi}^{\infty} p(t)\cdot dt$
F-Distribution	$\frac{\upsilon_1^{\upsilon_1/2}\cdot\upsilon_2^{\upsilon_2/2}}{B\left(\frac{\upsilon_1}{2},\frac{\upsilon_2}{2}\right)}\cdot x^{\frac{\upsilon_1-1}{2}}\cdot(\upsilon_2+\upsilon_1\cdot x)^{-\frac{\upsilon_1+\upsilon_2}{2}}$	$P\left(F\middle\|\upsilon_1,\upsilon_2\right)=I_{1-x}\left(\frac{\upsilon_1}{2},\frac{\upsilon_2}{2}\right)$ $x=\frac{\upsilon_2}{\upsilon_2+\upsilon_1\cdot F}$	$Q\left(F\middle\|\upsilon_1,\upsilon_2\right)=I_{x}\left(\frac{\upsilon_2}{2},\frac{\upsilon_1}{2}\right)$ $x=\frac{\upsilon_2}{\upsilon_2+\upsilon_1\cdot F}$
Student's t	$\frac{d\,P(x)}{dx}$	$\frac{1}{\sqrt{\upsilon}\cdot B\left(\frac{1}{2},\frac{\upsilon}{2}\right)}\cdot\int_{-x}^{x}\left(1-\frac{t}{\upsilon}\right)^{-\frac{\upsilon+1}{2}}\cdot dt$	$Q\left(t^2\middle\|1,\upsilon\right)$

5 Assignment

Examine the scalar classical example from last time, the maximum likelihood estimators for unknown mean and variance for a scalar distribution. Work though it until you understand it completely and are comfortable with all the steps.