DATA FUSION III: Estimation Theory

Date: March 30, 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 Course Closure

This is the last formal class for Data Fusion III. Closure plans are:

- A quiz to be handed out at the end of today's class.
- A study session one week from today, April 6.
- The quiz will be due to me by COB one week later, April 13.
- I will have your grades posted by April 20.

1.1 Course Evaluation Form

At the end of today's class I will hand out course evaluation forms. Please do NOT mark them while I am in the room. Fill them out and send them Steve Chin at Rowan either by e-mail (scan / pdf) chin@rowan.edu or by Fax : 856 256 5350.

1.2 The Quiz

We will have a full lecture today, and I will make the quiz available to you at the end of the class. This will be a take-home quiz with four problems, each multiple-part. Each of the four problems will be 25% of your grade.

You are encouraged to ask questions by e-mail to jkbeard@comcast.net (preferred) or beard@rowan.edu. Any responses to questions about the quiz problems will be by broadcast e-mail to all students.

One week from today, on April 6, we will have a study session in this room from 5:00 PM until 7:30 PM. At the study session, I may, if appropriate, begin with a short presentation of material that answers questions posed to me up to that time. The intent of this session is to allow face-to-face questions and answers with the whiteboard available to help you understand the quiz questions and how to proceed. I will NOT give solutions to the problems or otherwise compromise the quiz process. The best use of this session is for you to have worked through all of the quiz problems and be ready with questions. Your public questions must avoid giving information on solutions; if you have such a question, please give it to me on a piece of paper. Any response will be to all students.

The quiz is due two weeks from today, by 5:00 PM on Thursday, April 13. Please send it by e-mail if you can. If you must have a hard copy only, let me know by e-mail and I will make arrangements for it to be scanned to PDF here at WJHTC or for someone to collect your quiz for me to pick up. I will have it graded and an e-mail response to you that includes a PDF grading sheet and your grade within a week.

2 The Covariance Mapping Equation

The covariance mapping equation is one that is useful in most analyses. We have seen it in the extended Kalman filter and in most batch estimators.

2.1 Linear Relationship Between Vectors

We begin with a vector $\underline{v}$ of correlated Gaussian variables with zero mean,

$$\begin{aligned} \langle \underline{v} \rangle &= \underline{0} \\ \mathrm{Cov}\{\underline{v}\} &= \langle \underline{v} \cdot \underline{v}^T \rangle = R_v \end{aligned} \tag{2.1}$$

We have a linear relationship between $\underline{v}$ and another vector $\underline{w}$,

$$\underline{w} = A \cdot \underline{v} \tag{2.2}$$

where the matrix A is not necessarily square; the lengths of the vectors $\underline{v}$ and $\underline{w}$ may be different. We can map the covariance of $\underline{v}$ to the covariance of $\underline{w}$ by simply noting the definition of the covariance,

$$\begin{aligned} \mathrm{Cov}\{\underline{w}\} &= \langle \underline{w} \cdot \underline{w}^T \rangle = \langle A \cdot \underline{v} \cdot \underline{v}^T \cdot A^T \rangle \\ &= A \cdot \langle \underline{v} \cdot \underline{v}^T \rangle \cdot A^T \\ &= A \cdot R_v \cdot A^T \end{aligned} \tag{2.3}$$

2.2 Nonlinear Relationship between Vectors

Many times we have a functional relationship between vectors,

$$\underline{y} = \underline{f}(\underline{x}) + \underline{v} \tag{2.4}$$

We can map the covariance of the errors in $\underline{x}$ and $\underline{v}$ to the covariance in $\underline{y}$ through a linearization of the equation. We note that $\underline{f}(\underline{x})$ can be written as a Taylor expansion about the mean $\underline{x}_0$ of $\underline{x}$ as

$$\begin{aligned} \underline{f}(\underline{x}) &\approx \underline{f}(\underline{x}_0) + \left[\frac{\partial \underline{f}(\underline{x})}{\partial \underline{x}} \right]\Bigg|_{\underline{x}=\underline{x}_0} \cdot (\underline{x} - \underline{x}_0) + \underline{v} \\ &= \underline{f}(\underline{x}_0) + F \cdot (\underline{x} - \underline{x}_0) + \underline{v} \end{aligned} \tag{2.5}$$

where the matrix F is the gradient in the first line of (2.5). Here we have used only the first-order term in the Taylor series, and this approximation defines the limitations on the use of the covariance mapping equation for nonlinear relationships. For example, quadratic terms in the Taylor series will introduce a bias, or shift in the mean, of the mapped vector through a square-law rectification effect. The magnitude of higher order derivatives must be understood when using this approximation to ensure that you understand the implications of its use.

The covariance R_y of $\underline{y}$ is easily obtained from the definition of its covariance and the linearization we show in (2.5),

$$\begin{aligned} R_y &= \left\langle \left(\underline{y} - \underline{f}(\underline{x}_0) \right) \cdot \left(\underline{y} - \underline{f}(\underline{x}_0) \right)^T \right\rangle \\ &= F \cdot R_x \cdot F^T + R_v \end{aligned} \tag{2.6}$$

You will recognize (2.3) and (2.6) in the Kalman filter covariance extrapolation and update, and in the simpler derivations of estimate covariances in batch estimators.

3 Markov Processes

3.1 Definition

Markov processes are simply sequences of vectors of random variables that are produced by a linear relation and an inhomogeneity, also a random vector. We will call the Markov sequence $\underline{y}_i$ and the driving vector $\underline{v}$. The sequence is defined by an initial covariance R_{y0}, a transition matrix A, a driver noise mapping matrix B, and the covariance R_v of the driving vector $\underline{v}$. The Markov sequence is

$$\underline{y}_{i+1} = A \cdot \underline{y}_i + B \cdot \underline{v} \tag{3.1}$$

The covariances of the Markov sequence are easily found by the covariance mapping equation to be

$$Cov\left\{\underline{y}_{i+1}\right\} = R_{y,i+1} = A \cdot R_{y,i} \cdot A^T + B \cdot R_v \cdot B^T \tag{3.2}$$

We recognize that the state vector error in a Kalman or alpha-beta tracker is a Markov process.

3.2 Stability

The stability of a Markov process is defined as the condition that, if R_v is zero, that $R_{y,i}$ be bounded as i increases without bound. We can make this condition quite clear by looking taking R_v as zero and look at the determinant of both sides of (3.2):

$$\left|R_{y,i+1}\right| = \left|A\right|^2 \cdot \left|R_{y,i}\right| \tag{3.3}$$

We can say that a sufficient condition for unconditional stability of (3.1) is that all the eigenvalues of A have a magnitude less than 1.

The mechanics of Markov processes for continuous cases, where the difference equation (3.1) is replaced by a differential equation, is treated in Section 3.7 of Gelb, “Propagation of Errors” on page 75 through 78.

4 Interactive Multiple Models (IMM)

4.1 Definition

Interactive multiple models (IMM) are used in the extrapolation and update process of a Kalman filter. The concept is that the radar contact is in one of a few distinct maneuvering modes, such as left turn, right turn, or straight-ahead flight, and that the relative probability that the contact is in each mode is known. A properly designed IMM tracker will use the proper model, improving performance over that of a tracker that does not model distinct maneuver characteristics separately. In addition, the analysis of the maneuvers done as part of the IMM functions can be used as track file information for use in C2 functions such as alerts.

4.2 Thumbnail of Operation

When a radar contact is associated with a track file, the target motion is extrapolated forward to the time of the radar contact and three updates are made. The probability that the contact is in each mode is updated according to how accurately the target motion extrapolation predicts the position as measured by the radar contact. State vector updates are computed for all models and the tracker state vector is taken as the mean of these updates, as weighted by the relative probabilities of the motion models. The covariance is mapped from the definition of the state vector.

4.3 Discrete Markov Process: The Enabling Technology

A discrete Markov process is quite distinct from a Markov process (without the "discrete"). The Markov process as presented in 3 above is a set of vectors of random variables – i.e., noise. The discrete Markov process here consists of probabilities, not random variables.

Markov matrices and discrete Markov processes were originally developed for modeling of economic trends and phenomena. They have enormous utility in modeling and simulation in such areas as modeling terrain obscuration of a line of sight for Monte Carlo scenario simulations. Here we use discrete Markov processes to model different possibilities of radar contact behavior between tracker updates.

4.3.1 The Probability Vector

Discrete Markov process theory is based on the concept of a probability vector. A probability vector is a set of probabilities that a system is in a one of a set of mutually exclusive states, such as left turn, right turn, or straight-ahead flight. Its elements are nonnegative and sum to 1. Thus, for a probability vector $\underline{p}$ we write

$$\underline{1}^T \cdot \underline{p} = 1 \tag{4.1}$$

4.3.2 The Markov Matrix

A probability vector $\underline{p}_i$ can be propagated to another probability vector $\underline{p}_{i+1}$ by a linear transformation:

$$\underline{p}_{i+1} = M \cdot \underline{p}_i \tag{4.2}$$

We define the transition matrix M as a square matrix whose columns are probability vectors. Each element of column n of the Markov matrix M is the probability that the system will change from state "n" to state "m" during the transition between time t_i and time t_{i+1}. The diagonal elements of M are the probability that the system will remain in the same state. We stipulate that all the possible states are represented, so that the sum of the probabilities that the system will remain in state "j" and change to another state is 1. Since each column of M is itself a probability vector, we write

$$\underline{1}^T \cdot M = \underline{1}^T \tag{4.3}$$

If the elements of a matrix M are nonnegative and (4.3) holds, its columns are probability vectors and thus M is a Markov matrix. Thus we can define a Markov matrix as any square matrix of nonnegative elements for which (4.3) holds.

4.3.3 Multiplication of Markov Matrices

We now address the product of two Markov matrices. Each column of the product of $M_2 \cdot M_1$ is the product of M_2 a column of M_1. We see from (4.2) that this means that each column of the product is a probability vector, so the product of two Markov matrices is a Markov matrix.

4.3.4 Characteristic Values and Repeated Products of Markov Matrices

We see from (4.3) that at least one characteristic value of a Markov matrix is 1, and that the left eigenvector for that eigenvalue has elements that are all equal to one another. A Markov matrix that has no zero elements has one and only one characteristic value equal in magnitude to 1, and its value is +1, and the magnitudes of all other characteristic values is less than 1. Thus the limit

$$\underline{p}_\infty = \lim_{p \to \infty} M^p \cdot \underline{p}_0 \tag{4.4}$$

exists and is independent of the initial probability vector $\underline{p}_0$. This means that if the Markov matrix is static, the probability vector will have a steady state that is independent of the initial probability vector.

If even one element of M is zero, this is not necessarily true. As an example, consider the Markov matrix

$$MT_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{4.5}$$

Clearly each multiplication by MT_2 will reverse the positions of the elements in a two-element probability vector, and there is no steady state.

4.3.5 Continuous Propagation of Probability Vectors

We treat some radar contact models as differential equations or closed form expressions rather than difference equations, so we complete this material on Markov matrices and probability vectors with the continuous case. We consider the case of an invariant differential equation for the probability vector,

$$\frac{d\,\underline{p}(t)}{dt} = A \cdot \underline{p}(t) \tag{4.6}$$

which has the solution

$$\underline{p}(\quad) = \exp\left(A \cdot \left(t - t_0\right)\right) \cdot \underline{p}\left(t_0\right) \tag{4.7}$$

We know that $\exp(A \cdot x)$ must be a Markov matrix, so

$$\underline{1}^T \cdot \exp\left(A \cdot \Delta t\right) = \underline{1}^T \cdot \left(I + \sum_{p=1}^{\infty} \frac{\left(A \cdot \Delta t\right)^p}{p!} \right) = \underline{1}^T \tag{4.8}$$

so we know that

$$\underline{1}^T \cdot A = \underline{0}^T \tag{4.9}$$

This is consistent with the definition of the columns of *A* as the differential probability that the system will transition from state "j" to state "i" which means that the sum of the elements in each column of *A* must be zero.

We use the continuous model in our IMM formulation below to allow nonuniform time increments between updates, although the discrete formulation is possible.

We define the elements of *A* as follows.

$$a_{mn} \cdot \Delta t = \left\langle \text{probabilty that state n transitions to state m in time } \Delta t \right\rangle \tag{4.10}$$

$$1 - a_{nn} \cdot \Delta t = \left\langle \text{probabilty that state n remains in state n in time } \Delta t \right\rangle \tag{4.11}$$

4.4 Bayesian Update of the Probability Vector

We begin with a state vector $\hat{\underline{x}}\left(t_-\right)$ and covariance matrix P_- at the last tracker update, and a set of radar measurements at current time $\underline{y}$ as with trackers that don't have IMM. We also have a set of *S* distinct motion models for the radar contact that we will use, and the probability or likelihood that each is the correct model at the time of the last update collected as a probability vector $\underline{p}\left(t_-\right)$.

We begin by updating the probability vector,

$$\tilde{\underline{p}}\left(t\right) = \exp\left(A \cdot \left(t - t_-\right)\right) \cdot \underline{p}\left(t_-\right) \tag{4.12}$$

We also extrapolate the state vector from the last update using each of the motion models and compute a likelihood from each of the updates:

$$la_s = \frac{1}{\left(2\pi\right)^{K/2} \cdot \left|E_s\right|^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot \underline{e}_s^T \cdot E_s^{-1} \cdot \underline{e}_s \right), s = 1 \ldots S \tag{4.13}$$

where $\underline{e}_s$ is the tracker error for motion model *s* and E_s is its covariance,

$$\underline{e}_s = \underline{y} - \underline{h}\left(\tilde{\underline{x}}_s\right), E_s = H \cdot \tilde{P}_s \cdot H^T + R \tag{4.14}$$

We update each element of the probability vector using Bayes' theorem,

$$p_s\left(t\right) = P\left(s \middle| \underline{y}\right) = \frac{P\left(\underline{y} \middle| s\right) \cdot \tilde{p}_s\left(t\right)}{\sum_{k=1}^{S} P\left(\underline{y} \middle| k\right) \cdot \tilde{p}_k\left(t\right)} = \frac{la_s \cdot \tilde{p}_s\left(t\right)}{\sum_{k=1}^{S} la_s \cdot \tilde{p}_k\left(t\right)} \tag{4.15}$$

We implement (4.15) by weighting the extrapolated probability vector with the update likelihoods to form an unnormalized probability vector $\underline{p}_U$ and then normalizing it:

$$\underline{p}_U = \begin{bmatrix} \vdots \\ la_s \cdot \tilde{p}_s \\ \vdots \end{bmatrix}, \underline{p}(t) = \frac{1}{\underline{1}^T \cdot \underline{p}_U(t)} \cdot \underline{p}_U(t) \tag{4.16}$$

4.5 Updating to a Single State Vector

At this point, we have S state vectors, covariance matrices and probabilities. The tracker is to use only one state vector and one covariance. We unify the state vector to form the tracker updated state vector through use of the Bayesian mean,

$$\underline{\hat{x}}(t) = \sum_{s=1}^{S} P\left(k \middle| \underline{y}\right) \cdot \underline{\hat{x}}_s = \sum_{s=1}^{S} p_s(t) \cdot \underline{\hat{x}}_s \tag{4.17}$$

This is the mean state vector from the updates using the *S* motion models, weighted by their respective probabilities.

4.6 Updating to a Single Covariance Matrix

We find the covariance matrix for the updated state vector by mapping the errors using (4.17). The result is

$$\begin{aligned} P(t) &= \left\langle \left(\underline{\hat{x}} - x\right) \cdot \left(\underline{\hat{x}} - x\right)^T \right\rangle \\ &= \sum_{x=1}^{S} p_s(t) \cdot P_s + \sum_{s=1}^{S} p_s(t) \cdot \left(\underline{\hat{x}}_k - \underline{\hat{x}}\right) \cdot \left(\underline{\hat{x}}_k - \underline{\hat{x}}\right)^T \end{aligned} \tag{4.18}$$

4.7 Practical Aspects of IMM

Note that the covariance update as given by (4.18) will provide a very good update if one of the probabilities is quite near 1, but it can muddy the covariance matrix quite a bit if one of the models is not dominant. The dominance of one model over another is determined by the relative likelihoods as computed according to (4.13), which depends on the tracker error for each motion model. For IMM to improve tracker performance, the motion models must provide distinct differences in update likelihood as measured by (4.13). If this does not occur, the probabilities will be spread without dominance by the correct model and the use of IMM will actually decrease tracker performance.

Key points in applying IMM include:

- Use IMM only when the radar contact has more than one possible motion model that will result is significantly different tracker errors as measured by (4.13).
- Assign reasonable values to the initial probability vector $\underline{p}_0$ at the time you transition from single-model tracking to IMM tracking.
- Make sure that you can properly assign transition probabilities for the elements of *M* in (4.2) or *A* in (4.6) as appropriate.

- When one model clearly dominates, drop the other target motion models and use that model in single-model tracking until a radar contact motion model change is likely.

The bottom line is that IMM helps when the radar contact motion model is so uncertain as to essentially invalidate the constant velocity or other motion model of the tracker, but the motion model is not known – and the differences between the motion models will become clear in an update or two through the la_s values measured in (4.13)

5 Multiple Hypothesis Tracking (Thumbail; see Handout)

Multiple hypothesis trackers (MHT) are trackers that allow association of returns with more than one track file to make sure that the correct track file gets updated by the right return every time. This forces tracks of false contacts, missassociations, and other "ghost" tracks. The MHT handles this by keeping a score function of each track file that amounts to a cumulative sum of association likelihood, and reports a track to C2 only when the score function meets or exceeds a minimum threshold.

Many other technologies are necessary for successful MHTs. These include

- Score functions must allow declaration of true radar contacts to C2 on the first or second hit.
- Duplicate tracks must be pruned or merged with true tracks to prevent duplicate tracks from being reported to C2.
- Pruning of spurious tracks must be used to successfully keep the number of track files within the design limits of the data processor.

See separate handout for details of design, implementation, and analysis of MHTs.

6 When you Need Each Method

6.1 When you need a Batch Estimator

A batch estimator is indicated when:

- The requirement for estimation can be posed as a set of parameters that do not change over the term of the database, as an aircraft position and velocity.
- Motion of an object in track can be posed without process noise, as with ballistic or orbital objects, an aircraft at cruising altitude in straight, level flight on autopilot, or other tracking problems where you find yourself driving your process noise to zero and encountering covariance collapse with a Kalman filter.
- When you need to achieve highest possible accuracy with the available data, or specifically are required to approach the Cramer-Rao bound for minimum variance of estimation errors.
- A batch estimator is clearly the simplest solution to the problem.

6.2 When you need a Kalman Filter

When you are tracking an object with a motion model that involves process noise, as with an aircraft that is manually piloted, or is flying in weather, and the only accurate model of its motion involves process noise, you need a recursive estimator that incorporates this type of motion model. The generic name for this type of estimator is a Kalman filter.

A Kalman filter is a base technology, and an introduction and survey of this base technology is given in Gelb. Like any other base technology such as Laplace transforms, finite element analysis, or Monte Carlo scenario simulation, Kalman filter technology is a cornerstone of an approach, and its utility depends on how it is used along with other tools and technologies to help solve a particular problem. Successful use of a Kalman filter depends on success in solving basic problems such as aircraft motion modeling, estimation of radar noise errors, dealing with of correlation in measurements made available to the tracker – either between measurements or over time, recognition and handling of biases in the sensors, and other issues.

6.2.1 Evolution of Application of Kalman Filtering

Because Kalman filters are a base technology like least squares estimators, and because they provided an approach to a problem that had not been previously successfully dealt with – estimation of states of a process involving noise-driven motion, some of the base problems that are required to use a Kalman filter had no history or basis for application. In addition, the numerical issues in the variance analysis implicit in the Kalman filter were unknown in its inception. A timeline showing some events in it application illustrates the evolution of applications of applications using the Kalman filter:

- 1960 – Kalman first published the recursive estimator for noise-driven states; the community named it the Kalman filter, recognizing it as a fundamental advance over the Wiener filter.
- 1962 – Kalman and Bucy extended the Kalman filter to nonlinear state propagation dynamics and measurements; this is the extended Kalman filter or EKF. Nearly all recursive estimators that we call Kalman filters today are EKFs.
- 1972 – The primary Aegis sensor, the AN/SPY-1 requirements frozen. An alpha-beta tracker was used instead of a Kalman filter because it met requirements best among the technologies available at that time.
- 1975 – The radar community and others recognized that the “short form” of the Covariance update universally caused numerical problems and eschewed its use, making the Kalman filter numerically practical for use.
- 1977 – Gerald Bierman’s book on square root filters brought this technology from NASA to the entire community, providing a selection of “big hammers” for numerical issues of Kalman filters.

- 1980 – Statistical efficiency of estimators and other principles of mathematical statistics and estimation theory became more commonly associated with materials and publications involving Kalman filters
- 1985 – Computational requirements for the Kalman filter extrapolation and update became passé as the i860, i960, TMS320C series, and other products of the VHSIC program became available. The association problem became recognized as the problem to solve in tracking, not the Kalman filter.
- 1988 – The multiple hypothesis tracker (MHT) emerged as the core of various track-before-detect schemes based on "score functions" that kept track of the likelihood that a track was based on a real radar return. This technology was developed to deal with high target densities with significant association ambiguities, false alarm environments in high interference environments, jamming problems and heavy EW environments, and high false alarm sensors such as EO/IR sensors, and with emerging technologies such as ground moving target indicator (GMTI) radars. Samuel Blackman's first book is published, which thoroughly explains the MHT in one document for the first time.
- 1990 – Interactive multiple models (IMM) was introduced as a method for dealing with intensively maneuvering targets such as high performance aircraft, boosting missiles that abruptly change or discontinue thrust, or that maneuver with high, variable acceleration.
- 2000 – Diversity of tracker technologies is documented in Samuel Blackman's second book, with Robert Popoli, on tracker technologies to that date.

These dates are notional and the events were selected casually to make the point that Kalman filtering is a base technology, and the technology of application of Kalman filtering is a vast set of technologies first enabled by the introduction of the Kalman filter in 1960. The evolution of these technologies continues.

6.2.2 Problems and Solutions with Kalman Filter Applications

Problems and solutions that commonly occur when first applying a Kalman filter include, but are not limited to, the following:

a) A tracker may operate properly for a period, when its estimate of radar contact position suddenly exhibits an exponential divergence from the true position. An adaptive response to tracker error, if present, will increase the covariance of the process noise and the tracker will lock on again, only to diverge suddenly after a few minutes. This is covariance collapse and the divergence results when the covariance matrix in the Kalman filter is no longer positive definite – it has a negative eigenvalue. Usually this happens when someone is using the "short form" for the covariance update. If the Joseph stabilized form does not prevent covariance collapse, a square root filter should be considered, or a batch estimator may be more appropriate.

b) The tracker's covariance matrix shows an error covariance that is much larger than the actual tracker error, as measured by use of a Monte Carlo simulation or from real data with radar contact truth data such as INS or GPS data. This is very common and occurs when process noise is used to prevent covariance collapse. The purpose of process noise in the radar contact motion model is to accurately model the motion, not to address numerical issues, so the process noise should be reduced and the numerical issues addressed appropriately, such as by use of the Joseph stabilized form for covariance update, the normal form for the covariance update, or one of the square root filters.

c) The tracker position estimate shows a wandering about the true target position that is much larger than the covariance estimate. No amount of tracker tuning can prevent this, and the data seems to support the wandering motion, but the truth data or other information does not. This also is very common, and happens when the data made available to the tracker is filtered in some way, so that the errors from measurement to measurement are correlated. The solution is measurement differencing, which removes the correlation between measurements, with some minor complication to the implementation of the Kalman filter. Another solution to the problem of correlated measurements can be the use of a batch estimator, which naturally includes correlations between measurements as part of its base formulation.

d) The variance of velocity or acceleration states tends to zero, causing the Kalman filter to coast that state, which can cause large swooping deviations of the tracked position from the true position because of persistent deviations of the velocity or accelerations from those of the tracked radar contact. This will happen if your process noise is zero for the highest time derivative in your state vector – velocity states in a nonaccelerating motion model, for example. You *must* have nonzero process noise in the highest time derivative states in the motion model or the covariance of those states will tend to zero. You should consider mapping the square of the error in one of the radar measurements to this process noise; see the simple two-state tracker presented earlier and used in the Excel simulation presented in the first week for an example.

e) The process noise can't be made small enough to avoid driving tracker accuracy, even though the Joseph stabilized form or the normal form is used for the covariance update and covariance collapse is avoided. This means that normal numerical error is the noise floor of your implementation and you should consider a longer word length, or use one of the square root filters, or you should look at a batch estimator that does not use process noise in the state vector motion model.

6.3 When you Need IMM Tracking

When high performance tracking is required, and radar contacts are subject to maneuver changes that affect tracker performance, the use of IMM tracking can essentially cost by multiple maneuver models and let the update likelihood determine which target behavior is being tracked. Examples are combat, exhibition, acrobatic, or crop-duster aircraft. The tracker can adaptively switch from single-model tracking to IMM tracking when erratic radar contact maneuvers are detected through consistently lower association likelihoods.

6.4 When you Need MHT Tracking

Whenever you have a dense radar contact environment that forces association errors but high tracker performance is required, tracker performance and reliability can be improved through use of MHT tracking. Examples of situations where this is true include

- Very dense radar contact traffic, with crossing tracks that have similar velocities and directions but different altitudes, but altitude information is not directly available as part of the radar contact data.
- Dense radar contact environment with interference that causes the radar to report false contacts frequently enough so that filtering these out of the C2 data is required or desirable.
- Data merging situations in dense environments where significant ambiguities exist, such as two closely spaced, parallel moving targets seen by two radars with good range accuracy but fair angle accuracy, where triangulating between the two radars' range measurements gives four possible positions for the two targets.
- Use in sensor situations where a high false alarm rate is unavoidable, such as high interference, close proximity to similar radars, or high sensitivity EO/IR sensors.

6.5 When you Need a Square Root Filter

You need a square root filter when you have one of these problems:

- Your process noise cannot be made small enough to maximize tracker accuracy while preventing covariance collapse. You may want to look at a batch processor if the process noise should be modeled as zero, but a square root filter (or the use of double precision, or both) will increase the numerical capabilities of your Kalman filter.
- You have an observability problem in one of your states, particularly in the early part of the track, as when bearings-only data is used to estimate range of a radar (or sonar, or EO/IR) track. The normal form (a batch estimator based on the information matrix instead of the covariance matrix) is best for this type of problem, but if you must have process noise in the state vector motion model you need a Kalman filter. The square root information filter (SRIF) is an excellent solution for problems of this type.
- You have a working Kalman filter. Your analysis of the software shows that your numerical issues are under control but your margin (effective number of bits of significance in critical computations) is small. Double precision is not considered to be the best solution, or is already being used. You can drop in a UDUT square root filter for a Kalman filter in most applications and add numerical margin almost equal to your word length.
- You have a new problem and want to begin with the most robust Kalman filter formulation possible. The SRIF is probably the best choice for new applications

requiring maximum robustness. When observability of all states is not an issue at any point in the track, a UDUT will also perform well.

7 Batch Processing for Maneuver Detection (Overview)

For relatively benign radar contacts such as transport aircraft, the key to high performance tracking using maneuver detection is to

a) Characterize the maneuver with as few parameters as possible, such as time of turn, turn radius, and turn speed reduction.

b) Use an ordinary tracker to track the target, but keep a history of a few minutes of raw radar returns that have been associated with the target track, and use a batch estimator on this raw data history to estimate maneuver parameters.

c) When a maneuver has been detected, restart the tracker at the point where the maneuver occurred, correct the target motion model, and update the tracker sequentially with the data history from the time of maneuver to bring it back current.

d) Keep the batch estimator running to determine when the turn stops and repeat the process with this maneuver change.

A simple example where we characterize the radar contact behavior as a straight line up to a time of maneuver, at which time we see a velocity change and a constant radius turn. This can be characterized for 2-D radars using complex variables as

- A position z_0 at some point in the straight line portion of the flight.
- The velocity and heading during the straight line portion of the flight, expressed as a complex velocity $\dot{z}$.
- The center of the circle during the turn z_C .
- The velocity during the turn v_M.

Thus, the target position $z(t)$ can be modeled as

$$z(t)\begin{cases} = z_0 + \dot{z}\cdot(t - t_0),\, t \le t_M \\ = z_C + r_m \cdot \exp\left(j \cdot \omega_M \cdot (t - t_m)\right),\, t > t_m \end{cases} \tag{4.19}$$

where

$$\begin{aligned} z_M &= z_0 + \dot{z}\cdot(t_m - t_0) \\ r_m &= z_m - z_C \end{aligned} \tag{4.20}$$

Our state vector is

$$\underline{x} = \begin{bmatrix} \mathrm{Re}\{z_0\} \\ \mathrm{Im}\{z_0\} \\ \mathrm{Re}\{\ddot{z}\} \\ \mathrm{Im}\{\dot{z}\} \\ t_m \\ \mathrm{Re}\{z_C\} \\ \mathrm{Im}\{z_C\} \\ \omega_m \end{bmatrix} \tag{4.21}$$

Our measurement model $\underline{h}(\underline{x})$ is the radar measurements in terms of $z(t)$ as given by (4.19) and we apply an MLE batch estimator as presented previously.

8 References

(1) Bellman, Richard, *Introduction to Matrix Analysis*, Second Edition, SIAM Press (1997), ISBN 0-89871-399-4 (reprint of 1970 McGraw-Hill edition), Chapter 14 “Markoff Matrices and Probability Theory” pp. 263-280.

(2) Gerald J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Academic Press, ISBN 0-12-097350-2 (1977).

(3) Yaakov bar-shalom and Xiao-Rong Li, *Estimation and tracking: principles, techniques and software*, Artech house (1993).

(4) Yaakov bar-shalom and Xiao-Rong Li, *Multitarget-Multisensor tracking: principles and techniques*, ISBN 0-9648312-0-1 (1995).

(5) Samuel Blackman and Robert Popoli, *Design and analysis of modern tracking systems*, Artech house (1999).