Target Tracking

Jason F. Ralph
Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool, UK

1 INTRODUCTION

Tracking algorithms are an essential part of any modern radar or electro-optical sensor system. Tracking algorithms have many military applications: including missile guidance, targeting and geo-location, ballistic missile defense, counter battery fire, airborne early warning systems, reconnaissance and surveillance. There are many important applications outside defense: air traffic control, surveillance systems for law enforcement, communications systems, traffic flow management and – more recently – the analysis of football matches and other sporting events. However, this section will tend to concentrate on the military aspects of the tracking problem because it is these that have tended to drive the development of modern tracking techniques.

The function of the tracking algorithm is to allow a sequence of noisy measurements to be interpreted – to provide a consistent set of trajectories for all of the objects detected by the sensor. The trajectories (or tracks) correspond to a history of detections/measurements for each object – which can be used to determine or to confirm the identity of the object, to predict the future motion of the object, or to determine where is it likely to have come from. The location, speed, or maneuverability of an object can be used to determine whether an object is a possible threat (e.g., an in-coming missile), or a potential target (an in-coming aircraft for air defense systems). Such information is also useful to determine whether an object is benign – a false alarm generated by the sensor. Predicting the motion of a threat or target is important when optimizing an engagement: for example, determining the type of countermeasure or which maneuver to employ for a missile defense system or improving the probability of an intercept for a guided weapon.

Tracking algorithms are most commonly used with radar and electro-optical (EO) sensors (e.g., infrared imagers). These sensors can be used to detect aircraft, missiles, and other vehicles, and measure their position (relative to the sensor) and aspects of their motional state. A radar system normally provides relatively accurate range information (from the transmission–reception delay), a less accurate angular position, and often the radial speed of the object (from a measurement of the Doppler shift of the carrier wave). Passive electro-optical systems normally provide accurate angular position measurements, while range information needs to be inferred from sequences of angle measurements (passive ranging), or multiple views of the same object (stereo vision). Both methods for estimating range from passive EO measurements tend to be less accurate than radar systems, but they can be supplemented by the addition of a laser range finder, at the expense of making the sensor active and easier to detect.

Each sensor produces a series of detections and noisy measurements which need to be "pieced together" to form a set of

Encyclopedia of Aerospace Engineering.
Edited by Richard Blockley and Wei Shyy
 ISBN: 978-0-470-68665-2

continuous trajectories, one for each object being monitored. Measurements from radar and EO sensors vary in terms of noise and the measured quantities, but the structure of the algorithms, and the techniques used to construct the tracks follows the same pattern. Measurements of objects are taken and then associated with one of the existing tracks. If the measurement is not sufficiently close to an existing track to be one of the previously detected objects, a new track can be created. The measurement and its associated track are then passed to a state estimation algorithm so that the track states (position, velocity, etc.) can be updated. Any track that does not have a measurement associated with it can be updated based on previous measurements or deleted, if no suitable measurement has been associated with the track for a period of time. The state estimation algorithm – see section 2.2 – is at the heart of any tracking system, but without the ancillary algorithms for measurement-track association and track maintenance (creation and deletion), a tracker will tend to generate spurious or sub-optimal tracks whenever the sensor detects a false alarm or multiple targets. The mis-association of measurements with tracks reduces the accuracy of the state estimates and false alarms can generate enough tracks to affect the computational efficiency of the tracker and increase the risk of mis-association.

The literature for target tracking is large, with large numbers of research papers covering the fundamental algorithms and application-oriented papers making use of the techniques. However, for a general overview of the subject material, there are several excellent reference books which cover the main aspects of tracking and state estimation (Blackman, 1986; Bar-Shalom, Li and Kirubarajan, 2001).

2 SINGLE TARGET TRACKING

This section starts by introducing a simple problem – a single target with a reliable sensor – and expanding this to include intermittent detections and false alarms. The aim is to develop an understanding of each of the elements in turn and to build toward a robust modern tracking system. Central to this are the sensor measurement and the ability to observe all of the quantities that are contained in the target/track state vectors. Given a set of measurements, the target state and the expected accuracy of that state can be estimated using a state estimation filter (the Kalman filter), which can then be used to reduce the problems associated with imperfect target detection and the appearance of false alarms.

2.1 Measurement and state estimation

The main aim of a target tracking algorithm is to obtain estimates for parameters that describe the location and motion of objects that have been detected by a sensor. These estimates should form a continuous history or track that represents the motion of the object. More precisely, it represents the tracking system's *knowledge* of the motion of the object. The sensor may not measure all of the motional states directly, some of the states must be inferred from other directly measured states, for example, a radar measuring the relative position of a target can infer the three-dimensional velocity of an aircraft from the change in its position as a function of time – which could be augmented using the radial velocity of the target, calculated from the Doppler shift of the carrier wave. The number of motional states included in the state vector varies from application to application. Most tracking algorithms track position and velocity, and sometimes acceleration. However, other application specific quantities can also be included: for example, in counter battery fire, a state estimate of the drag characteristics of an in-coming projectile often helps when trying to determine the type of projectile and the original launch point (Johnson, 1973; Hutchins and Pace, 2006).

For a two-dimensional system, where the state vector contains four states: two positions (x and y) and the corresponding velocities (u and v), the relationship between the state vector and the measurement can be represented by a function, $\boldsymbol{H}(\boldsymbol{X})$, which acts on the state vector $\boldsymbol{X}$ to find the values that would be measured. For example, if one were to measure the two position states directly, but not the velocities, the measurement would reduce to a linear matrix expression,

$$\boldsymbol{H}(\boldsymbol{X}) = \boldsymbol{H} \cdot \boldsymbol{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ u \\ y \\ v \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

The action of the measurement matrix is to select only those states that are being measured by the sensor. The matrix $\boldsymbol{H}$ is a $m \times n$ matrix, where n is the number of states contained in $\boldsymbol{X}$ and m is the number of states being measured. This could be a function of time with different quantities being measured at different points in time. In more complicated situations, the function could be a nonlinear function of one or more of the states – see section 4.1.

Although the measurement operator does not necessarily measure all of the motional parameters represented in the state vector, it is important that all of the states are *observable* from the states that are measured, for example, velocity and acceleration states can be derived from successive measurements of position. For a discrete time linear system that is described by n states, whose dynamical evolution from time

step $k \rightarrow k+1$ is given by

$$\boldsymbol{X}(k+1) = \boldsymbol{F}(k) \cdot \boldsymbol{X}(k) + \boldsymbol{G}(k) \cdot \boldsymbol{U}(k) \tag{2}$$

where $\boldsymbol{F}(k)$ is the system (state transition) matrix and $\boldsymbol{G}(k)$ is the input gain associated with a set of controls $\boldsymbol{U}(k)$, observability can be determined by constructing the observability matrix given by

$$\boldsymbol{Q}_{\mathbf{O}} = \begin{bmatrix} \boldsymbol{H} \\ \boldsymbol{H} \cdot \boldsymbol{F} \\ \vdots \\ \boldsymbol{H} \cdot \boldsymbol{F}^{n-1} \end{bmatrix} \tag{3}$$

All of the states will be observable if the observability matrix has full rank (i.e. it must have n linearly independent rows or columns) – similar to controllability criteria for linear control systems, replacing $\boldsymbol{H}$ with $\boldsymbol{G}$. For target tracking applications, it is the tracking sensor that is controlled rather than the object being tracked, which is frequently "non-cooperative." (Examples will be restricted to discrete time cases throughout, for simplicity. However, the generalization to continuous time expressions is relatively straightforward).

The measurements will not be perfect, they will contain noise due to physical limitations on the resolution of the sensor or other inaccuracies. Generally, the measurement error is defined in terms of its covariance matrix $\boldsymbol{R}(k)$ which is a second-order measure of the expected errors and their correlations. For simple cases with Gaussian-distributed, white noise (i.e., uncorrelated in time) the statistical properties of the measurement errors are completely determined by the matrix $\boldsymbol{R}(k)$. For more complicated sources of measurement noise, the higher-order statistical moments, or full probability density function (PDF) are often required.

2.2 The Kalman and steady-state filters

The most common state estimation algorithm is the Kalman filter, which forms the basis for most practical tracking algorithms. It provides the backbone of many aerospace systems and is particularly common in tracking and navigation systems (Bar-Shalom, Li and Kirubarajan, 2001). The original paper Kalman (1960) is a classic reference, but introductory texts are often more approachable: good examples are du Plessis (1967), Welch and Bishop (2006) and Gershenfield (1999).

The main strengths of the Kalman filter are:

1. it is recursive – it only requires information on the current state and the current measurement, rather than reprocessing a whole time series of measurements;
2. it minimizes the covariance of the state – which is the most common measure of the size of the errors;
3. it is optimal for Gaussian-distributed, white measurement noise – and its performance is normally very good even when the measurement noise is not white and Gaussian;
4. it is (relatively) simple to implement.

The Kalman filter contains three main steps: measurement, innovation, and prediction. The iterative nature of the Kalman filter means that the ordering of the prediction and measurement steps can be different in different implementations, often it is given as prediction, measurement, and innovation. This is dependent on whether one views the first measurement as part of the filter or not. The initialization of the filter uses the first measured values to generate estimates for each of the states contained in the state vector: if some of the states are not measured directly (as in equation (1)) the initial state estimates are found from the first few measurements – for example, the difference of two position measurements giving a velocity.

At each time step, k, the measurement matrix $\boldsymbol{H}(k)$ is used to determine what the result of the measurement would be expected to be, given the current estimate of the state vector, which is denoted by the m-dimensional vector of measured states $\boldsymbol{Y}(k|k)$

$$\boldsymbol{Y}(k|k) = \boldsymbol{H}(k) \cdot \boldsymbol{X}(k) \tag{4}$$

This quantity represents the 'best guess' of what should be measured at time step k. The actual measurement at time step k is the m-dimensional vector $\boldsymbol{Y}(k)$. The difference between the expected measurement and the actual measurement, $\Delta\boldsymbol{Y}(k) = \boldsymbol{Y}(k) - \boldsymbol{Y}(k|K)$, represents the new information provided by the new measurement – in the sense that, if the actual measurement is exactly the same as the expected measurement, the measurement has not provided anything new or unexpected about the state vector, it has merely confirmed what was already known. The difference $\boldsymbol{\Delta Y}(k)$ is referred to as the *innovation* and it is used to update/innovate the state vector according to

$$\boldsymbol{X}_{+}(k) = \boldsymbol{X}(k) + \boldsymbol{K}(k) \cdot (\boldsymbol{Y}(k) - \boldsymbol{Y}(k|k)) \tag{5}$$

where $\boldsymbol{K}(k)$ is an $n \times m$ matrix called the *Kalman gain*. The Kalman gain matrix is constructed so as to minimize the covariance of the expected errors in the state vector. The Kalman filter tracks not only the state but also the expected error in the state vector through an estimate of the associated $(n \times n)$ covariance matrix $\boldsymbol{S}_X(k)$. As the system evolves, the added information derived from successive measurements

reduces the expected errors (i.e., improving the accuracy of the estimated states). The Kalman gain is dependent upon the expected errors in the state vector given by $\boldsymbol{S}_X(k)$ and the expected errors in the measurement given by the $m \times m$ covariance matrix $\boldsymbol{R}(k)$

$$\boldsymbol{K}(k) = \boldsymbol{S}_X(k) \cdot \boldsymbol{H}(k)^{\mathrm{T}} \cdot [\boldsymbol{H}(k) \cdot \boldsymbol{S}_X(k) \cdot \boldsymbol{H}(k)^{\mathrm{T}} + \boldsymbol{R}(k)]^{-1} \tag{6}$$

This choice of $\boldsymbol{K}(k)$ minimizes the expected error in the state vector, as determined by the covariance matrix which is given by

$$\boldsymbol{S}_X(k_+) = [\boldsymbol{I}_{n\times n} - \boldsymbol{K}(k) \cdot \boldsymbol{H}(k)] \cdot \boldsymbol{S}_X(k) \tag{7}$$

where $\boldsymbol{I}_{n\times n}$ is the n-dimensional identity matrix. The effect of the Kalman state update equation (5) is to form a weighted average with the weight associated with each measurement being inversely proportional to its expected error. If no measurement is available, the innovation step is not applied and the prediction step is applied to the unmodified state vector.

At each time step, the current state and its error are used to predict the expected state and expected error at the next time step, using the state transition matrix. The state estimates evolve according to equation (2), $\boldsymbol{X}(k_+) \rightarrow \boldsymbol{X}(k+1) = \boldsymbol{F}(k) \cdot \boldsymbol{X}(k_+)$ (ignoring the controls $\boldsymbol{U}(k)$ for simplicity) and the errors would normally evolve according to

$$\boldsymbol{S}_X(k+1) = \boldsymbol{F}(k) \cdot \boldsymbol{S}_X(k_+) \cdot \boldsymbol{F}(k)^{\mathrm{T}} \tag{8}$$

In the absence of other effects, the error covariances would vanish as more and more measurements are collected and the estimates become more and more accurate. The discrete nature of the measurement process and the restriction to linear systems means that the prediction step will always contain small errors arising from the incomplete characterization of the actual dynamics of the system being tracked. For example, in the two-dimensional example given above, for the four-dimensional state vector $\boldsymbol{X} = [x, u, y, v]^{\mathrm{T}}$, the system matrix is given by

$$\boldsymbol{F}(k) = \boldsymbol{F} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

which assumes that the object is moving with a constant velocity during each finite interval/time step Δt. The solution is to modify the evolution of the covariance matrix to allow for imperfections in the prediction step by adding a term to equation (8) called the *process noise*

$$\boldsymbol{S}_X(k+1) = \boldsymbol{F}(k) \cdot \boldsymbol{S}_X(k_+) \cdot \boldsymbol{F}(k)^{\mathrm{T}} + \boldsymbol{Q}(k) \tag{10}$$

where $\boldsymbol{Q}(k)$ is an $n \times n$ covariance matrix which depends on the nature of the inaccuracies in the prediction process. The effect of the process noise is to increase the expected error slightly during the prediction step.

The process noise matrix used in a practical tracking system depends upon the type of inaccuracies expected in the system. For example, the system matrix (9) assumes that the object being tracked has a constant velocity, which is unrealistic for most applications. A natural way to include the effect of a changing velocity is to assume that the target velocity is approximately constant but the target is subject to small perturbations (accelerations), which can be represented by an effective noise term. Perturbations can arise from a number of sources (target maneuver, drag, etc.) and the cumulative effect is equivalent to noise – being erratic and unpredictable. If the fluctuations in acceleration, (Δa), are assumed to be constant for each time step, the corresponding errors in velocity and position will be $(\Delta a)(\Delta t)$ and $\frac{1}{2}(\Delta a)(\Delta t)^2$ respectively. This is referred to as the *discrete white noise acceleration* model (Bar-Shalom, Li and Kirubarajan, 2001). If the fluctuations in acceleration are uncorrelated in x and y, this gives a process noise covariance matrix

$$\boldsymbol{Q}(k) = \boldsymbol{Q} = \begin{bmatrix} \frac{1}{4}(\Delta t)^4 & \frac{1}{2}(\Delta t)^3 & 0 & 0 \\ \frac{1}{2}(\Delta t)^3 & (\Delta t)^2 & 0 & 0 \\ 0 & 0 & \frac{1}{4}(\Delta t)^4 & \frac{1}{2}(\Delta t)^3 \\ 0 & 0 & \frac{1}{2}(\Delta t)^3 & (\Delta t)^2 \end{bmatrix} \sigma_{\mathrm{a}}^2 \tag{11}$$

where σ_{a} is the standard deviation of the acceleration (i.e., the "typical" acceleration fluctuation between time steps). In many cases, the size of the typical fluctuations will not be known a priori, and σ_{a} becomes a design parameter, which needs to be optimized for specific applications. This is the main way in which the Kalman filter can be "tuned" to optimize the performance of the tracker.

To summarize, the Kalman filter can be written as:

1. Obtain new measurement : $\boldsymbol{Y}(k)$
2. Generate expected measurement : $\boldsymbol{Y}(k|k) = \boldsymbol{H}(k)\boldsymbol{X}(k)$
3. Compute Kalman gain : $\boldsymbol{K}(k) = \boldsymbol{S}_X(k) \cdot \boldsymbol{H}(k)^{\mathrm{T}} \cdot (\boldsymbol{H}(k) \cdot \boldsymbol{S}_X(k) \cdot \boldsymbol{H}(k)^T + \boldsymbol{R}(k))^{-1}$
4. Update estimated state with innovation : $\boldsymbol{X}(k_+) = \boldsymbol{X}(k) + \boldsymbol{K}(k) \cdot (\boldsymbol{Y}(k) - \boldsymbol{Y}(k|k))$

5. Update estimated errors : $\boldsymbol{S}_X(k_+) = (\boldsymbol{I}_{n\times n} - \boldsymbol{K}(k) \cdot \boldsymbol{H}(k)) \cdot \boldsymbol{S}_X(k)$
6. Predict state vector : $\boldsymbol{X}(k+1) = \boldsymbol{F}(k) \cdot \boldsymbol{X}(k_+)$
7. Predict estimated errors : $\boldsymbol{S}_X(k+1) = \boldsymbol{F}(k) \cdot \boldsymbol{S}_X(k_+) \cdot \boldsymbol{F}(k)^{\mathrm{T}} + \boldsymbol{Q}(k)$
8. Repeat.

The Kalman filter can be shown to be optimal when the system is linear and the measurement noise is white and Gaussian-distributed because the construction of the Kalman gain minimizes the covariance at each state update, and the properties of the errors are completely defined by the covariance matrix when the noise is white and Gaussian. In most other situations, the performance of the Kalman filter is normally good, particularly when the system is approximately linear, because the central limit theorem ensures that the cumulative effect of the measurement noise tends to a Gaussian distribution as the number of measurements increases. There are a few cases where the measurement noise has an infinite variance and it is not Gaussian-stable and it does not obey the usual central limit theorem. In such cases, the Kalman filter can be modified to minimize a different statistical property that is finite – the "tail" covariance (Sournette and Ide, 2001).

In many cases, the Kalman filter converges quite rapidly to a steady state, where the reduction in the estimated errors due to the addition of a new measurement (innovation) balances the increase in the estimated errors due to the process noise. The Kalman gain matrix asymptotes to a fixed value. Alternatives to the Kalman filter use this convergence property to fix the Kalman gain matrix to the asymptotic value. They are sub-optimal because the weightings assigned to each measurement do not minimize the covariance over short periods of time. However, over longer periods of time, when the Kalman filter would have converged, these "steady state" filters provide accurate state estimates by fixing the Kalman gain matrix to the asymptotic value. Examples of these are the "alpha-beta" and "alpha-beta-gamma" filters (Bar-Shalom, Li and Kirubarajan, 2001), which are used to estimate target position and velocity states without the need to calculate the Kalman gain at each time step. These filters are useful because an analytic expression for the steady state gain matrix can be derived as a function of the target's ability to maneuver (Bar-Shalom, Li and Kirubarajan, 2001).

It is also possible to run a filter backwards in time, using later measurement data to reduce the errors in state estimates in the early stages of track formation. A reversed filter is often referred to as a *smoothing tracker*. It does not have the iterative form of the standard Kalman filter, since all the previous measurements need to be retained and reprocessed each at each subsequent time-step, but it is often useful for tracking targets backwards in time, for example, for counter battery fire or missile defense applications.

2.3 Track association and maintenance

When discussing the Kalman filter, it was assumed that measurements are produced at regular intervals (once per time step) and that each measurement can be assigned to the target being tracked. Unfortunately, tracking problems are rarely this simple. In practice, measurements are often intermittent or complicated by the detection of false alarms. For example, tracking a low-flying target with a radar might be complicated by radar returns from objects on the ground, or tracking a missile using an EO sensor can be difficult in cluttered scenes. At each time step it is necessary to verify each measurement to ascertain if it is likely to have come from the target being tracked or whether it comes from a background object or false alarm. The mis-association of false alarms with target tracks has two main effects. It tends to increase the actual tracking errors in the system because the introduction of measurements from false alarms will have different noise properties to those of the target. The estimated errors that propagate in the Kalman filter will not reflect the actual error distribution and the filter will be sub-optimal. Secondly, the track could become "locked" onto the false alarm rather than the true target. Once several false alarms have been associated with the track, the track states reflect the properties of the false alarm rather than the target, making mis-association more likely. This is the basis for many missile countermeasure systems (e.g., flares as EO countermeasures).

There are a number of methods that can be used to verify the measurements, the tracker produces an estimated position for the target and an estimate of the errors in that position. If the new measurement is too far from the existing track, then it is unlikely to correspond to the tracked object. A gating function can be applied to prevent associations being made when the distance between the measurement and the track exceeds a threshold value. The distance metric can be based either on the simple (Euclidean) distance, that is, the absolute difference between the new measurement and the predicted track as given by the norm of the innovation vector $\|\boldsymbol{\Delta Y}(k)\|$. With a Kalman filter, an estimate of the covariance matrix is available and can be used as a distance metric to determine the Mahalanobis distance

$$\boldsymbol{D}_{\mathrm{M}}(k) = \sqrt{\boldsymbol{\Delta Y}(k)^{\mathrm{T}} \boldsymbol{S}_{\mathrm{R}}(k)^{-1} \boldsymbol{\Delta Y}(k)} \tag{12}$$

where $S_{\mathrm{R}}(k) = \boldsymbol{H}(k) \cdot \boldsymbol{S}_X(k) \cdot \boldsymbol{H}(k)^{\mathrm{T}} + \boldsymbol{R}(k)$ is the residual covariance matrix. $\boldsymbol{D}_{\mathrm{M}}$ is a useful measure of the statistical significance of the difference between the new measurement and the existing track estimate. The statistical likelihood can be enforced by setting a threshold (gate) value for $\boldsymbol{D}_{\mathrm{M}}$, which has a χ^2 distribution with m degrees of freedom if the residual

errors are Gaussian distributed. If the acceptable error probability for valid measurements to be excluded from a track history is ϵ, a type-I error , the threshold value is set by

$$P(\boldsymbol{D}_{\mathrm{M}} < \boldsymbol{D}_{\mathrm{threshold}}) = \frac{\gamma(m/2, \boldsymbol{D}_{\mathrm{threshold}}/2)}{\Gamma(m/2)} = 1 - \epsilon \quad (13)$$

where $\gamma(a, x)$ and $\Gamma(a)$ are the incomplete Gamma function and the Gamma function respectively (Abramowitz and Stegun, 1964). Incorrectly assigning a false alarm to a track is a type-II error and the gate threshold must be fixed to balance the requirements for minimizing both types of errors – including true measurements and excluding false targets. To reduce false alarms, targets often have some physical property which is constant or approximately constant in time – for EO sensors this might be the intensity of the target or some geometrical property derived from the intensity distribution or shape. These attributes are often best employed just after a track has been created because the motional states will have large estimated errors, giving a wider window within which a false alarm could be incorrectly assigned to the track.

In addition to the generation of false alarms, the tracking system must also allow for possibility that the sensor may miss some target measurements. For low contrast or low signature targets, the detection probability could be significantly less than one. A missed detection means that the state estimation filter needs to generate a prediction (and estimated error) based on the current state alone. Over a series of missed detections, state errors can become large because the states are not being innovated/updated with new measurements. This increases the chances of mis-association of false alarms. In extreme cases, where no measurements have been generated for a period of time, the track may be deleted – indicating that there is no target. Once the target track has been deleted, a track can be re-initialized (created) once a number of measurements have been generated. The requirement for a minimum number of measurements to be taken before a track is created is another way to reduce the likelihood of tracking false alarms, which are often more intermittent than true targets. The parameters that control the track creation and deletion operations tend to be application specific and are determined by the detection probability of the sensor, the false alarm rate and other operational considerations.

3 MULTIPLE TARGET TRACKING

This section generalizes the tracking problem to multiple targets, where the number of targets may not be known and could be time dependent. The general structure of a multiple target tracking algorithm is similar to the single target case: detections/measurements are generated by a sensor, the measurements are associated with a target track or false alarm, if associated the track state vector is updated using the data from the measurement, and the resultant tracks are checked to remove redundant information. The main differences are in the complexity of the track maintenance algorithms and the introduction of multiple hypotheses corresponding to different possible ways of associating measurements with tracks. The state estimation process for updating each of the track state vectors is the same as for the single target case, using the Kalman filter (section 2.2) or more advanced algorithms (section 4).

3.1 Measurement-track association

For single targets, the association of measurements to tracks is relatively straightforward in the presence of a small number of false alarms. There is a single decision to be made for each measurement – does the measurement come from the target being tracked or not? When there is more than one target, there are additional problems deciding which target or track the measurement comes from. This is particularly important when targets are maneuvering and are close together. Where two tracks approach each other, the noisy measurements can be associated with more than one track and the construction of the tracks can be ambiguous, as in Figure 1. There are often several different ways of assigning measurements to tracks and, if the separation of the targets is similar to the size of the track or measurement uncertainties, these assignments could be equally likely. If this is the case, the preferred solution is a multiple hypothesis tracking algorithm based on ranking the different ways that the measurements could be assigned to the tracks according to their estimated probabilities over a series of measurements – see section 3.3.

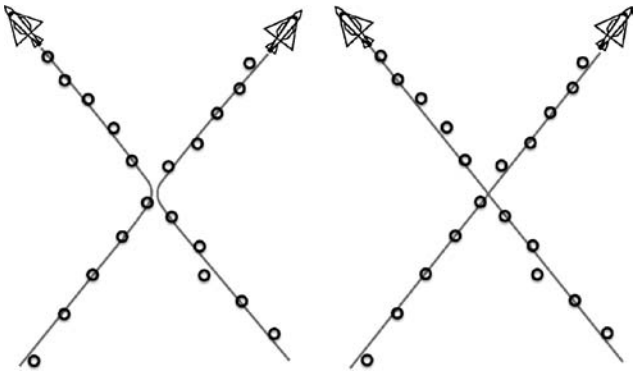

Figure 1. Two alternative trajectories which are consistent with sets of measurements.

Where the tracks are clearly distinguishable and there is little chance of ambiguous association, the problem is similar to that of the single target example, where the association can be made using a gate function based on a distance metric or Bayesian likelihood measure. The main difference for multiple targets is that a detection may fall within the allowable gate threshold for more than one track. The decision to associate a detection with a given track is then based on the number of tracks currently being held in the system, the geometrical distribution of tracks within the scene and the anticipated false alarm rate from the background.

For a system that has $\boldsymbol{N}(k)$ active tracks at a time step k (labelled $\boldsymbol{T}_K(k)$, $K = 1, \ldots, \boldsymbol{N}(k)$) and $\boldsymbol{M}(k)$ sets of measurements (labelled $\boldsymbol{Y}_J(k)$, $J = 1, \ldots, \boldsymbol{M}(k)$ and each corresponding to m measured states) will require the construction of an $\boldsymbol{M}(k) \times (\boldsymbol{N}(k) + 2)$ measurement-track association matrix, for example:

$$\boldsymbol{M}_{\text{assoc}}(k) = \begin{array}{c} \\ Y_1(k) \\ Y_2(k) \\ Y_3(k) \\ \vdots \\ Y_{M(k)}(k) \end{array} \begin{array}{c} \begin{array}{ccccccccc} \vdots & T_1(k) & T_2(k) & T_3(k) & \ldots & T_{N(k)} & \vdots & T_{\text{new}}(k) & FA \end{array} \\ \begin{bmatrix} \vdots & 0 & 1 & 0 & \ldots & 0 & \vdots & 0 & 0 \\ \vdots & 0 & 0 & 0 & \ldots & 0 & \vdots & 1 & 0 \\ \vdots & 0 & 0 & 1 & \ldots & 0 & \vdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & 1 & 0 & 0 & \ldots & 0 & \vdots & 0 & 0 \end{bmatrix} \end{array} \tag{14}$$

Each measurement should be associated to one and only one existing track, and if it is not associated to an existing track it will be associated to a new track or a possible false alarm (the final two columns in $\boldsymbol{M}_{\text{assoc}}$, labelled by T_{new} and FA respectively) so that the sum along each of the rows is one. The number of measurements associated to each of the existing tracks (the first $N(k)$ columns) will be zero or one – since the probability of detection for each of the targets ($P_{\text{D}}^{(K)}$)) is less than one, there is no guarantee that every existing track will be updated with a new measurement at each time step. The measurements that are not associated with existing tracks could be a genuine target track or a false alarm. In practice, to limit the number of tracks being created, a number of measurements will have to be associated with a new track before a full track is created (see section3.2). Intermittent false alarms will tend to be associated with false alarms (FA) so that they will not generate false tracks.

A common approach is to use the estimated state errors for each track and a model for the distribution of false alarms in the background to use the nearest neighbor as measured by the Mahalanobis distance (12), as with the single target case discussed in section 2.3. The probability for each of the measurements to be associated with each of the tracks (or associated as a false alarm, using an assumed distribution) is calculated. In the absence of prior knowledge regarding the distribution of false targets, the most common assumption is that false targets are uniformly distributed in space and their probability of occurrence is Poisson distributed with an average false alarm rate ($\bar{N}_{\text{FA}}$). The same assumption is often made for the appearance of new targets into the sensor field of view with an average rate of $\bar{N}_{\text{new}}$ (Cox and Hingorani, 1996).

Possible association matrices are ranked according to probability and the relative probability is constructed from several factors, corresponding to: the distance between the measurement and the corresponding track, the number of unassociated tracks/missed detections, and the number of false alarms, and new targets. If the track errors are Gaussian, the probability of associating measurements with tracks based on distance is related to the Mahalanobis distance by

$$\boldsymbol{P}_{\text{M}}[\boldsymbol{Y}_J(k), \boldsymbol{T}_K(k)] = \left|2\pi \boldsymbol{S}_{\text{R}}^{(J,K)}(k)\right|^{\frac{1}{2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{D}_{\text{M}}^{(J,K)}(k)\right]^2\right\} \tag{15}$$

where $\boldsymbol{D}_{\text{M}}^{(J,K)}(k) = \boldsymbol{\Delta Y}^{(J,K)}(k)^{\text{T}}(\boldsymbol{S}_{\text{R}}^{(J,K)})^{-1}\boldsymbol{\Delta Y}^{(J,K)}(k)$ is the Mahalanobis distance between the Jth measurement and the Kth track, and $\boldsymbol{S}_{\text{R}}^{(J,K)}$ is the residual covariance for the Jth measurement and the Kth track. Giving, for all of the tracks that have an associated measurement,

$$\boldsymbol{P}_{\text{Mah}}[\boldsymbol{M}_{\text{assoc}}(k)] = \prod_{K=1}^{N(k)} \boldsymbol{P}_{\text{M}}[\boldsymbol{Y}_J(k), \boldsymbol{T}_K(k)]^{\sum_{J=1}^{M(k)}[\boldsymbol{M}_{\text{assoc}}(k)]_{J,K}} \tag{16}$$

The sum $\sum_{J=1}^{M(k)}[\boldsymbol{M}_{\text{assoc}}(k)]_{J,K}$ is the sum over the Kth column of $\boldsymbol{M}_{\text{assoc}}(k)$, which is equal to one if a track has an associated measurement and zero if it does not have an associated measurement. The probability for $\boldsymbol{N}_{\text{det}}$ associated tracks and $\boldsymbol{N}_{\text{miss}} = N(k) - N_{\text{det}}$ missed detections is given by,

$$\begin{aligned}&\boldsymbol{P}_{\text{det}}[\boldsymbol{M}_{\text{assoc}}(k)]\\&= \frac{(N_{\text{FA}})!(N_{\text{new}})!}{M(k)!}\prod_{K=1}^{N(k)}\left[\begin{array}{l}\left(P_{\text{D}}^{(K)}\right)^{\sum_{J=1}^{M(k)}[\boldsymbol{M}_{\text{assoc}}(k)]_{J,K}}\\ \times\left(1-P_{\text{D}}^{(K)}\right)^{1-\sum_{J=1}^{M(k)}[\boldsymbol{M}_{\text{assoc}}(k)]_{J,K}}\end{array}\right]\end{aligned} \tag{17}$$

where $N_{\text{new}} = \sum_{J=1}^{M(k)}[\boldsymbol{M}_{\text{assoc}}(k)]_{J,N(k)+1}$ and $N_{\text{FA}} = \sum_{J=1}^{M(k)}[\boldsymbol{M}_{\text{assoc}}(k)]_{J,N(k)+2}$. The false alarm and new track probabilities are given by the corresponding density functions $p_{\text{FA}}(N_{\text{FA}})$ and $p_{\text{new}}(N_{\text{new}})$. If the distributions are taken to be Poission distributed,

$$p_{\bar{z}}(z) = \frac{(\bar{z})^z \mathrm{e}^{-\bar{z}}}{z!} \tag{18}$$

where z is either N_{new} or N_{FA}. These factors give a total probability for the association matrix,

$$\begin{aligned}&P_{\text{assoc}}^{(\text{tot})}[\boldsymbol{M}_{\text{assoc}}(k)]\\&= \frac{(\bar{N}_{\text{new}})^{N_{\text{new}}}(\bar{N}_{\text{FA}})^{N_{\text{FA}}}}{V}P_{\text{Mah}}[\boldsymbol{M}_{\text{assoc}}(k)]P_{\text{det}}[\boldsymbol{M}_{\text{assoc}}(k)]\end{aligned} \tag{19}$$

where V is a normalization constant – the probabilities must be normalized over all possible association matrices.

The space of all possible association matrices is often fairly sparse, there being a large number of very unlikely associations corresponding to measurements that are far from the existing track state estimates. In such cases, it is sometimes possible to select the association matrix that maximizes the total probability for all associations by maximizing the probability separately in each row. However, where there is more than one way that a measurement can be assigned to a track, the general problem can be solved by finding the association matrix that maximizes $P_{\text{assoc}}^{(\text{tot})}(\boldsymbol{M}_{\text{assoc}}k)$ over all possible combinations. The maximization of $P_{\text{assoc}}^{(\text{tot})}$ can be achieved using a standard optimization technique (Murty, 1968) but the sparse nature of problem can often be used to simplify the search strategy using an *ambiguity matrix*. The ambiguity matrix is an $\boldsymbol{M}(k) \times (\boldsymbol{N}(k) + 2)$ similar to the association matrix, which contains non-zero elements only where a measurement lies within the gating function for a track state and an association between track and measurement is possible or where a new track or false alarm are allowed (Cox and Hingorani, 1996).

3.2 Track creation and deletion

The track maintenance algorithms for a single target tracking system can be relatively simple. If no measurements are generated within a fixed period of time or the track errors become too large, the track can be placed in a quiescent or inactive state until another measurement is available. When more than one target is being tracked and the total number of targets is unknown, a more sophisticated set of track maintenance algorithms is required. The number of tracks will normally vary from time step to time step, with tracks being created when a series of detections represents a consistent set of measurements of the same object and then deleted once a number of time steps have elapsed with no associated measurement.

Track creation normally requires that a certain number of measurements can be associated to the same object before a track is generated. The number of measurements required to create a track is normally dependent on the expected false alarm rate for the sensor. The higher the false alarm rate the more likely it is that a series of false detections will be generated sufficiently close together to be associated together and a false track will be created. For some sensors, such as EO imagers, the background scene can contain clutter so that the probability of false alarms is not uniform in space. The resultant false alarms tend to be clustered together, thereby increasing the probability that a false track will be created. In these situations, it can be useful to retain a continuous false target track to prevent clutter from generating a large number of short duration tracks that will affect the computational efficiency of the tracking algorithms. In some situations, the maximum number of tracks will be fixed to limit the complexity of the tracking algorithms. To create a new track, a series of measurements must be sufficiently compelling for an existing track to be deleted – the likelihood of the measurements belonging to a new track must exceed the likelihood of one of the existing tracks.

Track deletion occurs either when the maximum number of tracks has been reached, when one of the tracks has had no measurement associated with it for a set period of time, or it can be determined stochastically with a probability being set for an unassociated track to be terminated (Cox and Hingorani, 1996). As with the single target tracking example, if no measurement is associated with a track the estimated errors in the state vector will increase because of dynamical inaccuracies. When the errors increase, the likelihood of associating a false alarm with the track may increase because

it widens the gate function used in measurement-track association. As a result, the parameters controlling the deletion process will depend on the detection probability and the false alarm rate of the sensor being used.

In addition to creation and deletion, it is often necessary to define other track maintenance functions which provide "book-keeping" functions and aim to simplify the track histories. Common examples are *track linking* and *track clustering* functions. If a valid target track is deleted because no measurements have been associated within a period of time but the corresponding target could still be within the field of view of the sensor, a further track could be created if the sensor detects the target again. An example could be a target which is being tracked but it becomes occluded (e.g., moving behind other objects in the scene). While the target is occluded, no detections will be generated and the track may be deleted. Once the target moves out from behind the occluding objects, the detections begin to be generate measurements. Track linking algorithms try to identify tracks which could correspond to the same target at different points and link the otherwise broken track. Track clustering algorithms perform a similar function, but they are aimed at tracks which are close together spatially and appear to share certain properties, such as the motional states. This could indicate several objects moving in formation or a single object which is being detected intermittently and the measurements are being associated with two or more tracks that "shadow" one another. In either case, it is possible to reduce the complexity by tracking the group of objects as one or merging the intermittent tracks into a single consistent track. In the case of multiple targets moving in formation, it is also necessary to maintain an estimate of the number of targets in each cluster and allow for targets within the group to separate and the track to branch. This group tracking approach is described in Blackman (1986), Chapter 11.

3.3 Multiple hypothesis trackers and disambiguation

The best way to handle cases where a measurement could be associated with more than one track is to create a set of tracks for each possible association. Each distinct set of tracks is referred to as a *hypothesis*, where the ith hypothesis at time step k, $\Theta^{(i)}(k)$ where $i = 1, \ldots, N_{\text{hyp}}(k)$, is derived from the same set of measurements but the track histories that it contains represent a different interpretation of the available data, that is, a different set of association matrices, $\boldsymbol{\Theta}^{(i)}(k) = (\boldsymbol{M}^{(i)}_{\text{assoc}}(1), \ldots, \boldsymbol{M}^{(i)}_{\text{assoc}}(k))$. Each hypothesis can be ranked according to how well the current tracks are thought to represent the underlying behavior of the targets. This is normally done by using the probability/likelihood for the measurement-track association over a number of time steps, possibly over the whole track history (Blackman, 1986). The likelihoods for each association are integrated over the track histories for each target present in each of the hypotheses, combining different $P^{(\text{tot})}_{\text{assoc}}$ values at each time step to find the probability for each hypothesis, $P^{(i)}_{\text{hyp}}$.

$$P^{(i)}_{\text{hyp}}[\boldsymbol{\Theta}^{(i)}(k)] = \prod_{r=1}^{k} P^{(\text{tot})}_{\text{assoc}} \left[M^{(i)}_{\text{assoc}}(r)\right] \tag{20}$$

The highest ranked hypothesis at any point in time is the most likely and therefore the preferred solution to the tracking problem. As the scenario evolves, and more measurements are acquired, the likelihoods can change and a relatively low-ranked hypothesis at one point in time can be reinforced by later measurements. For example, if two aircraft are being tracked by a radar – one agile fast-jet and one slower transport aircraft – the initial measurement data might indicate that target "one" is the jet and target "two" is the transport but a rapid maneuver by target "two" will re-weight the probabilities and reverse the target identities to reflect the new data. The original multiple hypothesis tracking (MHT) algorithm of this type was proposed by (Reid, 1979).

The problem with this approach is that it is computationally inefficient. For each potential ambiguity there will be at least two hypotheses, each containing a number of tracks. The total number of tracks – over all hypotheses – expands exponentially in time, taking more and more computational resources (memory and processing time). The challenge for a multiple target tracking system is to keep the computational demands of the algorithm within the bounds set by the available computational resources. Removing hypotheses where an association would not normally be made, as with the application of a gating function in section 2.3, can reduce the complexity of the tracking solution but including all of the viable hypotheses is still impractical in many cases (Nagarajan, Chidambara and Sharma, 1987). The most common approaches to this problem use a method based on an efficient implementation of the basic multiple hypothesis tracker proposed by Cox and Hingorani (1996).

The algorithm proposed by Cox and Hingorani uses the logarithm of the individual probabilities within equation (20) (the log likelihood) so that the highest ranked hypothesis is the one that maximizes the sum over all assignments within each of the possible hypotheses. This maximization can be cast as a standard cost minimization task – for which there are a number of standard algorithms, see Murty (1968) for examples – to find the best current solution to the problem. This best solution is removed from the list of hypotheses

and the process applied again to find the second best hypothesis, which is now the best in the absence of the previous hypothesis. This process can be applied iteratively to find the best $N_{\text{hyp}}(k)$ hypotheses, which can then propagate to the next time step $(k+1)$ and the next set of assignments. The Cox–Hingorangi algorithm also simplifies the problem by clustering nearby tracks and measurements and only considers ambiguous associations within the clusters, proposed by Reid (1979), and an efficient tree structure for storing the tracks, due to Kurien (1990). Finally, the algorithm reduces the computational load even further by pruning the hypothesis tree to a fixed number of time steps, $k_{\max}$ (Kurien, 1990). This is based on an assumption that any potential ambiguities will be resolved in a finite number of time steps. At time step $k = k'$, the hypothesis/track tree includes elements going back to time step $k = k' - k_{\max}$. At the subsequent time step, $k = k' + 1$, the pruning algorithm selects the branch of the track tree at $k = k' - k_{\max}$ that has a maximum probability on all of its subordinate branches. That branch is retained and all other branches at $k = k' - k_{\max}$ are removed.

4 AGILE TARGETS AND ADVANCED TRACKERS

The standard Kalman filter assumes that the measurement and the dynamical evolution of the system is linear. Both processes are represented by matrices and the update and evolution processes by linear matrix equations – see section 2.2. Of course, not all sensor systems are linear and many are manifestly nonlinear. The appearance of nonlinearities in the dynamics is often associated with agile targets. The use of a linear state transition matrix includes an assumption about the nature of the motion, where any unforeseen changes in the dynamics are represented by the process noise term in equation (10). The standard models for process noise tend to assume that changes in the target's maneuver are random, unbiased, and uncorrelated. In practice, targets maneuvers are highly correlated from time step to time step and correlations can be modeled by including nonlinear state transition processes. In this section, some common extensions of the standard Kalman filter for nonlinear systems are discussed with together with methods for dealing with agile targets.

4.1 The extended Kalman filter

The natural extension of the standard Kalman filter to deal with nonlinear measurements or nonlinear dynamical evolution is to modify equations (2) and (4) to replace the state transition and measurement matrices in the covariance update equations with matrices that represent linearizations of the nonlinearities around the current state estimates. This is valid as long as the nonlinearities are not too strong. To be a good approximation the nonlinearities should be approximately linear in the region of state space that is covered by the errors. For nonlinear measurements and dynamical evolution, the measurement and state transition equations become

$$\boldsymbol{Y}(k|k) = \boldsymbol{H}[k, X(k)] \tag{21}$$

$$\boldsymbol{X}(k+1) = \boldsymbol{F}[k, X(k_+)] \tag{22}$$

and the corresponding equations for the Kalman gain and the predicted covariance are

$$\boldsymbol{K}(k) = \boldsymbol{S}_X(k) \cdot \boldsymbol{H}'(k)^T \cdot [\boldsymbol{H}'(k) \cdot \boldsymbol{S}_X(k) \cdot \boldsymbol{H}'(k)^{\mathrm{T}} + \boldsymbol{R}(k)]^{-1} \tag{23}$$

and

$$\boldsymbol{S}_X(k+1) = \boldsymbol{F}'(k) \cdot \boldsymbol{S}_X(k_+) \cdot \boldsymbol{F}'(k)^{\mathrm{T}} + \boldsymbol{Q}(k) \tag{24}$$

where,

$$\boldsymbol{H}'(k) = \left.\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{X}=\boldsymbol{X}(k)} \tag{25}$$

and

$$\boldsymbol{F}'(k) = \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{X}=\boldsymbol{X}(k_+)} \tag{26}$$

Expanding the nonlinearities around the current state estimates allows the linear nature of the standard Kalman filter to be retained at the expense of introducing a dependence between the measurement and state transition processes and the state estimates. The danger in the dependence on the current state vector is that if the estimates are incorrect, due to a badly tuned state transition or measurement process, then the linearization will be incorrect, leading to increased inaccuracies in the subsequent state estimates. The increasing inaccuracies are a particular problem when using the extended Kalman filter and often additional checks need to be introduced to ensure that the filter errors are not reinforcing incorrect state estimates.

A related development of the standard Kalman filter which allows for nonlinear evolution of the error distribution is called the *unscented* Kalman filter (Julier and Uhlmann, 1997). Whereas the standard and extended Kalman filters

evolve the estimates for the state vector and the covariances of the state errors, the unscented Kalman filter generates a set of points surrounding the state vector which are determined by the current covariance estimates – the points are called sigma points. The sigma points are selected at each time step so that their mean and covariance match the current state vector and estimated error covariance. The sigma points are then evolved using the nonlinearities present in the system, giving a distribution from which a new mean and covariance can be found. Because the points evolve under the action of the nonlinearity, the state estimates and the estimated errors are usually more accurate than those derived from the linearized, extended Kalman filter. The unscented filter is also different from Monte Carlo methods because the sigma points are chosen deterministically – see section 4.2. The set of sigma points is minimal but it will only accurately represent the evolution of fairly well-behaved unimodal error distributions. For more complicated error distributions, a filter that estimates the distribution of errors explicitly is likely to be required.

4.2 Particle filters

The Kalman filter and its variants provides good state estimates in many situations where the measurement and process noise is Gaussian and any nonlinearities are relatively weak. Where the system nonlinearities are sufficiently strong, the state error distributions can be highly non-Gaussian, even when the input noise is Gaussian, in such cases Kalman-based methods are not necessarily robust. Strong nonlinearities tend to give rise to multi-modal distributions which cannot be modeled with a single Gaussian distribution, even using extended or unscented Kalman filters. One of the most popular approaches for dealing with such difficult cases is called the *particle filter* (Gordon, Salmond and Smith, 1993; Arulampalam, Maskell, Gordon and Clapp, 2002). The particle filter approximates the error distribution and its evolution by using a set of sample points drawn from the distribution which evolve under the action of the nonlinearities in the system. These sample points/particles (also known as or *support* points) are then used to construct an estimate of the new distribution at the end of the time step. Each sample point has a weight (or *importance*) associated with it, and it is the support points and their weights that propagate and are used by the particle filter to estimate the system state and the underlying error distribution.

The particle filter aims to estimate the posterior distribution for the state errors – that is, the probability distribution for the state vector given all of the measurements that have been obtained, $p(\boldsymbol{X}(k)|\boldsymbol{Y}(0), \ldots, Y(k))$. The aim is to use N_s sample points ($\boldsymbol{X}_i(k), i = 1, \ldots, N_s$) and their weights ($w_i(k)$) to construct an approximation of the posterior distribution,

$$p[\boldsymbol{X}(k)|\boldsymbol{Y}(0), \ldots, \boldsymbol{Y}(k)] \simeq \sum_{i=1}^{N_\mathrm{s}} w_i(k)\delta[\boldsymbol{X}(k) - X_i(k)] \quad (27)$$

where $\delta(\boldsymbol{X})$ is a delta function. The approximate density can then be used to find estimates for the expected values of a function $f(k)$ using a Monte Carlo approximation,

$$E[f(k)] = \int f[\boldsymbol{X}(k)]p(\boldsymbol{X}(k)|\boldsymbol{Y}(0), \ldots, \boldsymbol{Y}(k)) \simeq \sum_{i=1}^{N_\mathrm{s}} w_i(k) f[\boldsymbol{X}_i(k)] \quad (28)$$

The basic filter (Gordon, Salmond and Smith, 1993) updates the sample points $\boldsymbol{X}_i(k) \rightarrow \boldsymbol{X}_i(k_+)$ using a likelihood function,

$$q_i = \frac{p[\boldsymbol{Y}(k-1)|\boldsymbol{X}_i(k-1)]}{\sum_{j=1}^{N_\mathrm{s}} p[\boldsymbol{Y}(k-1)|\boldsymbol{X}_j(k-1)]} \quad (29)$$

where $p(\boldsymbol{Y}|\boldsymbol{X})$ is the probability of obtaining a measurement $\boldsymbol{Y}$ given a specific state vector $\boldsymbol{X}$. Given this likelihood function, a new set of sample points/particles are selected by taking a point from a uniform distribution u_i and selecting the sample point $\boldsymbol{X}_r(k-1)$ such that

$$\sum_{j=1}^{r-1} q_j < u_i \leq \sum_{j=1}^{r} q_j \quad (30)$$

This means that the new set of sample points $\boldsymbol{X}_i(k_+)$ are a re-weighted version of the previous set, $\boldsymbol{X}_i(k)$, the proportion of each distinct sample point in the new set acts as a simple weighting method where the proportion has been determined by the relative likelihood that the particle position is consistent with the measured state via equation (29).

The prediction stage then takes the N_s sample points and applies the state transition function to the updated points. To allow for the effect of inaccuracies in the dynamical model, a set of random sample points $v_i(k)$ drawn from the process noise distribution are included.

$$\boldsymbol{X}_i(k+1) = F[k, \boldsymbol{X}_i(k_+), v_i(k)] \quad (31)$$

giving a new set of points at time step k, with corresponding weights $w_i(k+1)$.

The basic particle filter requires an initial distribution for the sample points $p(\boldsymbol{X}(0))$, the likelihood function for the measurements $p(\boldsymbol{Y}(k)|\boldsymbol{X}(k))$ (which is equivalent to knowing the potentially nonlinear measurement interaction and the measurement noise distribution), and the density function for the process noise $p(\nu(k))$. It is iterative/recursive and relatively simple to implement. The main computational load comes from the number of sample points/particles required to adequately represent the underlying nonlinear distribution. However, a number of computational methods can be used to reduce the number of sample points and different variants of the basic particle filter have been developed to overcome other related problems (Arulampalam *et al.*, 2002). The more advanced versions of the basic particle filter and more general sequential Monte Carlo methods for have been used successfully in a wide range of applications (for a recent review see Doucet and Johansen, 2009).

4.3 Interacting multiple models

A full multiple hypothesis tracker is optimal but computationally inefficient. The Cox–Hingorangi algorithm simplifies the calculations but it is still computationally expensive. Reducing the complexity too far by limiting the number of hypotheses propagating from time step to time step improves the efficiency but reduces the performance of the tracker. An alternative approach to the problem is the interacting multiple model (IMM) algorithm (Bar-Shalom, Chang and Blom, 1989; Mazor *et al.*, 1998). This approach is sub-optimal but it is computationally simple and it is normally robust when tracking agile, maneuvering targets. The IMM algorithm uses multiple parallel trackers (or motion models) to track different possible maneuvers of the target or targets and provides a method for weighting each of the tracker outputs to provide an improved estimate for the target motion. The weights attached to each filter are recalculated after each time step with the weights being determined by the agreement between the track outputs and the track measurements. Often the tracking filters are simple Kalman filters with different process noise matrices, representing different target maneuver states – a large process noise for a target performing a high-g maneuver and a small process noise for smaller maneuvers. The ability of the IMM to adapt to rapid changes in target behavior is a major advantage of the IMM algorithm.

For the baseline IMM algorithm (Mazor *et al.*, 1998), if there are N_{tr} tracking filters, each producing a track state estimate $\boldsymbol{X}^{(j)}(k)$ and an estimated covariance matrix $\boldsymbol{S}_X^{(j)}(k)$ $(j = 1, \ldots, N_{\text{tr}})$ at time step k and the current probabilities assigned to each tracking filter are $p^{(j)}(k)$, then in the subsequent time step they will use a mixed filter state and error matrix,

$$\boldsymbol{X}_0^{(j)}(k) = \frac{1}{C^{(j)}} \sum_{i=1}^{N_{\text{tr}}} \boldsymbol{p}_{i,j} p^{(j)}(k) \boldsymbol{X}^{(j)}(k) \tag{32}$$

$$\boldsymbol{S}_{X0}^{(j)}(k) = \frac{1}{C^{(j)}} \sum_{i=1}^{N_{tr}} \boldsymbol{p}_{i,j} p^{(j)}(k) \times \left[\boldsymbol{S}_X^{(j)}(k) + \begin{bmatrix} \left[\boldsymbol{X}^{(i)}(k) - \boldsymbol{X}_0^{(j)}(k)\right] \\ \times \left[\boldsymbol{X}^{(i)}(k) - \boldsymbol{X}_0^{(j)}(k)\right]^{\text{T}} \end{bmatrix} \right] \tag{33}$$

where $C^{(j)}$ is a track normalization constant, $C^{(j)} = \sum_{i=1}^{N_{\text{tr}}} \boldsymbol{p}_{i,j} p^{(j)}(k)$, and $\boldsymbol{p}_{i,j}$ is the probability matrix for transitions to occur between the motion models. The jth filter is applied to the jth mixed filter state given by (32) using the mixed error covariance matrix (33), and the appropriate motion models (e.g., process noise matrix, $\boldsymbol{Q}^{(j)}$). Once the filters have been applied for the updated states, the weightings are updated using,

$$p^{(j)}(k+1) = \frac{1}{C} \Lambda^{(j)}(k) C^{(j)} \tag{34}$$

where C is a global normalization factor and $\Lambda^{(j)}(k)$ is the likelihood function for obtaining the innovation $\Delta \boldsymbol{Y}^{(j)}(k)$ from the jth filter – for Gaussian distributed errors with a residual covariance $\boldsymbol{S}_{\text{R}}^{(j)}(k)$, the likelihood is,

$$\Lambda^{(j)}(k) = \left|2\pi \boldsymbol{S}_{\text{R}}^{(j)}(k)\right|^{\frac{1}{2}} \times \exp\left\{ -\frac{1}{2} \left[\Delta \boldsymbol{Y}^{(j)}(k)^{\text{T}} \left(\boldsymbol{S}_{\text{R}}^{(j)} \right)^{-1} \Delta \boldsymbol{Y}^{(j)}(k) \right]^2 \right\} \tag{35}$$

The final mixed state estimate and mixed covariance matrix at the end of the time step is given by

$$\boldsymbol{X}(k+1) = \sum_{j=1}^{N_{\text{tr}}} p^{(j)}(k+1) \boldsymbol{X}^{(j)}(k+1) \tag{36}$$

$$\boldsymbol{S}_X(k+1) = \sum_{j=1}^{N_{\text{tr}}} p^{(j)}(k+1) \times \left[\boldsymbol{S}_X^{(j)}(k+1) + \begin{bmatrix} \left[\boldsymbol{X}^{(j)}(k+1) - \boldsymbol{X}(k+1)\right] \\ \left[\boldsymbol{X}^{(j)}(k+1) - \boldsymbol{X}(k+1)\right]^{\text{T}} \end{bmatrix} \right] \tag{37}$$

The IMM has several advantages in terms of computational complexity when compared to the MHT. It is iterative in both the trackers and the determination of the weightings attached to each filter. The use of multiple trackers/motion models means that the system can be modular, meaning that individual trackers can be modified, or replaced without a complete redesign of the whole algorithm.

5 SUMMARY

Radar and electro-optical sensors collect data but the measurements taken can be noisy and are sometimes unreliable. Objects do not always generate detections and the detections that are generated contain errors due to inherent inaccuracies in the sensor. The target tracking methods described in this section are used to turn these measurements into a consistent set of target/threat trajectories. The structure of a tracking algorithm normally follows a standard form: data alignment, measurement-track association, state estimation, and track maintenance. The detections are aligned to a common reference frame, within which the tracks are constructed. The aligned measurements are either associated with an existing track or they are used to generate a new track. Normally, this association is done using a probabilistic or other distance measure. The measurements are then combined with the existing state estimates form the track (position, velocity, etc.), to improve the accuracy of the estimates and update the motional state, in the case of an agile or a maneuvering target. Finally, the set of tracks is checked to see if it can be simplified – merging or linking tracks that seem to share common properties or removing tracks that are no longer generating measurements.

Simple target tracking systems tend to use simple nearest neighbor association techniques within some set distance (such as the likelihood gating techniques described in section 2.3) and basic state estimation techniques, such as the linear Kalman filter (see section 2.2). Difficulties arise when there are multiple targets or large numbers of false alarms, or when a target is agile and maneuvering rapidly.

Where a number of measurements are being generated at each time step and measurements could be associated with more than one track, each possible association needs to be considered. These ambiguities lead to a rapid increase in the computational complexity of the track structures, with each possible association (or hypothesis) propagating from time step to time step. The efficient tracking of multiple hypotheses requires that they are selected to maximize the posterior probabilities within defined computational resources – this is the basis for the multiple hypothesis trackers described in section 3.3. In such cases, the complexity of the tracking algorithm often needs to be traded-off against the accuracy and reliability of the tracking solution. Following all possible hypotheses may give optimal track estimates, but a slightly less reliable solution is often preferred if it can be generated in real time.

For agile targets that can change their motional state between measurements, there are added complications associated with the representation of the target dynamics. The linear dynamical model represented in the simple Kalman filter (section 2.2) is often inaccurate when describing maneuvering targets. The nonlinear properties of the target are often better captured in the nonlinear variants of the Kalman filter (described in section 4.1) or more sophisticated state estimation techniques such as the interacting multiple models (section 4.3) – where several possible motional states are propagated from time step to time step and the outputs selected, like the measurement-track hypotheses, by maximizing the posterior probabilities. For systems where the target dynamics and/or the measurement process is very nonlinear, it is necessary to model the evolution of the target and measurement probability density functions using sequential Monte Carlo or "particle" filtering techniques (see section 4.2). Like the multiple hypothesis trackers, these sophisticated state estimation techniques are often associated with relatively large computational overheads, but efficient trackers can be designed for specific applications that do not require large computational resources and do not compromise too much accuracy and reliability.

ACKNOWLEDGMENTS

The author would like to acknowledge the help of Prof. K. L. Edwards OBE (QinetiQ PLC) and Dr. N. Oxtoby (University of Liverpool) in the preparation of this manuscript.

REFERENCES

Abramowitz, M. and Stegun, I.A. (1964) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover.

Arulampalam, M.S., Maskell, S., Gordon, N. and Clapp, T. (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Proc.*, **50**(2), pp. 174–188.

Bar-Shalom, Y., Chang, K.C. and Blom, H.A.P. (1989) Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm. *IEEE Trans. Aerosp. Electron. Sys.*, **25**(2), 296–300.

Bar-Shalom, Y., Li, X.R. and Kirubarajan, T. (2001) *Estimation with Applications to Tracking and Navigation*. Wiley & Sons, Ltd.

Blackman, S. (1986) *Multiple Target Tracking with Radar Applications*. Artech House.

Cox, I.J. and Hingorani, S.L. (1996) An efficient inplementation of Reid's Multiple Hypothesis Tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**(2), 138–150.

Doucet, A. and Johansen, A. (2009) Particle filtering and smoothing: fifteen years later, in *Handbook of Nonlinear Filtering* (eds D. Crisan and B. Rozovsky), Oxford University Press, in press.

du Plessis, R.M. (1997) Poor Man's Explanation of Kalman Filters or How I Stopped Worrying and Learned to Love Matrix Inversion. *Rockwell International Technical Note* (Taygeta Scientific Incorporated, reprint 1997).

Gershenfeld, N. (1999) Filtering and State Estimation, in *The Nature of Mathematical Modelling*. Cambridge Press.

Gordon, N., Salmond, D. and Smith, A.F.M. (1993) Novel approach to nonlinear and non-Gaussian Bayesian state estimation. *IEE Proc. Pt. F*, **140**(2), 107–113.

Hutchins, R.G. and Pace, P.E. (2006) Studies in trajectory tracking and launch point determination for ballistic missile defense. *Proceedings of Signal and Data Processing of Small Targets* vol. 6236 (ed. Drummond OE), SPIE, Bellingham, WA. Paper 62360Y/1–8.

Johnson, S.L. (1973) Radar target idenitification system. US Patent 3733603, awarded May 1973.

Julier, S.J. and Uhlmann, J.K. (1997) A new extension of the kalman filter to nonlinear systems. *Proceedings of Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, (ed. Kadar I.), SPIE, Bellingham, WA, pp. 182–193.

Kalman, R.E. (1960) A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME–J. Basic Eng.*, **82**(1), 35–45.

Kurien, T. (1990) Issues in the design of pratical multitarget tracking algorithms. in *Multitarget-Multisensor Tracking Advanced Applications* (ed. Bar-Shalom Y.), Artech House, Norwood, MA, pp. 43–83.

Mazor, E., Averbuch, A., Bar-Shalom, Y. and Dayan, J. (1998) Interacting Multiple Model Mehtods in Target Tracking: A Survey. *IEEE Trans. Aerosp. Electron. Sys.*, **34**(1), 103–123.

Murty, K.G. (1968) An algorithm for ranking all the assignments in order of increasing cost. *Oper. Res.*, **16**, 682–687.

Nagarajan, V., Chidambara, M.R. and Sharma, R.N. (1987) Combinatorial problems in multitarget tracking – a comprehensive solution. *IEE Proc. Pt. F.*, **134**(1), 113–118.

Reid, D.B. (1979) An algorithm for tracking multiple targets. *IEEE Trans. Automat. Contr.*, **24**(6), 843–854.

Sournette, D. and Ide, K. (2001) The Kalman-Lévy filter. *Physica D*, **151**(2), 142–174.

Welch, G. and Bishop, G. (2006) An Introduction to the Kalman Filter. Technical Report University of Carolina. *TR 95-041* http://www.cs.unc.edu/w̃elch/media/pdf/kalman_intro.pdf (accessed 12 January 2009).